Algorithmic and complexity aspects of simple coalitional games

Haris Aziz

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Computer Science

Department of Computer Science

University of Warwick

Contents

Part II Computational voting

List of Tables

List of Figures

Acknowledgements

Gratitude is the memory of the heart.
- French Proverb

I am grateful to Almighty for being so kind to me. I am thankful to my wife Sana for being such a wonderful partner in every way. I thank my parents for their love and faith in me and to my siblings Kashif, Saba for supporting me. I also want to express my gratitude to both of my supervisors Prof. Mike Paterson and Prof. Dennis Leech for their assistance and patient guidance throughout the project. A PhD thesis does not only require academic support and, in this context, I am especially indebted to Mike who, despite his many commitments, always found time to discuss any issue.

I acknowledge the contribution of the Warwick 'Algorithms and Complexity' research group towards funding my research. It has been a pleasure to work with Oded Lachish and Rahul Savani. Prof. Artur Czumaj has also been a very supportive figure in the department. I want to thank the following people, projects and organizations for facilitating some useful research visits: International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), COST Action IC0602, British Colloquium for Theoretical Computer Science, Prof. Alexis Tsoukias and Dr. Ulle Endriss.

Thanks are also due to Elias, James and Jendrik for the numerous tennis hits we had. I would also like to express thanks to my teachers, especially Prof. Zaeem Jafri, Prof. Wasiq Hussain, Prof. Ismet Beg, Prof. Peter Jeavons and last, but certainly not the least, Prof. Sarmad Abbasi who got me hooked to theoretical computer science.

Kenilworth, *Haris Aziz*
July 2009

Declaration

This thesis contains published work and work which has been co-authored. The following chapters are based on various publications respectively: Chapter 3 [10], Chapter 4 [16], Chapter 5 [15], Chapter 6 [19], Chapter 7 [137], Chapter 8 [17], Chapter 9, [18], Chapter 10 [11], Chapter 12 [13] and Chapter 13 [14]. The research was led and conducted by me with support and feedback from my supervisors. The work on spanning connectivity games [13, 14] involved closely working with Rahul Savani and Oded Lachich.

Abbreviations and Symbols

SCG	spanning connectivity game
SVG	simple voting game (simple coalitional game)
MBC	minimal blocking coalition
MWC	minimal winning coalition
MWVG	multiple weighted voting game
WVG	weighted voting game
QMV	qualified majority voting
CoS	cost of stability
N	set of players/voters/agents
v	valuation/characteristic function
(N, v)	coalitional game
$\eta_i(v)$	Banzhaf value of player i in game v
$\beta_i(v)$	Banzhaf index of player i in game v
$\beta_i'(v)$	Probabilistic Banzhaf index/Penrose index of player i in game v
W	set of winning coalitions
W^m	set of minimal winning coalitions
ω	number of winning coalitions
ω_i	number of winning coalitions including player i
$\geq_D$	desirability relation between players
$H_i(v)$	Holler index of player i in game v
$D_i(v)$	Deegan-Packel index of player i in game v
$A(v)$	Coleman's Power of Collectivity to Act

$\tau[q; w_1, \ldots, w_n]$	tolerance of a weighted voting game
$\mu[q; w_1, \ldots, w_n]$	amplitude of a weighted voting game
$((N \setminus S) \cup \{\&S\}, v_{\&S})$	(N, v) where players in in coalition S have merged
$e(x, S)$	excess of coalition S according to payoff x
$-\epsilon_i(x, v)$	ith distinct worst excess for payoff x and game v
$-\epsilon_1(v)$	worst excess for a least core payoff of game v
$\delta_i(x, v)$	$1 - \epsilon_i(x, v)$
$\delta_1(v)$	least core payoff of a coalition with the worst excess
$A_x^i(v)$	set of coalitions that get excess $-\epsilon_i(x, v)$
$s_{ij}^v(x)$	maximum surplus of player i over player j with respect to x
$l(v)$	length of coalitional game (N, v)
$I^*(v)$	set of preimputations of game v
$I(v)$	set of imputations of game v
N_i	number of winning coalitions of cardinality i
G^x	weighted graph with edge e having weight $x(e)$
$\mathcal{E}(\alpha)$	edge partition for imputation α
$C_G(E')$	number of connected components in the graph $G \setminus E'$
$cr_G(E')$	cut-rate of edge set E' in graph G
opt	$\max_{E' \subseteq E} cr(E')$
$\mathcal{P}$	prime-partition of a graph
$\mathbf{O}$	parent-child relation in a graph

To late Professor Zaeem Jafri to whom I owe a lot

Abstract

Thesis Title:	Algorithmic & complexity aspects of simple coalitional games
By:	Haris Aziz
Place:	University of Warwick
Year:	2009
Supervisor:	Prof. Mike Paterson
Co-supervisor:	Prof. Dennis Leech
Thesis committee:	Prof. Paul Goldberg, Prof. Artur Czumaj and Prof. Alex Tiskin.

Simple coalitional games are a fundamental class of cooperative games and voting games which are used to model coalition formation, resource allocation and decision making in computer science, artificial intelligence and multiagent systems. Although simple coalitional games are well studied in the domain of game theory and social choice, their algorithmic and computational complexity aspects have received less attention till recently. The computational aspects of simple coalitional games are of increased importance as these games are used by computer scientists to model distributed settings. This thesis fits in the wider setting of the interplay between economics and computer science which has led to the development of algorithmic game theory and computational social choice. A unified view of the computational aspects of simple coalitional games is presented here for the first time. Certain complexity results also apply to other coalitional games such as skill games and matching games. The following issues are given special consideration: influence of players, limit and complexity of manipulations in the coalitional games and complexity of resource allocation on networks. The complexity of comparison of influence between players in simple games is characterized. The simple games considered are represented by winning coalitions, minimal winning coalitions, weighted voting games or multiple weighted voting games. A comprehensive classification of weighted voting games which can be solved in polynomial time is presented. An efficient algorithm which uses generating functions and interpolation to compute an integer weight vector for target power indices is proposed. Voting theory, especially the Penrose Square Root Law, is used to investigate the fairness of a real life voting model. Computational complexity of manipulation in social choice protocols can determine whether manipulation is computationally feasible or not. The computational complexity and bounds of manipulation are considered from various angles including control, false-name manipulation and bribery. Moreover, the computational complexity of computing various cooperative game solutions of simple games in different representations is studied. Certain structural results regarding least core payoffs extend to the general monotone cooperative game. The thesis also studies a coalitional game called the spanning connectivity game. It is proved that whereas computing the Banzhaf values and Shapley-Shubik indices of such games is #P-complete, there is a polynomial time combinatorial algorithm to compute the nucleolus. The results have interesting significance for optimal strategies for the wiretapping game which is a noncooperative game defined on a network.

Keywords: Cooperative games, game theory, algorithms and complexity, multiagent systems, network connectivity, network security, power indices, Shapley-Shubik index, Banzhaf index, Chow parameters, computational social choice, simple voting games, weighted voting games, nucleolus, least-core, cost of stability, resource allocation, preference aggregation, Nash equilibria, kernel, bargaining set, stable set, linear programming.

Association for Computing Machinery (ACM) Categories:
F.2 [Theory of Computation]: Analysis of Algorithms and Problem Complexity
I.2.11 [Distributed Artificial Intelligence]: Multiagent Systems
J.4 [Computer Applications]: Social and Behavioral Sciences - Economics

Mathematics Subject Classification (MSC): 91A12, 91A43, 91A46, 05C40, 68Q15, 68Q17, 68W40

Biographical sketch

Haris Aziz received a BSc(Honours) in Computer Science from *Lahore University of Management Sciences* in 2003 and an MSc in Mathematics and Foundations of Computer Science from the *University of Oxford* in 2005. He expects to receive a PhD in Computer Science from the *University of Warwick* in 2009. From August 2009, he is a European Science Foundation (ESF) postdoctoral researcher in a Europe-wide collaborative research project on "Computational Foundations of Social Choice" and is based at the *Ludwig-Maximilians-University Munich*.

Part I

Introduction

1

Introduction

There are no disciplines, nor branches of knowledge - or rather of research; there are only problems and the need to solve them.
- Karl Popper, "Realism and the Aim of Science" from the 'Postscript to the Logic of Scientific Discovery (1983)'

Many applications in computer science involve issues and problems that decision theorists have addressed for years, issues of preference, utility, conflict and cooperation, allocation, incentives, consensus, social choice, and measurement. A similar phenomenon is apparent more generally at the interface between computer science and the social sciences.
- Fred S. Roberts [186]

Abstract In this chapter, the general background and an outline of the thesis is presented.

1.1 Background

I do think there are some very worthwhile and interesting analogies between complexity issues in computer science and in economics. For example, economics traditionally assumes that the agents within an economy have universal computing power . . . Computer scientists deny that an algorithm can have infinite computing power.
- Richard Karp [117]

1.1.1 Game theory and computer science

Decision theory, game theory, and social choice theory are well-established fields which involve modeling the interaction between agents. Social choice concerns the aggregation of different self-interested agents' preferences. It encapsulates various important processes such as voting, markets and auctions where distributed agents want to make joint decisions. Game theory is the study of the conflict, cooperation and outcomes of interactions amongst multiple agents. Mechanism design is the design of games in such a way that individual players motivated by self-interest satisfy the desired goals of the designer. The desired goal could be individual rationality, budget balance, maximize total social welfare or to elicit truthful behaviour. Mechanism design[1] has been considered as the inverse of game theory [167].

Game theory, social choice theory and mechanism design, which were traditionally in the domain of economics and decision theory, are increasingly being used by computer scientists as tools to analyse distributed settings. With the growth of the internet, these fields provide an appropriate framework to model agents in the network [81]. Moreover, economics models and paradigms are being examined in the new light of the inherent computational complexity of the relevant problems. Nisan [158] elaborates on the two-way flow of ideas between economics and computer science. Not only are algorithmic challenges taking into account social choice and economic paradigms, but various economic interactions such as voting, coalition formation and resource allocation are requiring deeper algorithmic study. Similarly, Tennenholtz discusses the trend of the interaction between computer science/artificial intelligence and game theory/economics [203]. The same trend has also been pointed out elsewhere [47, 138]. In an earlier paper, Urken [208] shows that voting theory is essential in distributed decision making and network reliability. Rosenschein and Procaccia [187] observe how social choice theory is fundamental to analysing and designing multiagent systems and why algorithmic and complexity exam-

[1] Hurwicz, Maskin and Myerson received the 2007 Nobel Memorial Prize in Economic Sciences "for having laid the foundations of mechanism design theory".

ination is critical in social choice protocols. Wellman [215] points out that an economic approach is fundamental for resource allocation, rationality abstraction and decentralized control. The interface between game theory and computer science is further highlighted by Al Roth [188]. A computational perspective on game theoretic models is fundamental to new developments in computer science, game theory and multiagent systems. A combined game theoretic and algorithmic approach is even more important because of the convergence of social and technological networks [119].

1.1.2 Algorithmic game theory & computational social choice theory

> *As the ice separating Game Theory from Theoretical Computer Science is melting, some of the fundamental results in Game Theory come under increased complexity-theoretic scrutiny.*
> *- Fabrikant, Papadimitriou and Talwar [72]*

Mathematical economics has been around for decades. Although major developments have been made in presenting predictive theories and sound solution concepts, the treatment has mostly been non-algorithmic. It is essential that the solutions are not only axiomatically desirable and predictive but also computationally tractable. A computational perspective tries to answer various questions which are not tackled in classical economics and game theory: how efficiently can a model be represented? What is the complexity of computing a certain solution? How much memory will be required? Is this the best we can do? If computation is done on a network, how much communication is needed? Work on social choice and mechanism design has ignored such computational concerns till recent times. Therefore, there is a pressing need to revisit concepts in mathematical economics and game theory from a computational point of view. Economic systems work because all the participants try to selfishly optimize their own objectives. Often, this optimization is intractable because of the computational complexity of the optimization or lack of information. In that case, computational considerations are paramount.

The computational complexity of computing social choice functions, rankings, various concepts of equilibria, cooperative game solution concepts, tournament solutions, power indices, optimal or dominant strategies helps us understand what can be computed efficiently and what requires alternative approaches like approximation algorithms, randomized algorithms, parameterized complexity and heuristics. As we will see in the thesis, notions of computational intractability such as NP-hardness are useful barriers to manipulative behaviour in social choice settings. These have parallels with cryptographic protocols where the lack of efficient algorithms to factorize numbers helps avoid harmful attacks. The approach is that whereas social choice protocols may not be strategy-proof, it is desirable to design them in a way so that they are strategy-resistant. Another aspect of decision theory and game theory, which requires computer science considerations, is compact representation. As various models in game theory are implemented in computer science applications, there is a need to find more compact representations of games and ways of encoding information.

The interaction of social choice and game theory with computer science includes bounded rationality, computation of Nash equilibrium, algorithmic mechanism design, price of anarchy, learning in games and efficient representations of games, Byzantine agreement and implementing mediators. Interestingly, this intimate encounter between computer science and game theory dates back to von Neumann who made ground-breaking contributions to both fields. Game theory, decision theory and computer science have had more fruitful developments in recent years [186]. In the general realm of decision theory, decision making models have been classified (see Figure 1.1 [127]):

Among these models, simple coalitional games belong to the domains of both decision theory and cooperative game theory. Compared to non-cooperative game theory in which individual agents are analysed, cooperative game theory is concerned with analyzing which coalitions will form and how the coalitions should divide the payoff among their members. In recent years, there has been significant work in the theoretical computer science community on non-cooperative game theory. Whereas the computational complexity aspects of non-cooperative

<table>
<tr><td colspan="2" rowspan="2"></td><td colspan="2">Objectives</td></tr>
<tr><td>One</td><td>Two or more</td></tr>
<tr><td rowspan="2">Players</td><td>One</td><td>Operations Research</td><td>Multicrtieria Decision</td></tr>
<tr><td>Two or more</td><td>Cooperative Games</td><td>Noncooperative Games</td></tr>
</table>

Fig. 1.1. Decision making models

game theory, such as computing Nash equilibria, have started to be examined by theoretical computer scientists with greater intensity [168], there is a need to revisit cooperative game theory with a computational lens. Yoav Shoham and Leyton-Brown in the introduction of [195] have observed this need. Interestingly, in Chapter 13, we will find a case where cooperative game theory is used to solve a problem in non-cooperative game theory. In modeling decision making by one player, many problems turn out to be combinatorial optimization problems. However, when multiple players are involved in decision making, cooperative game theory [58] has a role to play in maximizing objectives of players and resource allocation. The computational complexity of computing solutions and deciding whether a payoff is in a class of solutions is an important consideration. This consideration also ties in with bounded rationality which argues that a decision maker can not spend an unbounded amount of resources.

Voting models are not necessarily restricted to analysing political scenarios. Similar interaction happens in multiagent systems and virtual environments. Consensus and voting problems arise in meta-search, collaborative filtering and distributed computing [128]. Shoham in a recent survey [194] observes that the interaction between computer science and game theory has currently been focused on six areas among which the first three are: 1) compact game representations, 2) complexity of, and algorithms for, computing solution concepts and 3) algorithmic aspects of mechanism design. In many respects, these current issues are addressed from the point of view of simple coalitional games.

1.2 Thesis introduction

> *...few structures arise in more contexts and lend themselves to more diverse interpretations than do simple games*
> *- Taylor and Zwicker [202]*

1.2.1 Overview

This thesis examines the computational and algorithmic aspects of simple voting games or cooperative simple games which are not only an important class of cooperative games but also a widely used voting model. The mathematical model of simple games is generic enough to model various scenarios. The research focusses on algorithms and the complexity of analysing the influence of players in game theoretic situations. This study of influence is significant in fields as diverse as percolation theory, reliability theory, political science and game theory [116]. The thesis also examines susceptibility of simple games, especially weighted voting games, to various kinds of manipulations. Manipulation is an urgent issue in multiagent systems and it has been observed that not only do coalitional voting games model various multiagent scenarios well, but computational complexity is seen as a useful barrier against manipulation.

A comprehensive investigation of the influence of players in simple games promises to be a useful contribution to the literature considering that the notion has not been explored much in the works [213] and [202]. Taylor and Zwicker note in the preface of [202] that the cardinal notions of power have not been mentioned in their book. Interestingly, such notions of power are now being explored much more in communities as diverse as reliability theory, political science and multi-agent systems. Voting power is also used in joint stock companies where each shareholder gets votes in proportion to the ownership of a stock [94]. An algorithms and complexity study of the influence of players is particularly relevant with the increase of large scale multi-agent systems. Moreover, in the manuscript on ‘Challenges for Theoretical Computer Science’ by Johnson [115], the following challenges are highlighted: preventing strategic voting, computing power indices, continuing exploring the impact of bounded rationality and developing a theory of algorithmic mechanism design.

1.2.2 Simple games

> *...we will arrive at an extensive class of games, to be called simple. It will be seen that a study of this class yields a body of information which is of value for a deep understanding of the general theory...*
> *- von Neumann and O. Morgenstern [213]*

Simple games (which are yes/no decision games) were introduced in the classical work of von Neumann and Morgenstern [213]. Von Neumann and Morgenstern point out that a study of simple games makes it possible to get an understanding of more general but harder to study zero-sum n-person games. Simple games have a rich mathematical history with contributions from game theorists, computer scientists, electrical engineers and combinatorialists. The history of simple games could even be stretched back to the famous *Dedekind problem* [120]. In 1897, Dedekind asked for the number $d(n)$ of free distributive lattices on n elements. This problem is equivalent to the number of simple games on n players. The Dedekind problem has been well studied. The function $d(n)$ grows rapidly and $d(n)$ is only known for very small n. Various algorithms have been proposed for efficient computation of $d(n)$ [85]. Simple games also have connections with *Sperner theory* [71]. As in the case of Taylor and Zwicker [202], we will discuss simple games in a voting-theoretic context. This is convenient both from an intuition and notation point of view.

Simple games and weighted voting games (which are a sub-class of simple games) are known in different literatures and communities by different names. There is considerable work on these models in threshold logic [216, 109, 154] and also in game theory (see [202] for a detailed literature references).

Weighted voting games (WVGs) are mathematical models which are used to analyze voting bodies in which the voters have different number of votes. In WVGs, each voter is assigned a non-negative weight and makes a vote in favour of or against a decision. The decision is made if and only if the total weight of those voting in favour of the decision is greater than or equal to some fixed quota. Since the weights of the players do not always exactly reflect how critical a player is in decision making, voting power attempts to measure the ability of a player in a WVG to determine the outcome of the vote. WVGs are also encountered in threshold logic, reliability theory, neuroscience and logical computing devices ([202], [208]). Parhami [171] points out that voting has a long history in reliability systems dating back to von Neumann [212]. For reliability systems, the weights of a WVG can represent the significance of the components whereas the quota can represent the threshold for the overall system to fail. Systems of this type are used in various areas such as target and pattern recognition, safety monitoring and human organization systems. WVGs have been applied in various political and economic organizations ([1]).

1.2.3 Approach of the thesis

> *...every definite mathematical problem must necessarily be susceptible of an exact settlement, either in the form of an actual answer to the question asked, or by the proof of the impossibility of its solution*
> *- David Hilbert (1900 lecture)*

The approach of the thesis is algorithmic. For many problems in cooperative game theory and social choice theory, there are mathematical results such as the existence or non-existence of properties. However, there is a need for an algorithmic study of these topics so that efficient constructive methods can be devised to test different properties of games. For various computational problems associated with simple coalitional games, polynomial time exact algorithms, pseudo-polynomial algorithms, approximation algorithms and parameterized algorithms are presented. In other cases, a proof is provided that the problem is, for instance, NP-hard or #P-complete.

1.3 Prerequisites

The thesis presupposes familiarity with combinatorial optimization and computational complexity. For readers unfamiliar with these areas, the following excellent book is recommended: [169]. In Section 2.2, non-technical definitions of fundamental complexity classes are given. One may also refer to Section 1.2 of [173] which outlines some basics of discrete optimization.

1.4 Thesis outline

The thesis is one of the first unified treatments of simple coalitional games from a computational perspective.

Chapter 3: In this chapter, the complexity of comparison of influence between players in simple games is characterized. The chapter is based on [10]. The influence of players is gauged from the viewpoint of basic player types, desirability relations and classical power indices such as the Shapley-Shubik index, Banzhaf index, Holler index, Deegan-Packel index and Chow parameters. Among other results, it is shown that for a simple game represented by its set of minimal winning coalitions W^m, although it is easy to verify whether a player has voting power zero or one, computing the Banzhaf value of the player is #P-complete. Moreover, it is proved that for multiple weighted voting games, it is NP-hard to verify whether the game is linear or not. For a simple game on n players and represented by W^m, a $O(n.|W^m| + n^2 \log\ n)$ algorithm is presented which returns 'no' if the game is non-linear and returns the strict desirability ordering otherwise. It is also shown that, for any reasonable representation of a simple game, a polynomial time algorithm to compute the Shapley-Shubik indices implies a polynomial time algorithm to compute the Banzhaf indices. As a corollary, we settle the complexity of computing the Shapley value of a number of network games. The complexity of transforming simple games into compact representations is also examined.

Chapter 4: It is well known that computing Banzhaf indices in a weighted voting game is #P-complete. We give a comprehensive classification of those weighted voting games which can be solved in polynomial time. Among other results, we provide a polynomial ($O(k(\frac{n}{k})^k)$) algorithm to compute the Banzhaf indices in weighted voting games in which the number of weight values is bounded by k. The chapter is based on [16].

Chapter 5: We study the mathematical and computational aspects of multiple weighted voting games which are an extension of weighted voting games. We analyse the structure of multiple weighted voting games and some of their combinatorial properties especially with respect to dictatorship, veto power, dummy players and Banzhaf indices. An illustrative Mathematica program to compute voting power properties of multiple weighted voting games is also provided. The chapter is based on the following publication: [15].

Chapter 6: The calculation of voting powers of players in a weighted voting game has been extensively researched in the last few years. However, the inverse problem of designing a weighted voting game with a desirable distribution of power has received less attention. We present an efficient algorithm which uses generating functions and interpolation to compute an integer weight vector for target Banzhaf power indices. This algorithm has better performance than any other known to us. It can also be used to design egalitarian two-tier weighted voting games and a representative weighted voting game for a multiple weighted voting game. The results in this chapter are based on paper [19] written with my supervisors.

Chapter 7: This chapter is based on a paper [137] jointly written with Dennis Leech. We tested a heuristic on the real life case-study of the EU constitution. The Double Majority rule in the Reform Treaty agreed in Rome in September 2004 is claimed to be simpler, more transparent and more democratic than the existing rule. We use voting power analysis to examine these questions against the democratic ideal that the votes of all citizens in whatever member country should be of equal value. We also consider possible future enlargements involving candidate countries and then a number of hypothetical future enlargements. We find the Double Majority rule fails to measure up to the democratic ideal in all cases. We find the Jagiellonian compromise to be very close to this ideal.

Chapter 8: An important aspect of mechanism design in social choice protocols and multiagent systems is to discourage insincere behaviour. Manipulative behaviour has received increased attention since the famous Gibbard-Satterthwaite theorem. We examine the computational complexity of manipulation in weighted voting games, which are ubiquitous mathematical models used in economics, political science, neuroscience, threshold logic, reliability theory and distributed systems. It is a natural question to check how changes in a weighted voting game may affect the overall game. The tolerance and amplitude of a weighted voting game signify the possible variations in a weighted voting game which still keep the game unchanged. We characterize the complexity of computing the tolerance and amplitude of weighted voting games. Tighter bounds and results for the tolerance and amplitude of key weighted voting games are also provided. Results from this chapter were published in [17].

Chapter 9: We examine the computational complexity of false-name manipulation in weighted voting games. This includes checking how much the Banzhaf index of a player increases or decreases if it splits up into sub-players. A pseudo-polynomial algorithm to find the optimal split is also provided. In the chapter, we also examine the cases where a player annexes other players or merges with them to increase their Banzhaf index or Shapley-Shubik index payoff. We characterize the computational complexity of such manipulations as well as providing limits to the manipulation. The *Annexation Non-monotonicity paradox* is also discovered in the case of the Banzhaf index. The results give insight into coalition formation and manipulation. The chapter is based on a paper [18] co-authored with Mike Paterson.

Chapter 10: This chapter is based on the following paper: [11]. Length and width are important characteristics of coalitional voting games which indicate the efficiency of making a decision. Duality theory also plays an important role in artificial intelligence. In this chapter, the complexity of problems concerning the length, width and minimal winning coalitions of simple games is analysed. The complexity of questions related to duality of simple games such as DUAL, DUALIZE and SELF-DUAL is also examined. Since susceptibility to manipulation is a major issue in multiagent systems, it is observed that the results obtained have direct bearing on susceptibility to optimal bribery in simple games.

Chapter 11: In this chapter, cooperative games and cooperative game solutions are introduced. The trend of using computational tractability as a criterion for cooperative game solutions is both recent and prevalent in the mathematics of operations research and theoretical computer science. In this chapter, the computational aspects of various cooperative game solutions in simple games are examined. Questions considered include the following: 1) for solution set X and simple game v, is X of v empty or not, 2) compute an element in X of v and 3) verify if a payoff is in X of v. Some representations taken into account are simple games represented by W, W^m, weighted voting games and multiple weighted voting games. The cooperative solutions considered are the core, ϵ-core, least-core, nucleolus, prekernel, kernel, bargaining set and stable sets. The complexity of checking the stability of the core of simple games is also examined. A theorem from the paper "The nucleolus and kernel for simple games or special valid inequalities for $0-1$ linear integer programs" by Wolsey is corrected. Finally, the relation between *cost of stability* and the least core is examined. A natural and desirable solution called the super-nucleolus is also proposed.

Chapters 12 and 13 concern spanning connectivity games (SCGs). They are based on joint work with Oded Lachish, Mike Paterson and Rahul Savani.

Chapter 12:

We examine the computational complexity of computing the voting power indices of edges in the SCG. It is shown that computing Banzhaf values is #P-complete and computing Shapley-Shubik indices or values is NP-hard for SCGs. Interestingly, Holler indices and Deegan-Packel indices can be computed in polynomial time. Among other results, it is proved that Banzhaf indices can be computed in polynomial time for graphs with bounded tree-width. Results from this chapter were published in [13].

Chapter 13: We consider the least core imputations and the nucleolus of SCGs. For any least core imputation, we refer to the *value* of SCGs as the payoff of any coalition with the worst excess. We show that the value is equal to the reciprocal of the *strength* of the underlying graph.

We efficiently compute a unique partition of the edges of the graph, called the prime-partition, and find the set of coalitions which always get the worst excess for every least core imputation. We define a partial order on the elements of the prime-partition which allows us to compute the nucleolus.

We also consider the problem of maximizing the probability of hitting a strategically chosen *hidden network* by placing a wiretap on a single link of a communication network. This can be seen as a two-player win-lose (zero-sum) game that we call the *wiretap game*. The nucleolus turns out be the unique maxmin strategy which satisfies certain desirable properties. Results from the chapter will be published in the following paper: [14].

Chapter 14: Conclusions and future directions of research are discussed.

2

Preliminaries

The advanced reader who skips parts that appear too elementary may miss more than the reader who skips parts that appear too complex.
- G. Polya

The beginning of wisdom is the definition of terms.
- Socrates

A definition is the enclosing of a wilderness of idea within a wall of words.
- Samuel Butler, Notebooks (1912)

Abstract In this chapter, the preliminary definitions concerning simple coalitional games and computational complexity are presented.

2.1 Simple coalitional games

Definition 2.1. *A* cooperative game with transferable utility *is a pair* (N, v) *where* $N = \{1, \ldots, n\}$ *is a set of players and* $v : 2^N \mapsto R$ *is a* characteristic/valuation function *that associates, for each coalition* $S \subseteq N$*, a payoff* $v(S)$ *which the coalition members may distribute among themselves.*

Throughout the thesis, when we refer to a cooperative game, we assume such a *TU-cooperative game* with transferable utility which can be freely transferred among players.

Definitions 2.2. *A* simple coalitional game/simple voting game *is a pair* (N, v) *with* $v : 2^N \rightarrow \{0, 1\}$ *where* $v(\emptyset) = 0$*,* $v(N) = 1$ *and* $v(S) \leq v(T)$ *whenever* $S \subseteq T$*. A coalition* $S \subseteq N$ *is* winning *if* $v(S) = 1$ *and* losing *if* $v(S) = 0$*. A simple voting game can alternatively be defined as* (N, W) *where W is the set of winning coalitions. This is called the* extensive winning form. *A* minimal winning coalition *(MWC) of a simple game v is a winning coalition in which defection of any player makes the coalition losing. The set of minimal winning coalitions of a simple game v can be denoted by* $W^m(v)$*. A simple voting game can be defined as* (N, W^m)*. This is called the* extensive minimal winning form.

For the sake of brevity, we will abuse the notation to sometimes refer to game (N, v) as v.

Definitions 2.3. *For each player* $x \in N$ *have weight* x_n*. The simple voting game* (N, v) *where*
$W = \{X \subseteq N, \sum_{x \in X} w_x \geq q\}$ *is called a* weighted voting game*(WVG). A weighted voting game is denoted by* $[q; w_1, w_2, ..., w_n]$ *where* w_i *is the non-negative voting weight of player i. Usually,* $w_i \geq w_j$ *if* $i < j$*.*

For many of the algorithms, our assumption that the weights of the WVG are non-negative is essential. Of course, any computational hardness results that hold for WVG with non-negative weights also hold for WVGS which have both negative and positive weights.

We now define multiple weighted voting games [2] which are an extension of weighted voting games.

Definitions 2.4. *An* m-multiple weighted voting game (MWVG) *is the simple game* $(N, v_1 \wedge \cdots \wedge v_m)$ *where the games* (N, v_t) *are the WVGs* $[q^t; w_1^t, \ldots, w_n^t]$ *for* $1 \leq t \leq m$*. Then* $v = v_1 \wedge \cdots \wedge v_m$ *is defined as:*

$$v(S) = \begin{cases} 1, \text{ if } v_t(S) = 1,\ \forall t,\ 1 \leq t \leq m, \\ 0, \text{ otherwise.} \end{cases}$$

The dimension *of* (N, v) *is the least k such that there exist WVGs* $(N, v_1), \ldots, (N, v_k)$ *such that* $(N, v) = (N, v_1) \wedge \ldots \wedge (N, v_k)$*.*

We now define some of the important properties of simple games:

Definition 2.5. *A simple game is*

- proper *if the complement of every winning coalition is losing.*
- strong *if the complement of every losing coalition is winning.*
- dual-comparable *if it is proper or strong.*
- decisive *if it is both proper and strong.*

The Banzhaf index [31] and Shapley-Shubik index [192] are two classic and popular indices to gauge the voting power of players in a simple game. They are used in the context of weighted voting games, but their general definition makes them applicable to any simple game.

Definition 2.6. *A player i is* critical *in a coalition S when $S \in W$ and $S \setminus \{i\} \notin W$. For each $i \in N$, we denote the number of* swings *or the number of coalitions in which i is critical in game v by the* Banzhaf value $\eta_i(v)$. *The* Banzhaf index *of player i in a simple game v is*

$$\beta_i(v) = \frac{\eta_i(v)}{\sum_{i \in N} \eta_i(v)}.$$

The probabilistic Banzhaf index (or Penrose index) *of player i in game v is equal to*

$$\beta_i'(v) = \eta_i(v)/2^{n-1}.$$

Intuitively, the Banzhaf value is the number of coalitions in which a player plays a critical role and the Shapley-Shubik index is the proportion of permutations for which a player is *pivotal*. For a permutation π of N, the $\pi(i)$th player is pivotal if coalition $\{\pi(1), \ldots, \pi(i-1)\}$ is losing but coalition $\{\pi(1), \ldots, \pi(i)\}$ is winning.

Definitions 2.7. *The* Shapley-Shubik value *is the function κ that assigns to any simple game (N, v) and any voter i a value $\kappa_i(v)$ where*

$$\kappa_i(v) = \sum_{X \subseteq N} (|X| - 1)!(n - |X|)!(v(X) - v(X - \{i\})).$$

The Shapley-Shubik index *of i is the function ϕ defined by*

$$\phi_i(v) = \frac{\kappa_i(v)}{n!}.$$

The *Shapley value* is a generalization of the Shapley-Shubik index. It has the same definition as the Shapley-Shubik index but is also applied to non-simple cooperative games. The Banzhaf index and the Shapley-Shubik index are the normalized versions of the Banzhaf value and the Shapley-Shubik value respectively. Since the denominator of the Shapley-Shubik index is fixed, computing the Shapley-Shubik index and Shapley-Shubik value have the same complexity. This is not necessarily true for the Banzhaf index and Banzhaf value. Only fact known is that if Banzhaf values can be computed, then they can be used to compute the Banzhaf indices.

Example 2.8. Consider WVG $[v = 51; 50, 49, 1]$ where the players are $\{A, B, C\}$. Then the winning coalitions are $\{A, B, C\}$, $\{A, B\}$ and $\{A, C\}$. Players A and B are critical in $\{A, B\}$, A and C are critical in $\{A, C\}$ and A is critical in $\{A, B, C\}$. Therefore $\eta_A(v) = 3$, $\eta_B(v) = 1$ and $\eta_C(v) = 1$ which means that $\beta_A(v) = 3/5$, $\beta_B(v) = 1/5, \beta_C(v) = 1/5$.

For the Shapley-Shubik index, we consider permutations. We identify the pivotal player in each of the following permutations. Player B is pivotal in ABC because $\{A\}$ is not winning but $\{A, B\}$ is winning. Similarly C is pivotal for ACB and A is pivotal for BAC, BCA, CAB and CBA. Therefore $\phi_A(v) = 2/3$, $\phi_B(v) = 1/6$ and $\phi_C(v) = 1/6$.

In voting games, another relevant consideration is the ease with which a decision can be made. This concept was introduced by Coleman in [49]:

Definition 2.9. Coleman's power of the collectivity to act*, A, is defined as the ratio of the number of winning coalitions* $|W|$ *to* 2^n*:* $A = |W|/2^n$.

Both Coleman's power of the collectivity to act and the probabilistic Banzhaf index (or Penrose index) will be used in Chapter 7. Chow parameters are another important parameters of a simple game.

Definition 2.10. *([63, 48]) For a simple game v, the* Chow parameters*, CHOW(v) are given by* $(|W_1|, \ldots |W_n|; |W|)$ *where* $W_i = \{S \in W : i \in S\}$. $|W|$ *and* $|W_i|$ *are also denoted by* ω *and* ω_i.

Apart from the Banzhaf and Shapley-Shubik indices, there are other indices which are also used. Both the Deegan-Packel index [56] and the Holler index [106] are based on the notion of minimal winning coalitions. Minimal winning coalitions are significant with respect to coalition formation [55]. The Holler index, H_i of a player i in a simple game is similar to the Banzhaf index except that only swings in minimal winning coalitions contribute toward the Holler index.

Definitions 2.11. *Let* M_i *be* $\{S \in W^m : i \in S\}$. *We define the* Holler value *as* $|M_i|$. *The* Holler index *(also called the* public good index*) is defined by*

$$H_i(v) = \frac{|M_i|}{\sum_{j \in N} |M_j|}.$$

The Deegan Packel index *for player i in voting game v is defined by*

$$D_i(v) = \frac{1}{|W^m|} \sum_{S \in M_i} \frac{1}{|S|}.$$

Compared to the Banzhaf index and the Shapley-Shubik index, both the Holler index and the Deegan-Packel index do not always satisfy the monotonicity condition.

2.2 Computational complexity

O time! thou must untangle this, not I; It is too hard a knot for me to untie!
- William Shakespeare

There is no greater harm than that of time wasted.
- Michelangelo

The *computational complexity* of problems related to simple games is central to this thesis. Computational complexity may refer to time complexity or space complexity. We will normally refer to the time complexity of a problem as the complexity of the problem. The time complexity of a problem is the number of steps required to solve an instance of the problem as a function of the size of the input (measured in bits), using the most efficient algorithm. The big O notation is a standard way to describe computational complexity. Let $f(x)$ and $g(x)$ be functions defined on some subset of the real numbers. Then

$$f(x) = O(g(x)) \text{ for all } x \to \infty$$

if and only if there exists a positive real number M and a real number x_0 such that

$$|f(x)| \leq M|g(x)| \text{ for all } x > x_0.$$

We define some basic computational complexity classes in lay terms for readers not familiar with computational complexity.

Definition 2.12. *A problem is in complexity class* P *if it can be solved in time which is polynomial in the size of the input. A problem is in the complexity class* EXP *if it can be solved in time exponential in the size of the input. A problem is in the complexity class* NP *if its solution can be verified in time which is polynomial in the size of the input of the problem. A problem is in complexity class* co-NP *if and only if its complement is in* NP*. A problem is in the complexity class* NP-hard *if any problem in* NP *is polynomial time reducible to that problem.* NP-complete *problems are in* NP *and are as hard as the hardest problems in* NP*. A counting problem is in complexity class* #P *if the objects being counted can be verified in polynomial time. A* #P-hard *problem is a counting problem which is as hard as the counting version of any* NP-hard *problem. A counting problem which in* #P *and is* #*P*-hard *is* #*P*-complete

Polynomial time algorithms are desirable because they 'scale' well and finish in a reasonable time compared to exponential time algorithms.

The *Partition Problem* is an example of a classic NP-complete problem which we will use at times in the thesis:

Name: PARTITION
Instance: A set of k integer weights $A = \{a_1, \ldots, a_k\}$.
Question: Is it possible to partition A, into two subsets $P_1 \subseteq A$, $P_2 \subseteq A$ so that $P_1 \cap P_2 = \emptyset$ and $P_1 \cup P_2 = A$ and $\sum_{a_i \in A_1} a_i = \sum_{a_i \in A_2} a_i$?

Readers unfamiliar with computational complexity may ask what is the use of this concept. Computational complexity is an inherent mathematical property of a problem irrespective of the model of computer. Some may still ask that why would bad news of a problem being NP-hard be of any use in real life. Of course one would prefer that a problem has an algorithm which can be run in time polynomial of its input. However, NP-hardness of a problem implies that no polynomial time algorithm is possible unless P=NP, i.e. the computational classes P and NP coincide, which is generally considered unlikely.

The theory of parameterized complexity is motivated by the fact that several NP-hard problems (for which no polynomial time algorithm is known) are solvable in a time that is polynomial in the input size and exponential in a (small) parameter k. Any problem τ can be defined in its corresponding parameterized form where the parameterized problem is the original problem τ along with some parameter k.

Definition 2.13. *A parameterized problem τ with an input instance n and parameter k is called* fixed-parameter tractable *if there is an algorithm which can solve τ in $O(f(k)n^c)$ where $c > 0$ and f is a computable function depending solely on k. The class of all fixed-parameter tractable problems is called* FPT.

Part II

Computational voting

3

Complexity of comparison of influence of players in simple games

The mathematical study (under different names) of pivotal agents and influences is quite basic in percolation theory and statistical physics, as well as in probability theory and statistics, reliability theory, distributed computing, complexity theory, game theory, mechanism design and auction theory, other areas of theoretical economics, and political science.
- Kalai and Safra, [116]

Not everything that counts can be counted, and not everything that can be counted counts.
- Einstein

Abstract In this chapter, the complexity of comparison of influence between players in coalitional voting games is characterized. The possible representations of simple games considered are by winning coalitions, minimal winning coalitions, weighted voting game or multiple weighted voting games.

It is also shown that for any reasonable representation of a simple game, a polynomial time algorithm to compute the Shapley-Shubik indices implies a polynomial time algorithm to compute the Banzhaf indices. As a corollary, we settle the complexity of computing the Shapley value of a number of network games.

3.1 Introduction

3.1.1 Overview and outline

John von Neumann and Morgenstern [213] observe that minimal winning coalitions are a useful way to represent simple games. A similar approach has been taken in [88]. We examine the complexity of computing the influence of players in simple games represented by winning coalitions, minimal winning coalitions, weighted voting games and multiple weighted voting games.

In Section 3.2, we outline different representations and properties of simple games. In Section 3.3, compact representations of simple games are considered. After that, the complexity of computing the influence of players in simple games is considered from the point of view of player types (Section 3.4), desirability ordering (Section 3.5), power indices and Chow parameters (Section 3.6). The final Section 3.7 includes a summary of results and some open problems.

3.2 Background

We first provide some important definitions and facts needed for the chapter.

3.2.1 Definitions

Definition 3.1. *A coalition S is* blocking *if its complement $(N \setminus S)$ is losing. For a simple game $G = (N, W)$, there is a* dual game $G^d = (N, W^d)$ *where W^d contains all the blocking coalitions in G.*

Definitions 3.2. *A WVG $[q; w_1, \ldots, w_n]$ is* homogeneous *if $w(S) = q$ for all $S \in W^m$. A simple game (N, v) is* homogeneous *if it can be represented by a homogeneous WVG. A simple game (N, v) is symmetric if $v(S) = 1$, $T \subset N$ and $|S| = |T|$ implies $v(T) = 1$.*

It is easy to see that symmetric games are homogeneous with a WVG representation of $[k; \underbrace{1, \ldots, 1}_{n}]$ for some k. That is the reason they are also called k-out-of-n simple games.

We will often use the following lemma.

Lemma 3.3. *For a simple game* (N, W)*,* W^m *can be computed in polynomial time.*

Proof. For every $S \in W$, check if $S \setminus \{i\}$ is winning for all $i \in S$. If yes for any such i, then $S \notin W^m$. Otherwise $S \in W^m$. This takes time $|input|^2$. □

3.2.2 Desirability relation and linear games

The individual desirability relations between players in a simple game date back at least to Maschler and Peleg [150].

Definitions 3.4. *In a simple game* (N, v)*,*

- *A player* i *is* more desirable/influential *than player* j *(*$i \succeq_D j$*) if* $v(S \cup \{j\}) = 1 \Rightarrow v(S \cup \{i\}) = 1$ *for all* $S \subseteq N \setminus \{i, j\}$*.*
- *Players* i *and* j *are* equally desirable/influential *or* symmetric *(*$i \sim_D j$*) if* $v(S \cup \{j\}) = 1 \Leftrightarrow v(S \cup \{i\}) = 1$ *for all* $S \subseteq N \setminus \{i, j\}$*.*
- *A player* i *is* strictly more desirable/influential *than player* j *(*$i \succ_D j$*) if* i *is more desirable than* j*, but* i *and* j *are not equally desirable.*
- *A player* i *and* j *are* incomparable *if there exist* $S, T \subseteq N \setminus \{i, j\}$ *such that* $v(S \cup \{i\}) = 1$*,* $v(S \cup \{j\}) = 0$*,* $v(T \cup \{i\}) = 0$ *and* $v(T \cup \{j\}) = 1$*.*

Linear simple games are a natural class of simple games:

Definitions 3.5. *A simple game is* linear *whenever the desirability relation* $\succeq_D$ *is complete, that is, any two players* i *and* j *are comparable (*$i \succ_D j$*,* $j \succ_D i$ *or* $i \sim_D j$*).*

For linear games, the relation $R_\sim$ divides the set of voters N into equivalence classes $N/R_{\sim_D} = \{N_1, \ldots, N_t\}$ such that for any $i \in N_p$ and $j \in N_q$, $i \succ_D j$ if and only if $p < q$.

Definitions 3.6. *A simple game* v *is* swap robust *if an exchange of two players from two winning coalitions cannot render both coalitions losing. A simple game is* trade robust *if any arbitrary redistributions of players in a set of winning coalitions does not result in all coalitions becoming losing.*

It is easy to see that trade robustness implies swap robustness. Taylor and Zwicker [202] proved that a simple game can be represented by a WVG if and only if it is trade robust. Moreover they proved that a simple game being linear is equivalent to it being swap robust.

Taylor and Zwicker [202] show in Proposition 3.2.6 that v is linear if and only if $\succ_D$ is acyclic which is equivalent to $\succ_D$ being transitive. This is not guaranteed in other desirability relations defined over coalitions [64].

Proposition 3.7. *A simple game with three or fewer players is linear.*

Proof. For a game to be non-linear, we want to players 1 and 2 to be incomparable, i.e., there exist coalitions $S_1, S_2 \subseteq N \setminus \{1,2\}$ such that $v(\{1\} \cup S_1) = 1$, $v(\{2\} \cup S_1) = 0$, $v(\{1\} \cup S_2) = 0$ and $v(\{2\} \cup S_2) = 1$. This is clearly not possible for $n = 1$ or 2. For $n = 3$, without loss of generality, v is non-linear only if $v(\{1\} \cup \emptyset) = 1$, $v(\{2\} \cup \emptyset) = 0$, $v(\{1\} \cup \{3\}) = 0$ and $v(\{2\} \cup \{3\}) = 1$. However the fact that $v(\{1\} \cup \emptyset) = 1$ and $v(\{1\} \cup \{3\}) = 0$ leads to a contradiction. □

In Example 3.17, we present a 4-player simple game which is not linear.

3.3 Compact representations

Since WVGs and MWVGs are compact representations of coalitional voting games, it is natural to ask which voting games can be represented by a WVG or MWVG and what is the complexity of answering the question. Deineko and Woeginger [57] show that it is NP-hard to verify the dimension of MWVGs. We know that every WVG is linear but not every linear game has a corresponding WVG. Carreras and Freixas [42] show that there exists a six-player simple linear game which cannot be represented by a WVG. We now define problem *X-Realizable* as the problem to decide whether game v can be represented by form X.

Proposition 3.8. *WVG-Realizable is NP-hard for a MWVG.*

Proof. This follows from the proof by Deineko and Woeginger [57] and Elkind et al. (Theorem 5, [69]) that it is NP-hard to check if the dimension of a MWVG is one or more. □

Proposition 3.9. *WVG-Realizable is in* P *for a simple game represented by its minimal winning, or winning, coalitions.*

This follows from Theorem 2 in [174] where the complexity of the problem was examined in the context of set covering problems. The idea in [174] is that if a simple game represented by minimal winning coalitions is not linear, then it is not WVG-Realizable. Peled and Simone [174] showed that this can be checked in polynomial time. They also showed that for linear simple games represented by minimal winning coalitions, all maximal losing coalitions can be computed in polynomial time. Also any simple game can be represented by linear inequalities for minimal winning coalitions and maximal losing coalitions. The idea dates back at least to [109]. However it is one thing to know whether a simple game is WVG-Realizable and another thing to actually represent it by a WVG. It is not easy to represent a WVG-Realizable simple game by a WVG where all the weights are integers as the problem transforms from linear programming to integer programming.

Proposition 3.10. *(Follows from Theorem 1.7.4 of Taylor and Zwicker[88]) Any simple game is MWVG-Realizable.*

Taylor and Zwicker [202] showed that for every $n \geq 1$, there is a simple game of dimension n. In fact it has been pointed out by Freixas and Puente [91] that, for every $d \geq 1$, there is linear simple game of dimension d. This shows that there is no clear relation between linearity and dimension of simple games. However it appears exceptionally hard to actually transform a simple game (N, W) or (N, W^m) to a corresponding MWVG. The dimension of a simple game may be exponential ($2^{(n/2)-1}$) in the number of players [202]. A simpler question is to examine the complexity of computing, or getting a bound for, the dimension of simple games.

3.4 Complexity of player types

A player in a simple game may be of various types depending on its level of influence.

Definitions 3.11. *For a simple game v on a set of players N, player i is a*

- dummy *if and only if* $\forall S \subseteq N$ *if* $v(S) = 1$, *then* $v(S \setminus \{i\}) = 1$*;*
- passer *if and only if* $\forall S \subseteq N$ *if* $i \in S$, *then* $v(S) = 1$*;*
- vetoer *if and only if* $\forall S \subseteq N$ *if* $i \notin S$, *then* $v(S) = 0$*;*
- dictator *if and only if* $\forall S \subseteq N$ $v(S) = 1$ *if and only if* $i \in S$.

It is easy to see that if a dictator exists, it is unique and all other players are dummies. This means that a dictator has voting power one, whereas all other players have zero voting power. We examine the complexity of identifying the dummy players in voting games. We already know that for the case of WVGs, Matsui and Matsui [151] proved that it is NP-hard to identify dummy players. For any of the player type T (dummies/passers/vetoers/dictator), we shall refer to the problem of computing players of type T by IDENTIFY-T.

Lemma 3.12. *A player i in a simple game v is a dummy if and only if it is not present in any minimal winning coalition.*

Proof. Let us assume that player i is a dummy but is present in a minimal winning coalition. That means that it is critical in the minimal winning coalition which leads to a contradiction. Now let us assume that i is critical in at least one coalition S such that $v(S \cup \{i\}) = 1$ and $v(S) = 0$. In that case, we can delete all players j other than i the deletion of which does not change the coalition from winning to losing. Then, there is an $S' \subset S$ such that $S' \cup \{i\}$ is a MWC. □

Proposition 3.13. *For a simple game v,*

1. *Dummy players can be identified in linear time if v is of the form* (N, W^m).
2. *Dummy players can be identified in polynomial time if v is of the form* (N, W).

Proof. We examine each case separately:

1. By Lemma 3.12, a player is a dummy if and only if it is not in any member of W^m. Therefore, check each $S \in W^m$ and if a player i is in S, then it is not a dummy. Then, any player which is not in any $S \in W^m$ is a dummy.
2. By Lemma 3.3, W^m can be computed from W in polynomial time.

□

From the definition, we know that a player has veto power if and only if the player is present in every winning coalition.

Proposition 3.14. *Vetoers can be identified in linear time for a simple game in the following representations:* (N, W)*,* (N, W^m)*, WVG and MWVG.*

Proof. We examine each of the cases separately:

1. (N, W): Initialize all players as vetoers. For each winning coalition, if a player is not present in the coalition, remove him from the list of vetoers.
2. (N, W^m): If there exists a winning coalition which does not contain player i, there will also exist a minimal winning coalition which does not contain player i.
3. WVG: For each player i, i has veto power if and only if $w(N \setminus \{i\}) < q$.
4. MWVG: For each player i, i has veto power if and only if $N \setminus \{i\}$ is losing.

□

Proposition 3.15. *For a simple game represented by* (N, W)*,* (N, W^m)*, WVG or MWVG, it is easy to identify the passers and the dictator.*

Proof. We check both cases separately:

1. Passers: This follows from the definition of a passer. A player i is a passer if and only if $v(\{i\}) = 1$.
2. Dictator: It is easy to see that if a dictator exists in a simple game, it is unique. It follows from the definition of a dictator that a player i is a dictator in a simple game if $v(\{i\}) = 1$ and $v(N \setminus \{i\}) = 0$.

□

3.5 Complexity of desirability ordering

A *desirability ordering* on linear games is any ordering of players such that

$$1 \geq_D 2 \geq_D \ \ldots \ \geq_D n.$$

A *strict* desirability ordering is any ordering on players: $1 \ \circ \ 2 \ \circ \ \ldots \ \circ \ n$ where $\circ$ is either $\sim_D$ or $\succ_D$.

Proposition 3.16. *For a WVG:*

1. *A desirability ordering of players can be computed in polynomial time.*
2. *It is NP-hard to compute a strict desirability ordering of players.*

Proof. WVGs are linear games with a desirability ordering. For (1), it is easy to see that one desirability ordering of players in a WVG is the ordering of the weights. When $w_i = w_j$, then we know that $i \sim j$. Moreover, if $w_i > w_j$, then we know that i is at least as desirable as j, that is $i \geq j$. For (2), the result immediately follows from the result by Matsui and Matsui [151] where they prove that it is NP-hard to check whether two players are symmetric. □

Let v be a MWVG of m WVGs on n players. It is easy to see that if there is an ordering of players such that such that $w_1^t \geq w_2^t \geq \ldots \geq w_n^t$ for all t, then v is linear. However, if an ordering like this does not exist, this does not imply that the game is not linear. For example, it is easy to give such a game with 3 players and by Proposition 3.7, this must be a linear game. Whereas simple games with 3 players are linear, it is easy to construct a 4 player non-linear MWVG:

Example 3.17. In game $v = [10; 10, 9, 1, 0] \wedge [10; 9, 10, 0, 1]$, players 1 and 2 are incomparable.

Proposition 3.18. *It is NP-hard to verify whether a MWVG is linear or not.*

Proof. We prove this by a reduction from an instance of the classical NP-hard PARTITION problem.

Given an instance of PARTITION $\{a_1, \ldots, a_k\}$, we may as well assume that $\sum_{i=1}^{k} a_i$ is an even integer, $2t$ say. We can transform the instance into the multiple weighted voting $v = v_1 \wedge v_2$ where $v_1 = [q; 100a_1, \ldots, 100a_k, 10, 9, 1, 0]$ and $v_2 = [q; 100a_1, \ldots, 100a_k, 9, 10, 0, 1]$ for $q = 10 + 100t$ and $k+4$ is the number of players.

If A is a 'no' instance of PARTITION, then we see that a subset of weights $\{100a_1, \ldots, 100a_k\}$ cannot sum to $100t$. Since, each weight in $\{100a_1, \ldots, 100a_k\}$ is a multiple of 100 any subset of $\{100a_1, \ldots, 100a_k\}$ is a multiple of 100. Thus if $S \subseteq \{100a_1, \ldots, 100a_k\}$ is losing in v, the inclusion of the last four players which contribute a total of at most 20 to v_1 and v_2 cannot make S winning. This implies that players $k+1$, $k+2$, $k+3$, and $k+4$ are not critical for any coalition. Since players $1, \ldots, k$ have the same desirability ordering in both v_1 and v_2, v is linear.

Now let us assume that A is a 'yes' instance of PARTITION with a partition (P_1, P_2). In that case players $k+1$, $k+2$, $k+3$, and $k+4$ are critical for certain coalitions. We see that $v(\{k+1\} \cup (\{k+4\} \cup P_1)) = 1$, $v(\{k+2\} \cup (\{k+4\} \cup P_1)) = 0$, $v(\{k+1\} \cup (\{k+3\} \cup P_1)) = 0$ and $v(\{k+2\} \cup (\{k+3\} \cup P_1)) = 1$. Therefore, players $k+1$ and $k+2$ are not comparable and v is not linear. □

Proposition 3.19. *For a simple game $v = (N, W^m)$, it can be verified in $O(n^2 + n|W^m|)$ time if v is linear or not.*

Proof. Monotone simple games have a direct correspondence with positive boolean functions where minimal true vectors correspond to minimal winning coalitions and linear simple games corresponds to 2-monotonic boolean functions. Makino [147] proved that for a positive boolean function on n variables represented by the set of all minimal true vectors $minT(f)$, it can be checked in time $O(n.|minT(f)|)$ whether the function is *2-monotonic* (linear) or not. The result was an improvement on the algorithm by Peled and Simone [174]. Makino's algorithm (which we will refer to as IS-LINEAR) takes $minT(f)$ as input and outputs 'yes' if f is 2-monotonic and 'no' otherwise. Then it follows that it can be verified in $O(n^2 + n(|W^m|))$ whether a simple game $v = (N, W^m)$ is linear or not.
□

Corollary 3.20. *For a simple game $v = (N, W)$, it can be verified in polynomial time if v is linear or not.*

Proof. We showed earlier that (N, W) can be transformed into (N, W^m) in polynomial time. After that we can use Proposition 3.19 to verify whether the game is linear or not. □

Muroga [154] cites Winder [216] for a result concerning comparison between boolean variables and their incidence in prime implicants of a boolean function. Hilliard [105] points out that this result can be used to check the desirability relation between players in WVG-Realizable simple games. We generalize Winder's result by proving both sides of the implications and extend Hilliard's observation to that of linear simple games.

Proposition 3.21. *Let $v = (N, W^m)$ be a linear simple game and let $d_{k,i} = |\{S : i \in S, S \in W^m, |S| = k\}|$. Then for two players i and j,*

1. *$i \sim_D j$ if and only if $d_{k,i} = d_{k,j}$ for $k = 1, \dots n$.*
2. *$i \succ_D j$ if and only if for the smallest k where $d_{k,i} \neq d_{k,j}$, $d_{k,i} > d_{k,j}$.*

Proof. 1. (⇒) Let us assume $i \sim_D j$. Then by definition, $v(S \cup \{j\}) = 1 \Leftrightarrow v(S \cup \{i\}) = 1$ for all $S \subseteq N \setminus \{i, j\}$. So $S \cup \{i\} \in W^m$ if and only if $S \cup \{j\} \in W^m$. Therefore, $d_{k,i} = d_{k,j}$ for $k = 1, \dots n$.
(⇐) Let us assume that $i \nsim_D j$. Since v is linear, i and j are comparable. Without loss of generality, we assume that $i \succ_D j$. Then there exists a coalition $S \setminus \{i, j\}$ such that $v(S \cup \{i\}) = 1$ and $v(S \cup \{j\}) = 0$ and suppose $|S| = k - 1$. If $S \cup \{i\} \in W^m$, then $d_{k,i} > d_{k,j}$. If $S \cup \{i\} \notin W^m$ then there exists $S' \subset S$ such that $S' \cup \{i\} \in W^m$. Thus there exists $k' < k$ such that $d_{k',i} > d_{k',j}$.

2. (⇒) Let us assume that $i \succ_D j$ and let k' be the smallest integer where $d_{k',i} \neq d_{k',j}$. If $d_{k',i} < d_{k',j}$, then there exists a coalition S such that $S \cup \{j\} \in W^m$, $S \cup \{i\} \notin W^m$ and $|S| = k' - 1$. $S \cup \{i\} \notin W^m$ in only two cases. The first possibility is that $v(S \cup \{i\}) = 0$, but this is not true since $i \succ_D j$. The second possibility is that $v(S \cup \{i\}) = 1$ but $S \cup \{i\}$ in not a minimal winning coalition. Then, there exists a coalition $S' \subset S$ such that $S' \cup \{i\} \in W^m$. But that would

mean that $v(S' \cup \{i\}) = 1$ and $v(S' \cup \{j\}) = 0$. This also leads to a contradiction since k' is the smallest integer where $d_{k',i} \neq d_{k',j}$.
($\Leftarrow$) Let us assume that for the smallest k where $d_{k,i} \neq d_{k,j}$, $d_{k,i} > d_{k,j}$. This means there exists a coalition S such that $S \cup \{i\} \in W^m$, $S \cup \{j\} \notin W^m$ and $|S| = k-1$. This means that either $v(S \cup \{j\}) = 0$ or $S \cup \{j\}$ is winning coalition but not a minimal winning coalition. If $v(S \cup \{j\}) = 0$, that means $i \succ_D j$. If $S \cup \{i\}$ is winning coalition but not a minimal winning coalition, then there exists a coalition $S' \subset S$ such that $S' \cup \{j\} \in W^m$. Then $d_{k',j} > d_{k',i}$ for some $k' < k$. This leads to a contradiction. $\square$

We can use this theorem and Makino's algorithm [147] to make an algorithm which takes as input a simple game (N, W^m) and returns NO if the game is not linear and returns the strict desirability ordering otherwise. Note that Makino's algorithm IS-LINEAR [147] is used in a black box manner. The algorithm implicitly does compute the permutation of players for which the game is linear. However, it does not divide the players into desirability classes.

Algorithm 1 takes as input the set of players and the set of MWCs. If the game is not linear, a $|W^m| \times n$ matrix D is constructed with entries $d_{k,i} = |\{S : i \in S, S \in W^m, |S| = k\}|$. The set of players N is set to class_1 which needs to be divided and ordered into desirability classes which look for example like $\mathsf{class}_{1.1}$, $\mathsf{class}_{1.2}$, $\mathsf{class}_{2.1.1}$, $\mathsf{class}_{2.2.2}\ldots$. The function $\mathsf{classify}(\mathsf{class}_1, D, 1)$ is called where $\mathsf{classify}$ takes as input the set of players $\mathsf{class}_{\mathsf{index}}$, matrix D and k (which is the size of the MWCs being considered). If k is $|W^m|+1$ or $|\mathsf{class}_{\mathsf{index}}| = 1$, then the set of players $\mathsf{class}_{\mathsf{index}}$ is returned. Otherwise, the set $\mathsf{class}_{\mathsf{index}}$ is divided into subclasses based on matrix D and k. These new subclasses are named with the use of further sub-indexing. The process of further refining the subclasses of players is repeated by incrementing k by 1 and recursively calling $\mathsf{classify}$. The process stops when the set of $\mathsf{class}_{\mathsf{index}}$ has been partitioned into desirability equivalence classes with more desirable classes ordered first.

Algorithm 1 Strict-desirability-ordering-of-simple-game

Input: Simple game $v = (N, W^m)$ where $N = \{1, \dots, n\}$ and $W^m(v) = \{S_1, \dots, S_{|W^m|}\}$.

Output: NO if v is not linear. Otherwise output desirability equivalence classes starting from most desirable, if v is linear.

```
X = IS-LINEAR(W^m)
if X = NO then
    return NO
else
    Initialize an |W^m| × n matrix D with entries d_{i,j} = 0 for all i and j in N
    for i = 1 to |W^m| do
        for each player x in S_i do
            d_{|S_i|,x} ← d_{|S_i|,x} + 1
        end for
    end for
    class_1 ← N
    return classify(class_1, D, 1)
end if
```

Algorithm 2 classify

Input: set of integers $\mathsf{class}_{\mathsf{index}}$, $|W^m| \times n$ matrix D, integer k.

Output: subclasses.

if $k = |W^m| + 1$ or $|\mathsf{class}_{\mathsf{index}}| = 1$ **then**
  return $\mathsf{class}_{\mathsf{index}}$
end if
$s \leftarrow |\mathsf{class}_{\mathsf{index}}|$
$\mathsf{mergeSort}(\mathsf{class}_{\mathsf{index}})$ in descending order such that $i > j$ if $d_{k,i} > d_{k,j}$.
for $i = 2$ to s **do**
  $\mathsf{subindex} \leftarrow 1$; $\mathsf{class}_{\mathsf{index.subindex}} \leftarrow \mathsf{class}_{\mathsf{index}}[1]$
  if $d_{k,\mathsf{class}_{\mathsf{index}}[i]} = d_{k,\mathsf{class}_{\mathsf{index}}[i-1]}$ **then**
    $\mathsf{class}_{\mathsf{index.subindex}} \leftarrow \mathsf{class}_{\mathsf{index.subindex}} \cup \mathsf{class}_{\mathsf{index}}[i]$
  else if $d_{k,\mathsf{class}_{\mathsf{index}}[i]} < d_{k,\mathsf{class}_{\mathsf{index}}[i-1]}$ **then**
    $\mathsf{subindex} \leftarrow \mathsf{subindex} + 1$
    $\mathsf{class}_{\mathsf{index.subindex}} \leftarrow \{\mathsf{class}_{\mathsf{index}}[i]\}$
  end if
end for
$\mathsf{Returnset} \leftarrow \emptyset$
$A \leftarrow \emptyset$
for $j = 1$ to $\mathsf{subindex}$ **do**
  $A \leftarrow \mathsf{classify}(\mathsf{class}_{\mathsf{index.j}}, D, k + 1)$
  $\mathsf{Returnset} \leftarrow A \cup \mathsf{Returnset}$
end for
return $\mathsf{Returnset}$

Proposition 3.22. *The time complexity of Algorithm 1 is* $\mathcal{O}(n.|W^m| + n^2 log(n))$

Proof. The time complexity of IS-LINEAR and computing matrix D is $\mathcal{O}(n^2 + n.|W^m|)$. For each iteration, sorting of sublists requires at most $\mathcal{O}(nlog(n))$ time. There are at most n loops. Therefore the total time complexity is $\mathcal{O}(n^2 + n.|W^m|) + \mathcal{O}(n^2 log(n)) = \mathcal{O}(n.|W^m| + n^2 log(n))$. □

Corollary 3.23. *The strict desirability ordering of players in a linear simple game* $v = (N, W)$ *can be computed in polynomial time.*

Proof. The proof follows from Algorithm 1. Moreover, we know that the set of all winning coalitions can be transformed into a set of minimal winning coalitions in polynomial time. □

3.6 Power indices and Chow parameters

In [151], Matsui and Matsui prove that it is NP-hard to compute the Banzhaf index, Shapley-Shubik index and Deegan-Packel index of a player. We can use a similar technique to also prove that it is NP-hard to compute the Holler index of players in a WVG. This follows from the fact that it is NP-hard to decide whether a player is dummy or not. Prasad and Kelly [179] and Deng and Papadimitriou [60] proved that for WVGs, computing the Banzhaf values and Shapley-Shubik values is #P-parsimonious-complete and #P-metric-complete respectively. (For details on #P-completeness and associated reductions, see [76]). Unless specified, reductions considered with #P-completeness will be Cook reductions (or polynomial-time Turing reductions).

What we see is that although it is NP-hard to compute the Holler index and Deegan-Packel index of players in a WVG, the Holler index and Deegan-Packel index of players in a simple game represented by its MWCs can be computed in linear time:

Proposition 3.24. *For a simple game* (N, W^m)*, the Holler index and Deegan-Packel index for all players can be computed in linear time.*

Proof. We examine each of the cases separately:

- Initialize M_i to zero. Then for each $S \in W^m$, if $i \in S$, increment M_i by one.
- Initialize d_i to zero. Then for each $S \in W^m$, if $i \in S$, increment d_i by $\frac{1}{|S|}$. Then $D_i = \frac{d_i}{|W^m|}$.

□

Proposition 3.25. *For a simple game* $v = (N, W)$*, the Banzhaf index, Shapley Shubik index, Holler index and Deegan-Packel index can be computed in polynomial time.*

Proof. The proof follows from the definitions. We examine each of the cases separately:

- Holler index: Transform W into W^m and then compute the Holler indices.

- Deegan-Packel: Transform W into W^m and then compute the Deegan-Packel indices.
- Banzhaf index: Initialize Banzhaf values of all players to zero. For each $S \in W$, check if the removal of a player results in S becoming losing (not a member of W). In that case increment the Banzhaf value of that player by one.
- Shapley-Shubik index: Initialize Shapley values of all players to zero. For each $S \in W$, check if the removal of a player results in S becoming losing (not a member of W). In that case increment the Shapley value of the player by $(|S|-1)!(n-|S|)!$.

The time complexity for all cases is polynomial in the order of the input. □

For a simple game (N, W^m), listing W the winning coalitions may take time exponential in the number of players. For example, let there be only one minimal winning coalition S which contains players $1, \ldots, \lceil n/2 \rceil$. Then the number of winning coalitions to list is exponential in the number of players. Moreover, if $|W^m| > 1$, minimal winning coalitions can have common supersets. It is shown below that for a simple game (N, W^m), even counting the total number of winning coalitions is #P-complete. Moreover, whereas it is possible in polynomial time to check if a player has zero voting power (a dummy) as seen in Proposition 3.13 or whether it has voting power 1 (dictator) as seen in Proposition 3.15, it is #P-complete to find the actual Banzhaf or Shapley-Shubik index of the player.

Proposition 3.26. *For a simple game $v = (N, W^m)$, the problem of computing the Banzhaf values of players is #P-complete.*

Proof. The problem is clearly in #P. We prove the #P-hardness of the problem by providing a reduction from the problem of computing $|W|$. Ball and Provan [30] proved that computing $|W|$ is #P-complete. Their proof is in the context of reliability functions so we first give the proof in terms of simple games. It is known [181] that counting the number of vertex covers is #P-complete (a vertex cover in a graph $G = (V, E)$ is a subset C of V such that every edge in E has at least one endpoint in C). Now take a simple game $v = (N, W^m)$ where for any $S \in W^m$, $|S| = 2$. Game v has a one-to-one correspondence with a graph $G = (V, E)$ such that $N = V$ and $\{i, j\} \in W^m$ if and only if $\{i, j\} \in E(G)$. For a losing coalition, the set of players that do not belong to the losing coalition must correspond to a vertex cover of G

In that case the total number of losing coalitions in v is equal to the number of vertex covers of G. Therefore the total number of winning coalitions is equal to 2^n-(number of vertex covers of G) and computing $|W|$ is #P-complete.

Now we take a game $v = (N, W^m)$ and convert it into another game $v' = (N \cup \{x\}, W^m(v'))$ where $W^m(v') = \{S \cup \{x\} | S \in W^m(v)\}$. In that case computing $|W(v)|$ is equivalent to computing the Banzhaf value of player x in game v'. Therefore, computing Banzhaf values of players in games represented by MWCs is #P-hard. $\square$

It follows from the proof that computing the *power of collectivity to act*$(\frac{|W|}{2^n})$ and the Chow parameters for a simple game (N, W^m) is #P-complete. Goldberg remarks in the conclusion of [100] that computing the Chow parameters of a WVG is #P-complete. It is easy to prove this. We remember that the Chow parameter for player i is dented by $|W_i|$ or ω_i. The problem of computing $|W|$ and $|W_i|$ for any player i is in #P since a winning coalition can be verified in polynomial time. It is easy to reduce in polynomial time the counting version of the SUBSET-SUM problem to counting the number of winning coalitions. Moreover, for any WVG $v = [q; w_1, \ldots, w_n]$, $|W(v)|$ is equal to $|W_x(v')|$ where v' is $[q; w_1, \ldots, w_n, 0]$. Therefore computing $|W_i|$ and $|W|$ for a WVG is #P-complete.

We remember that η_i is the Banzhaf value of a game, ω_i is the number of winning coalitions which include player i and ω denotes the total number of winning coalitions.

Lemma 3.27.

$$\eta_i = 2\omega_i - \omega$$

Proof. Let ω_{i^-} be the number of those winning coalitions which do not include i. Then, the total number of winning coalitions is $\omega = \omega_i + \omega_{i^-}$. The number of winning coalitions which include i is equal to the number of losing coalitions for which i is critical plus the number of coalitions which do not include i but are winning. From this, we know that $\omega_i = \omega_{i^-} + \eta_i$. Then, $\omega = \omega_{i^-} + \eta_i + \omega_{i^-} = 2\omega_{i^-} + \eta_i$. Then, it follows that $\eta_i = \omega_i - \omega_{i^-} = \omega_i - (\omega - \omega_i) = 2\omega_i - \omega$. □

Computing the Shapley-Shubik indices of players in (N, W^m) is also #P-complete.

Proposition 3.28. *For a simple game $G = (N, W^m)$, the problem of computing the Shapley-Shubik indices of players is #P-complete.*

Proof. Computing Shapley-Shubik indices of any cooperative game is clearly in #P. We show that computing the Shapley-Shubik indices is at least as hard as computing the total number of winning coalitions. Let N_i be the number of winning coalitions of a certain size i.

We get a new game G_0 by doing the following.

$$G_0 = (N \cup \{x\}, \{S \cup \{x\} : S \in W^m(G)\}).$$

Then, by the definition, Shapley-Shubik value of player x in G_0 is

$$\kappa_x(G_0) = \sum_{r=0}^{n} r! N_r (n-r)! = \sum_{r=0}^{n} r! N'_r,$$

where we write N'_r for $N_r(n-r)!$, for all r. Similarly we can construct G_i by adding i extra vetoers to G_0. If coalition S is winning in G, then S will require the inclusion of all the new i players plus player x to be winning in G_i. Therefore,

$$\sum_{r=0}^{n} (r+i)! N'_r = \kappa_x(G_i). \tag{3.1}$$

For $i = 0$ to $i = n$, we get an equation of the form of Equation (3.1) for each G_i. The coefficients of the left-hand side of the set of equations can be represented by the matrix A which is an $(n+1) \times (n+1)$ matrix where $A_{ij} = (i + j - 2)!$. The set of equations is independent because A has a non-zero determinant of $(1!2! \cdots n!)^2$ (this follows from Theorem 1.1 [20]). If there is a polynomial time algorithm to compute the Shapley-Shubik index of each edge in a simple graph, then we can compute the right-hand side of each equation corresponding to G_i.

The biggest possible number in the equation is less than $(2n)!$ and can be represented efficiently. This follows from the fact that $n! \leq n^n$ and hence to represent $(2n)!$, one will use at most $\log_2((2n)^{2n}) = 2n(1 + \log_2 n) \leq 3n \log_2 n$ bits.

We can use Gaussian elimination to solve the set of linear equation in $O(n^3)$ time. Moreover, each number that occurs in the algorithm can also be stored in a number of bits quadratic of the input size (Theorem 4.10 [123]). If there is an algorithm polynomial in the number of edges to compute the Shapley-Shubik index of all edges in the graph, then each N_i can be computed in polynomial time. □

A representation of a simple game is considered *reasonable* if, for a simple game (N, v), the new game $(N \cup \{x\}, v')$ where $v(S) = 1$ if and only if $v'(S \cup \{x\}) = 1$, can also be represented with only a polynomial blowup. The following theorem characterizes the relation between the computational complexity of the Banzhaf value and the Shapley-Shubik indices.

Theorem 3.29. *For a simple game with a reasonable representation, if computing the Banzhaf values is #P-complete, then computing the Shapley-Shubik indices is #P-complete.*

Proof. Assume that computing the Banzhaf values is #P-complete. The proof technique in Proposition 3.6 can be used to show that for any reasonable representation of the simple game, a polynomial time algorithm to compute the Shapley-Shubik indices implies a polynomial time algorithm to compute the Banzhaf indices.

It is first proved that if computing the Banzhaf values of players in a reasonable representation is #P-complete, then computing ω, the total number of winning coalitions is #P-complete. Assume that there is an oracle which can compute ω of simple game (N, v) in polynomial time. Since $\eta_i(N, v) = \omega(N, v) - \omega(N \setminus \{i\}, v)$, $\eta_i(N, v)$ can be computed in polynomial time. Therefore, computing $\omega(N, v)$ is #P-complete. However, we saw that if there is an oracle to compute the Shapley-Shubik indices of a simple game (in a reasonable representation) in polynomial time, then N_is can be computed in polynomial time. Since $\sum_{j=1}^{n} N_i = \omega$, ω can be computed in polynomial time. □

As a corollary, we strengthen the complexity results for two other network games which are representations of simple games and answer open questions about the complexity of a host of skill based games:

Corollary 3.30. *Computing Shapley value is #P-complete for*

1. *Threshold Network Flow Games [26]*
2. *Vertex Connectivity Games [27]*
3. *STSG (Single Task Skill Game), TCSG (Task Count Skill Game), WTSG (Weighted Task Skill Game), TCSG-T (Task Count Skill Game with thresholds) and WTSG-T (Weighted Task Skill Game with thresholds) [25]*

Proof. For the given games, computing Banzhaf values is #P-complete. It is easy to see that the games Threshold Network Flow Games, Vertex Connectivity Games, STSG (Single Task Skill Game), TCSG-T (Task Count Skill Game with thresholds) and WTSG-T (Weighted Task Skill Game with thresholds) are simple games with reasonable representations. Also, TCSG (Task Count Skill Game) and WTSG (Weighted Task Skill Game) are generalizations of the STSG (Single Task Skill Game). □

As we will see later, the proof technique of Theorem 3.29 will be used in the proof of Proposition 12.5.

3.7 Conclusion

A summary of results has been listed in Table 3.1. A question mark indicates that the specified problem is still open. It is conjectured that it is NP-hard to compute Banzhaf indices for a simple game represented by (N, W^m). It is found that although WVG, MWVG and even (N, W^m) are relatively compact representations of simple games, some of the important information encoded in these representations can apparently only be accessed by unraveling these representations. There is a need for a greater examination of transformations of simple games into compact representations.

Table 3.1. Complexity of comparing players

	(N,W)	(N,W^m)	WVG	MWVG
IDENTIFY-DUMMIES	P	linear	NP-hard	NP-hard
IDENTIFY-VETOERS	linear	linear	linear	linear
IDENTIFY-PASSERS	linear	linear	linear	linear
IDENTIFY-DICTATOR	linear	linear	linear	linear
CHOW PARAMETERS	linear	#P-complete	#P-complete	#P-complete
IS-LINEAR	P	P	(Always linear)	NP-hard
DESIRABILITY-ORDERING	P	P	P	NP-hard
STRICT-DESIRABILITY	P	P	NP-hard	NP-hard
BANZHAF-VALUES	P	#P-complete	#P-complete	#P-complete
BANZHAF-INDICES	P	?	NP-hard	NP-hard
SHAPLEY-SHUBIK-VALUES	P	#P-complete	#P-complete	#P-complete
SHAPLEY-SHUBIK-INDICES	P	#P-complete	#P-complete	#P-complete
HOLLER-INDICES	P	linear	NP-hard	NP-hard
DEEGAN-PACKEL-INDICES	P	linear	NP-hard	NP-hard

4

Classification of computationally tractable weighted voting games

In order to distinguish what is most simple from what is complex, and to deal with things in an orderly way, what we must do, whenever we have a series in which we have directly deduced a number of truths one from another, is to observe which one is most simple, and how far all the others are removed from this—whether more, or less, or equally.
- Descartes (Rule VI, Rules for the Direction of the Mind)

Abstract It is well known that computing Banzhaf indices in a weighted voting game is #P-complete. We give a comprehensive classification of weighted voting games which can be solved in polynomial time. Among other results, we provide a polynomial ($O(k(\frac{n}{k})^k)$) algorithm to compute the Banzhaf indices in weighted voting games in which the number of weight values is bounded by k. Computational results concerning weighted voting games with special distributions of weights are also presented.

4.1 Introduction

4.1.1 Motivation and outline

The Banzhaf index is considered the most suitable power index by voting power theorists ([132] and [82]). As mentioned before in Chapter 3, the computational complexity of computing Banzhaf indices in WVGs is well studied. Prasad and

Kelly [179] show that the problem of computing the Banzhaf values of players is #P-complete. It is even NP-hard to identify a player with zero voting power or two players with same Banzhaf indices [151].

Klinz and Woeginger [121] devised the fastest exact algorithm to compute Banzhaf indices in a WVG. In the algorithm, they applied a partitioning approach that dates back to Horowitz and Sahni [108]. However the complexity of the algorithm is still $O(n^2 2^{\frac{n}{2}})$. In this chapter, we restrict our analysis to exact computation of Banzhaf indices instead of examining approximate solutions. We show that although computing Banzhaf indices of WVGs is a hard problem in general, it is easy for various classes of WVGs, e.g., for WVGs with a bounded number of weight values, an important sub-class of WVGs.

The outline of the chapter is as following. Section 4.2 identifies WVGs in which Banzhaf indices can be computed in constant time. In Section 4.3, we examine WVGs with a bounded number of weight values, and provide algorithms to compute the Banzhaf indices. Section 4.4 examines WVGs with special weight distributions. Section 4.5 considers WVGs with integer weights. Section 4.6 provides a survey of approximate approaches to computing power indices in WVGs. We conclude with some open problems in the final section.

Generally, $\frac{1}{2}\sum_{1\le i\le n} w_i < q \le \sum_{1\le i\le n} w_i$ so that there can be no two disjoint winning coalitions. Such weighted voting games are *proper*.

The problem of computing the Banzhaf indices of a WVG can be defined formally as following:

Name: BI-WVG

Instance: WVG, $v = [q; w_1, ..., w_n]$

Question: What are the Banzhaf indices of the players?

Here, we will suppose that arithmetic operations on $O(n)$-digit numbers can be done in constant time.

4.2 Extreme cases

If the WVG v is $[q; \underbrace{u, u, \ldots, u}_{n}]$, then the Banzhaf indices β_1, ... , β_n are equal to $1/n$. The Banzhaf indices can be found in constant time, and the following theorem gives the actual number of swings for each player.

Theorem 4.1. *In a WVG with n equal weights, u, each player is critical in* $\binom{n-1}{\lceil q/u \rceil - 1}$ *coalitions. Moreover, the total number of winning coalitions w is* $\sum_{i=\lceil q/u \rceil}^{n} \binom{n}{i}$.

Proof. The minimum number of players needed to form a winning coalition is $\lceil q/u \rceil$. A player is critical in a coalition if there are exactly $\lceil q/u \rceil - 1$ other players in the coalition. There are $\binom{n-1}{\lceil q/u \rceil - 1}$ such coalitions. There are $\binom{n}{i}$ coalitions of size i and such a coalition is winning if $i \geq \lceil q/u \rceil$. □

Also, in a WVG with n equal weights u, the probabilistic Banzhaf index of each player is then $\binom{n-1}{\lceil q/u \rceil - 1}/2^{n-1}$. We can also compute Coleman's power of the collectivity to act, A, which is equal to $\frac{w}{2^n}$.

A *dictator* is a player who is present in every winning coalition and absent from every losing coalition. This means that the player 1 with the biggest weight is a dictator if and only if $w_1 \geq q$ and $\sum_{2 \leq i \leq n} w_i < q$. In that case, $\beta_1 = 1$ and $\beta_i = 0$ for all $i > 1$.

If $0 < q \leq w_n$ then the only minimal winning coalitions are all the singleton coalitions. So there are n minimal winning coalitions and every player is critical in one coalition. Thus, for all $i, \beta_i = 1/n$ and the Banzhaf indices can be found in constant time (i.e., $O(1)$). Moreover, the probabilistic Banzhaf index $\beta_i' = 1/2^{n-1}$ for all i, and Coleman's power of collectivity to act $A = \frac{2^n - 1}{2^n}$

If $q \geq \sum_{1 \leq i \leq n} w_i - w_n$, then the only minimal winning coalition is $\{1, 2, \ldots, n\}$ and it becomes losing if any player gets out of the coalition. Thus the weighted voting game acts like the unanimity game. Then for all $i, \beta_i = 1/n$. The Banzhaf indices can be found in constant time (i.e., $O(1)$). Moreover, for all $i, \beta_i' = 1/2^{n-1}$ and $A = 1/2^n$.

4.3 Bounded number of weight values

In this section we estimate the time complexity of several algorithms. We start off with the case when all weights except one are equal and give exact formulas for the Banzhaf indices. We then use this as a warm up exercise to consider more general cases where there are 2 weight values and then k weight values.

4.3.1 All weights except one are equal

We start off with the case when all weights except one are equal.

Theorem 4.2. *Let v be a WVG, $[q; w_a, w_b, ..., w_b]$, where there is w_a and m weights of value w_b, where $w_b < q$. Let x be $\lceil \frac{q-w_a}{w_b} \rceil$ and $y = \lceil q/w_b \rceil$. Then the total number of coalitions in which a player with weight w_b is critical is $\binom{m-1}{y-1} + \binom{m-1}{x}$. Moreover, the number of coalitions in which the player with weight w_a is critical is $\sum_{i=x}^{\mathrm{Min}(y-1,m)} \binom{m}{i}$.*

Proof. A player with weight w_b is critical in 2 cases:

1. *It makes a winning coalition with other players with weight w_b only.* Let y be the minimum number of players with weight w_b which form a winning coalition by themselves. Thus $y = \lceil q/w_b \rceil$. The number of such coalitions in which a player with weight w_b can be critical is $\binom{m-1}{y-1}$.
2. *It makes a winning coalition with the player with weight w_a and none or some players with weight w_b.* Let x be the minimum number of players with weight w_b which can form a winning coalition with the inclusion of the player with weight w_a. Thus $x = \lceil \frac{q-w_a}{w_b} \rceil$. Then, the number of such coalitions in which a player with weight w_b can be critical is $\binom{m-1}{x}$.

The total number of swings for a player with weight w_b is thus $\binom{m-1}{y-1} + \binom{m-1}{x}$.

The player with weight w_a is critical if it forms a winning a coalition with some players with weight w_b but the coalition becomes losing with its exclusion. The player with weight w_a can prove critical in coalition with varying number of players with weight w_b. The maximum number of players with weight w_b with which it forms a winning coalition and is also critical is $y-1$ in case $y \leq m$ and m

in case $y > m$. Therefore the total number of coalitions in which the player with weight w_a is critical is $\sum_{i=x}^{\mathsf{Min}(y-1,m)} \binom{m}{i}$. □

4.3.2 Only two different weight values

Unlike Theorem 4.2, we do not give a short formula for the Banzhaf values in the next theorem. However Theorem 4.3 considers a more general case than Theorem 4.2. As we shall we later Theorem 4.2 provides us with an idea to consider the case of k weight values.

Theorem 4.3. *For a WVG with n players and only two weight values, the Banzhaf indices and numbers of swings can be computed in $O(n^2)$ time.*

Proof. We look at a WVG, $v = [q; w_a, ...w_a, w_b, ...w_b]$, where there are n_a players with weight w_a and n_b players with weight w_b. We analyse the situation when a player with weight w_a proves to be critical in a coalition which has i other players with weight w_a and the rest with weight w_b. Then the minimum number of players with weight w_b required is $\lceil \frac{q-(i+1)w_a}{w_b} \rceil$. Moreover the maximum number of players with w_b is $\lceil \frac{q-iw_a}{w_b} \rceil - 1$. Therefore j, the number of players with weight w_b, satisfies the following inequality: $x_1(i) = \lceil \frac{q-(i+1)w_a}{w_b} \rceil \leq j \leq \mathsf{Min}(\lceil \frac{q-iw_a}{w_b} \rceil - 1, n_b) = x_2(i)$. Let $A_i = \binom{n_a-1}{i}$, and let $B_i = \sum_{j=x_1(i)}^{x_2(i)} \binom{n_b}{j}$. We define, the maximum possible number of extra players with weight a, to be $\mathsf{maxa} = \mathsf{Min}(\lceil q/w_a \rceil - 1, n_a - 1)$. Then the total number of swings of the player with weight w_a is $\sum_{i=0}^{\mathsf{maxa}} A_i B_i$. The total number of swings for a player with weight w_b can be computed by a symmetric method. □

We can devise an algorithm (Algorithm 4) from the method outlined in the proof.

Algorithm 3 SwingsFor2ValueWVG

Input: $v = [q; (n_a, w_a), (n_b, w_b)]$.

Output: Total swings of a player with weight w_a.

$\mathsf{swings}_a \leftarrow 0$
$\mathsf{maxa} \leftarrow \mathsf{Min}(\lceil q/w_a \rceil - 1, n_a - 1)$
for $i = 0$ to maxa **do**
  $x_1(i) \leftarrow \lceil \frac{q-(i+1)w_a}{w_b} \rceil$
  $x_2(i) \leftarrow \mathsf{Min}(\lceil \frac{q-i(w_a)}{w_b} \rceil - 1, n_b)$
  $A_i \leftarrow \binom{n_a-1}{i}$
  if $x_1(i) > n_b$ **then**
    $B_i \leftarrow 0$
  else if $x_2(i) < 0$ **then**
    $B_i \leftarrow 0$
  else
    $B_i \leftarrow 0$
    for $j = x_1(i)$ to $x_2(i)$ **do**
      $B_i \leftarrow B_i + \binom{n_b}{j}$
    end for
  end if
  $\mathsf{swings}_a = \mathsf{swings}_a + A_i B_i$
end for
return swings_a

Algorithm 4 BIsFor2ValueWVG

Input: $v = [q; (n_a, w_a), (n_b, w_b)]$.

Output: Banzhaf indices, $\beta = (\beta_a, \beta_b)$.

$\mathsf{swings}_a = \mathsf{SwingsFor2ValueWVG}(v)$
$v' = [q; (n_b, w_b), (n_a, w_a)]$
$\mathsf{swings}_b = \mathsf{SwingsFor2ValueWVG}(v')$
$\mathsf{totalswings} = n_a \mathsf{swings}_a + n_b \mathsf{swings}_b$
$\beta_a = \frac{\mathsf{swings}_a}{\mathsf{totalswings}}$
$\beta_b = \frac{\mathsf{swings}_b}{\mathsf{totalswings}}$
return (β_a, β_b)

The algorithm for 2 weight values serves as warm-up for the general case of k weight values in the next section.

4.3.3 k weight values

Theorem 4.4. *The problem of computing Banzhaf indices of a WVG with k possible values of the weights is solvable in $O(n^k)$.*

Proof. We can represent a WVG v with k weight classes as follows:

$$[q; (n_1, w_1), (n_2, w_1), ..., (n_k, w_k)]$$

where n_i is the number of players with weights w_i for $i = 1, \ldots, k$. Here, we extend Algorithm 4 to Algorithm 6 for k weight classes.

We can write v' as $[q; (1, w_0), (n_1 - 1, w_1), ..., (n_k, w_k)]$ where $w_0 = w_1$. This makes it simpler to write a recursive function to compute the number of swings of player with weight w_0. Let $A_{i_1,i_2,\ldots,i_m}$ be the number of swings for w_0 where there are i_j players with weight w_j in the coalition for $1 \leq j \leq m$.

Then

$$A_{i_1,i_2,\ldots,i_k} = \begin{cases} \binom{n_1-1}{i_1}(\Pi_{j=2}^{k}\binom{n_j}{i_j}) & \text{if } q - w_0 \leq \sum_{j=1}^{k} i_j w_j < q \\ 0 & \text{otherwise.} \end{cases}$$

Now for $1 \leq m \leq k$,

$$A_{i_1,i_2,\ldots,i_{m-1}} = \sum_{i_m} A_{i_1,i_2,\ldots,i_m}.$$

Here the summation is taken over all values of i_m for which the contribution is non-zero. Explicitly, this range is given by

$$\mathsf{Max}(\left\lceil \frac{q - w_0 - \sum_{j=1}^{m-1} i_j w_j - \sum_{j=m+1}^{k} n_j w_j}{w_m} \right\rceil, 0) \leq i_m \leq \mathsf{Min}(\left\lceil \frac{q - \sum_{j=1}^{m-1} i_j w_j}{w_m} \right\rceil - 1, n_m).$$

The total number of swings of the player with weight w_0 is then A_ϵ.

Algorithm 5 SwingsForWVG

Input: $v = [q; (n_1, w_1), (n_1, w_1), \ldots, (n_k, w_k)]$.

Output: Total number of swings, swings_0, of a player with weight w_1.

$w_0 = w_1$
$v' = [q; (1, w_0), (n-1, w_1), ..., (n_k, w_k)]$
$\mathsf{swings}_0 = A_\epsilon$
return swings_0

Algorithm 6 BIsFor-k-ValueWVG

Input: $v = [q; (n_1, w_1), (n_1, w_1), \ldots, (n_k, w_k)]$.

Output: Banzhaf indices, $\beta = (\beta_1, \ldots \beta_k)$.

$\mathsf{swings}_1 = \mathsf{SwingsForWVG}(v)$
$\mathsf{totalswings} \leftarrow 0$
for $i = 2$ to k **do**
  $v = \mathsf{Swap}(v, (n_1, w_1)(n_i, w_i))$
  $\mathsf{swings}_i = \mathsf{SwingsForWVG}(v)$
  $\mathsf{totalswings} \leftarrow \mathsf{totalswings} + n_i \mathsf{swings}_i$
end for
for $i = 1$ to k **do**
  $\beta_i = \frac{\mathsf{swings}_i}{\mathsf{totalswings}}$
end for
return $(\beta_1, \ldots \beta_k)$

We note that the exact computational complexity of BI-WVG for a WVG with k weight values is $O(k(\frac{n}{k})^k)$ where $(\frac{n}{k})^k \geq n_1 \cdots n_k$. None of the algorithms presented for WVGs with bounded weight values extends naturally for multiple weighted voting games.

4.4 Distribution of weights

4.4.1 Geometric sequence of weights, and unbalanced weights

Definition 4.5. *An r-geometric WVG $[q; w_1, ..., w_n]$ is a WVG where $w_i \geq rw_{i+1}$ for $i = 1, ..., n-1$.*

We observe that in a 2-geometric WVG (such as $[q; 2^n, 2^{n-1}, ...,]$), for any target sum of a coalition, we can use a greedy approach, trying to put bigger weights first, to come as close to the target as possible. This greedy approach was first identified by Chakravarty, Goel and Sastry [43] for a broader category of weighted voting games in which weights are *unbalanced*:

Definition 4.6. *An* unbalanced WVG *is a WVG such that, for* $1 \leq j \leq n$, $w_j > w_{j+1} + w_{j+2} ... + w_n$.

Example 4.7. The game $[22; 18, 9, 4, 2, 1]$ is an example of an unbalanced WVG where each weight is greater than the sum of the subsequent weights.

Chakravarty, Goel and Sastry [43] showed that the greedy approach for unbalanced WVG with integer weights can help to compute all Banzhaf indices in $O(n)$. We notice that the same algorithm can be used for an unbalanced WVG with real weights without any modification. In fact it is this property of 'geometric weights' being unbalanced which is the reason that we can find suitable coalitions for target sums so efficiently. We characterise those geometric sequences which give unbalanced WVGs:

Theorem 4.8. *If $r \geq 2$ then every r-geometric WVG is unbalanced.*

Proof. Let v be an r-geometric WVG. We prove by induction that $w_j > w_{j+1} + \ldots + w_n$. This is true for $j = n$. Suppose it is true for all i, $j + 1 \leq i \leq n$. Since v is r-geometric, $w_j \geq 2w_{j+1}$. But, $2w_{j+1} = w_{j+1} + w_{j+1} > w_{j+1} + w_{j+2} + \ldots + w_n$. Therefore v is unbalanced. □

Corollary 4.9. *For an r-geometric WVG v where $r \geq 2$, the Banzhaf indices of players in v can be computed in $O(n)$ time.*

Proof. Since the condition of $r \geq 2$ makes v an unbalanced WVG, then we can use the greedy algorithm from [43] which computes the Banzhaf indices in $O(n)$. □

Definition 4.10. *A WVG is a k-*unbalanced WVG *if, for* $1 \leq j \leq n$, $w_j > w_{j+k} + \cdots + w_n$. *So an unbalanced WVG is '1-unbalanced'.*

Note that an r-geometric WVG is 2-unbalanced when $r \geq \frac{1+\sqrt{5}}{2} \approx 1.61803... = \varphi$, the golden ratio, since then

$$\frac{1}{r^2} + \frac{1}{r^3} + \cdots < \frac{1}{r(r-1)} \leq 1 \text{ since } r(r-1) \geq \varphi(\varphi - 1) = 1.$$

We check whether 2-unbalanced WVGs have properties similar to those of unbalanced WVGs.

Example 4.11. Consider a WVG v with $2m$ players and weights

$$3^{m-1}, 3^{m-1}, \ldots, 3^j, 3^j, \ldots, 3, 3, 1, 1.$$

It is easy to see that $\sum_{i=0}^{j-1} 2 \cdot 3^i < 3^j$, so the game is 2-unbalanced.

In the unbalanced game, for each target coalition sum, there is either one corresponding coalition or none. This does not hold for 2-unbalanced WVGs. In Example 4.12 with target total $1 + 3 + \cdots + 3^{m-1} = \frac{1}{2}(3^m - 1)$, there are exactly 2^m coalitions which give this target, namely those coalitions with exactly one player out of each equal pair.

We prove that even for the class of 2-unbalanced (instead of simply unbalanced WVGs) the problem of computing Banzhaf indices becomes NP-hard.

Theorem 4.12. *BI-WVG is NP-hard for the class of 2-unbalanced WVGs .*

Proof. We will use a reduction from the following NP-hard problem:

Name: *SUBSET SUM*
Instance: $z_1, \ldots, z_m, T \in$ N.
Question: Are there x_js in $\{0, 1\}$ so that $\sum_{j=1}^{m} x_j z_j = T$?

For the reduction from SUBSET SUM, we scale and modify the weights from the WVG v of Example 4.12. For any instance $I = \{z_1, \ldots, z_m, T\}$ of SUBSET SUM, we will define a game v_I with $2m + 1$ players. Let $Z = 1 + \sum_{j=1}^{m} z_j$, and we may assume that $T < Z$. Whereas v had pairs of weights $3^j, 3^j$ for $0 \leq j \leq m - 1$, in v_I there is one "unit player" with weight 1 and $2m$ pairs of players with weights $3^j Z, 3^j Z + z_j$ for $0 \leq j \leq m-1$. The quota for v_I is $\frac{1}{2}(3^m - 1)Z + T + 1$. The unit player has nonzero Banzhaf index if and only if there exists a coalition among the other $2m$ players with weight exactly $\frac{1}{2}(3^m - 1)Z + T$. We will show that to determine this is equivalent to answering the SUBSET SUM instance I, and so even this special case of BI-WVG is NP-hard.

In Example 4.12, it was necessary (and sufficient) for achieving the target total of $\frac{1}{2}(3^m - 1)$ to take exactly one player from each pair. In game v_I, since $\sum_{j=1}^{m} z_j < Z$, this is still a necessary condition for achieving the total of $\frac{1}{2}(3^m - 1)Z + T$, and whether or not there is such a selection achieving the total is exactly the condition of whether there is a subset of the z_js which sums to T. □

4.4.2 Sequential weights

Definition 4.13. *The set of weights* $\{w_1, w_2, ..., w_n\}$ *is* sequential *if*

$$w_n | w_{n-1} | w_{n-2} ... | w_1,$$

i.e. each weight is a multiple of the next weight.

Example 4.14. $[32; 20, 10, 10, 5, 1, 1, 1]$ is an example of a WVG with sequential weights.

Chakravarty, Goel and Sastry [43] show that Banzhaf indices can be computed in $O(n^2)$ time if the weights are sequential and they satisfy an additional dominance condition. The dominance conditions states that a weight in one weight class should be more than the sum of weights of any subsequent weight class.

Definition 4.15. *Let* $d_1 > d_2 > \cdots > d_r$ *be the distinct values of weights* $w_1, \ldots, w_n$ *of a sequential set. Then* $d_k = m_k d_{k+1}$ *where* $m_k > 1$, $\forall k$, $1 \leq k < r$. *Let* $N_k = \{i \mid w_i = d_k\}$ *and* $n_k = |N_k|$. *Then the* dominance condition *holds if* $m_k > n_{k+1}$ $\forall k$, $1 \leq k < r$.

We now define the alternative dominance condition for WVGs.

Definition 4.16. *Let $d_1 > d_2 > \cdots > d_r$ be the distinct values of weights $w_1, \ldots, w_n$ of a sequential set. Let $N_k = \{i | w_i = d_k\}$ and $n_k = |N_k|$. Then the* alternative dominance condition *holds if $\forall j \in N_k$, $1 \leq k < r$, $w_j > \sum\{w_p \mid p \in N_i, i > k\}$.*

We provide an alternative dominance condition for weights which are not necessarily sequential. It is easy to see that a 2-unbalanced WVG does not necessarily satisfy the alternative dominance condition.

Definition 4.17. *Let $d_1 > d_2 > \cdots > d_r$ be the distinct values of weights $w_1, \ldots, w_n$ of a sequential set. Let $N_k = \{i | w_i = d_k\}$ and $n_k = |N_k|$. Then the* alternative dominance condition *holds if $\forall j \in N_k$, $1 \leq k < r$, $w_j > \sum\{w_p \mid p \in N_i, i > k\}$.*

Proposition 4.18. *Suppose a WVG v satisfies the alternative dominance condition. Then for v, BI-WVG has time complexity $O(n^2)$.*

Proof. This follows from Theorem 10 in [43] where the proof is for a sequential WVG which obeys the dominance condition. However we notice that since the argument in the proof can be made for any WVG which satisfies the alternative dominance condition, the proposition holds for v. □

4.5 Integer weights

When all weights are integers, other methods may become applicable.

4.5.1 Moderate sized integer weights

Matsui and Matsui [151] prove that a dynamic programming approach provides a pseudo-polynomial algorithm to compute Banzhaf indices of all players with time complexity $O(n^2 q)$. Since q is less than $\sum_{i \in N} w_i$, the Banzhaf indices can be computed in polynomial time if the weight sizes are moderate.

4.5.2 Polynomial number of coefficients in the generating function of the WVG

A generating function is a formal power series whose coefficients encode information about a sequence. Bilbao et al. [36] observe, for a WVG $v = [q; w_1, \ldots, w_n]$, that if the number of coalitions for which a player i is critical is $b_i = |\{S \subset N : v(S) = 0, v(S \cup \{i\}) = 1\}| = \sum_{k=q-w_i}^{q-1} b^i_k$, where b^i_k is the number of coalitions which do not include i and with total weight k, then the generating functions of the numbers $\{b^i_k\}$ are given by $B_i(x) = \prod_{j=1,j\neq i}^{n}(1+x^{w_j}) = 1+b^i_1 x+b^i_2 x^2+\cdots+b^i_{W-w_i} x^{W-w_i}$. This was first pointed out by Brams and Affuso [39].

Example 4.19. Let $v = [6; 5, 4, 1]$ be a WVG.

- $B_1(x) = (1 + x^4)(1 + x^1) = 1 + x + x^4 + x^5$
  The coalitions in which player 1 is critical are $\{1, 2\}$, $\{1, 3\}$, $\{1, 2, 3\}$. Therefore $\eta_1 = 3$.
- $B_2(x) = (1 + x^5)(1 + x^1) = 1 + x + x^5 + x^6$ The coalition in which player 2 is critical is $\{1, 2\}$. Therefore $\eta_2 = 1$.
- $B_3(x) = (1 + x^5)(1 + x^4) = 1 + x^4 + x^5 + x^9$
  The coalition in which player 3 is critical is $\{1, 3\}$. Therefore $\eta_3 = 1$.

Consequently, $\beta_1 = 3/5$, $\beta_2 = 1/5$ and $\beta_3 = 1/5$.

The generating function method provides an efficient way of computing Banzhaf indices if the voting weights are moderate integers. Bilbao et al. [36] prove that the computational complexity of computing Banzhaf indices by generating functions is $O(n^2C)$ where C is the number of non-zero coefficients in $\prod_{1\leq i\leq n}(1 + x^{w_j})$. We note that C can be bounded by the sum of the weights but the bound is not tight. C can be relatively small even if the weight values are exponential in n. Therefore if a WVG has a generating function in which the number of non-zero terms is polynomial in n, then the computational complexity of computing the Banzhaf indices is in P.

4.6 Approximation approaches

The earliest approximate algorithm for power indices was the Monte Carlo approach by Mann and Shapley [148]. Owen [164, 165] devised an approach using *multilinear extension(MLE)* which provides an exact computation of power indices. The method has exponential time complexity. However the MLE approach can be utilized in large games for approximations using the central limit theorem [135]. Leech [131] succinctly outlines the basic idea of approximations using Owen's MLE approach. Holzman et al. [107] and then Freixas [87] provide bounds for Owen's MLE. Matsui and Matsui [151] in their survey of voting power algorithms also include the Monte Carlo approach. However, they do not focus on the analysis of the errors induced. Fatima et al. [80] propose a variation of Mann and Shapley's [148] algorithm by treating the players' weights instead of the players' numbers of swings as random variables. Bachrach et al. [22] suggest and analyse the randomized approximate algorithm to compute the Banzhaf index and the Shapley-Shubik index with a comprehensive theoretical analysis of the confidence intervals and errors induced.

4.7 Open problems & conclusion

Table 4.1 contains a summary of the algorithms or complexity results for different classes of WVGs. A&P refers to Aziz and Paterson. In this chapter we have classified WVGs for which Banzhaf indices can be computed in polynomial time. It would be interesting to identify further important classes of WVGs which have less than exponential time complexity. The extensive literature on the SUBSET-SUM problem should offer guidance here. It appears an interesting question to analyse the expected number of terms in the generating function for sequential WVGs. Another challenging open problem is to devise an algorithm to compute exactly the Banzhaf indices of a general WVG in time complexity which is less than $O(n^2 2^{\frac{n}{2}})$.

Table 4.1. Complexity of WVG classes

WVG Class	R/Z	Complexity Class	Time	Remarks
General	R/Z	NP-Hard	$O(n^2 1.415^n)$	[121]
Unbalanced	R/Z	P	$O(n)$	[43]
k-Unbalanced($k \geq 2$)	R/Z	NP-Hard		A&P [16]
Sequential with dominance	R/Z	P	$O(n^2)$	[43]
Alternative dominance	R/Z	P	$O(n)$	A&P [16] + [43]
Bounded(k) #(weight values)	R/Z	P	$O(n^k)$	A&P [16]
r-geometric	R/Z	P	$O(n)$	A&P [16] + [43]
Moderate integer weights	Z	P	$O(n^2 q)$	[151]
Moderate GF	Z	P	$O(n^2 C)$	[39], [36]

5

Multiple weighted voting games

In a capitalist democracy, there are essentially two methods by which social choices can be made: voting, typically to make "political decisions", and the market mechanism, typically used to make "economic decisions".
- Kenneth J. Arrow (Social Choice and Individual Values [8])

Abstract We provide mathematical and computational properties of *multiple weighted voting games* which are an extension of weighted voting games. We analyse the structure of multiple weighted voting games and some of their combinatorial properties especially with respect to dictatorship, veto power, dummy players and Banzhaf indices. An illustrative Mathematica program to compute voting power properties of multiple weighted voting games is also provided.

5.1 Introduction

MWVGs are utilized in various situations. The treaty of Nice made the overall voting games of the EU countries a triple majority weighed voting game with certain additional constraints. MWVGs are useful in multi-criteria multi-agent systems. We analyse combinatorial properties of multiple weighted voting games especially with respect to dictatorship, veto power and dummy players. The chapter also outlines algorithmic considerations when computing voting power of players in multiple weighted voting games. In [77], the authors generalize multiple weighted voting games and consider *boolean weighted voting games* which are logical combinations of weighted voting games. However, we restrict out attention to MWVGs which are the most common extension of WVGs.

5.2 MWVGs

In this section, we examine some standard properties of MWVGs which were defined in Definitions 2.4.

5.2.1 Structure

We define S_i as the set of coalitions not including player i. Then S_i can be partitioned into three mutually exclusive sets:

$$S_i = W_i(v) \cup C_i(v) \cup L_i(v)$$

where

- $W_i(v)$ is the set of coalitions not including player i which are winning in the multiple game v
- $L_i(v)$ is the set of coalitions not including player i which are losing in the multiple game v even if player i joins the coalitions.
- $C_i(v)$ is the set of coalitions not including player i which are losing in the multiple game v but winning in v if player i joins the coalitions.

The number of coalitions in which player i is critical in the multiple game v is $\eta_i(v) = |C_i(v)|$. In a MWVG $v = \wedge v_j$, i is critical in a coalition S if

$$(\forall j : (S \in C_i(v_j) \vee S \in W_i(v_j)) \wedge (\exists j : S \in C_i(v_j))$$

We define $W(v)$ as the set of winning coalitions in v and $W(v_i)$ as the set of winning coalitions in v_i. In that case

$$W(v) = W(v_1) \cap W(v_2) ... \cap W(v_m)$$

Similarly if we define $L(v)$ as the set of losing coalitions in v and $L(v_i)$ as the set of losing coalitions in v_i. In that case

$$L(v) = L(v_1) \cup L(v_2) ... \cup L(v_m)$$

5.2.2 Trade robustness

Deineko and Woeginger [57] show that it is NP-hard to verify the dimension of multiple-weighted voting games. In [90], it is pointed out that the dimension of a game is at most the number of maximal losing coalitions. This kind of bound is not very helpful though in estimating the actual dimension of a MWVG.

In Definition 3.6, swap-robust and trade-robust were defined. Taylor and Zwicker [202] proved that a simple game is *trade robust* if and only if it is a WVG. However, MWVGs are not even *swap-robust*:

Example 5.1. Let $(N, v) = (N, v_1 \wedge v_2)$ where $v_1 = [20; 18, 5, 0, 5, 5, 2, 5]$ and $v_2 = [20; 0, 5, 18, 5, 5, 2, 5]$. We see that coalitions $\{1, 3, 6\}$ and $\{2, 4, 5, 7\}$ are winning in v. However if we have a trade so that the resultant coalitions are $\{2, 3, 6\}$ and $\{1, 4, 5, 7\}$, then both coalitions are losing.

5.3 Properties of MWVGs

We define u as the *unanimity* WVG in which a coalition is only winning if it is the grand coalition $N = \{1, 2, ..., n\}$. Every player has veto power in u. We know that in u, all players are critical only in N and therefore have uniform Banzhaf indices. Similarly we define s as the *singleton* weighted voting game in which every coalition is winning except the empty coalition.

Proposition 5.2. *In a MWVG, the constituent* unanimity *WVG acts as a zero and the* singleton *WVG acts as a unit.*

Proof. For a WVG (N, v) and a unanimity WVG (N, u), we notice that for any coalition c to be winning in $(N, v \wedge u)$, it must be winning in both (N, v) and (N, u). Thus the grand coalition is the only winning coalition. So $v \wedge u = u$.

For a WVG (N, v) and a singleton WVG (N, s), we notice that for any coalition c to be winning in $(N, v \wedge u)$ it just has to be non-empty. So $v \wedge s = v$. □

So for $v = v_1 \wedge ... \wedge v_m$, if $\exists j : v_j = u$, then $v = u$. This implies that even if player i is a dictator in one game of the MWVG, it does not mean it is a dictator in the MWVG. Moreover, even if a player is a dummy in all the games apart from the unanimity game v_j, then that player will have Banzhaf power of $1/n$.

Example 5.3. $v = v_1 \wedge v_2$ where $v_1 = [3; 4, 1, 1]$ and $v_2 = [3; 1, 1, 1]$. Player 1 is a dictator in v_1 but it is not a dictator in v.

Proposition 5.4. *For MWVG,* $v = v_1 \wedge ... \wedge v_m$*:*

1. *(*$\forall i$ *: player 1 is a dictator in* v_i*)* $\Longrightarrow$ *player* 1 *is a dictator in* v
2. *(*$\forall j$ *: player i is a dummy in* v_j*)* $\Longrightarrow$ *player i is a dummy in* v
3. *(*$\exists j$ *: player i has veto power in* v_j*)* $\Longrightarrow$ *player i has veto power in* v
4. *(*$\exists j$ *:* v_j *is proper)* $\Longrightarrow$ v *is proper.*

Proof. 1. Let player 1 be a dictator in v_i for all $i = 1, \ldots m$. Thus $\forall i$, $1 \leq i \leq m$, $w_1^i \geq q$ and $\sum_{2 \leq j \leq m} w_j < q$. This means that $\{1\}$ is winning in v and $\{2, ..., n\}$ is losing in v
2. We know that $\forall j$, $C_i(v_j) = \emptyset$. Then by definition, $C_i(v) = \emptyset$.
3. If for some $t = 1, \ldots, m$, we have $N \setminus \{i\} \notin W(v_t)$. Then $N \setminus \{i\} \notin W(v)$.
4. Assume that v_j is proper. This means that if $v_j(S) = 1$ then $v_j(N \setminus S) = 0$. If $v(S) = 1$, then by definition $v_t(S) = 1$, for $1 \leq t \leq m$. Then $v_j(N \setminus S) = 0$ which implies that $v_t(N \setminus S) = 0$. □

Example 5.5. The converses for the previous proposition do not hold:

1. Let $v = v_1 \wedge v_2$ where $v_1 = [4; 5, 1, 1]$ and $v_2 = [2; 5, 1, 1]$. Although player 1 is a dictator in v, it is not a dictator in v_2.
   Moreover, even if there is a WVG v_i in which player 1 does not have the biggest weight, it can still be the dictator: $v = v_1 \wedge v_2$ where $v_1 = [2; 5, 1]$ and $v_2 = [2; 2, 3]$. Player 1 is a dictator in v.

2. Let $v = v_1 \wedge v_2$ where $v_1 = [5;3,2,1]$ and $v_2 = [5;3,2,2]$. Player 3 is a dummy in v but not a dummy in v_2.
   In fact a player can be a dummy in v even if he is not a dummy in any of the games. For example, let $v = v_1 \wedge v_2$ where $v_1 = [7;4,3,3,1]$ and $v_2 = [8;7,3,3,1]$. Player 4 is a not a dummy in v_1 and v_2 but a dummy in v.
3. Let $v = v_1 \wedge v_2$ where $v_1 = [5;3,2,1]$ and $v_2 = [6;5,2,1]$. Player 2 has veto power in v but does not have veto power in v_2
4. Let $v = v_1 \wedge v_2$, where $v_1 = [5;5,2,2,1]$ and $v_2 = [5;1,2,2,5]$. We see that although v_1 and v_2 are not proper, v is proper.

Proposition 5.6. *For MWVG, $v = v_1 \wedge ... \wedge v_m$, if $\exists i$: player 1 is a dictator in v_i, then player 1 has veto power in v.*

Proof. If player 1 is a dictator in v_i, he is in every winning coalition of v_i. Therefore for any coalition c which is winning in v, if the dictator opts out of c, c loses in v_i and therefore loses in v. □

5.4 Analysing MWVGs

Klinz and Woeginger [121] devised the fastest exact algorithm to compute Banzhaf indices in a WVG. In the algorithm, they applied a partitioning approach that dates back to Horowitz and Sahni [108]. The complexity of the algorithm is $O(n^2 2^{\frac{n}{2}})$. This partitioning approach does not directly generalize for MWVGs though.

5.4.1 Generating functions for MWVGs

That was a proof by generating functions, another of the popular tools used by the species Homo sapiens for the proof of identities before the computer era..
- M. Petkovsek, H. S. Wilf and D. Zeilberger, p. 24, [143].

The generating function method provides an efficient way of computing Banzhaf indices if the voting weights are integers [151]. Algaba et al. [2] outline a generating function method to find the Banzhaf indices of players in a multiple weighted majority game. Their algorithm m-BanzhafPower computes the Banzhaf index of the players in $O(\max(m, n^2c))$ time where c is the number of terms of

$$B(x_1, ..., x_m) = \prod_{j=1}^{n}(1 + x_1^{w_j^1}...x_m^{w_j^m}) = \sum_{k_t=0, 1\leq t\leq m}^{w^t(N)} b_{k_1...k_m} x_1{}^{k_1}...x_m{}^{k_m}.$$

The coefficient, $b_{k_1...k_m}$ of each term $x_1{}^{k_1}...x_m{}^{k_m}$ is the number of coalitions such that $w^t(S) = k_t$ for t ranging from 1 to m.

One can make generating functions, $B_i(x_1, ..., x_m)$ for each player i by excluding its influence from the considered coalitions just as in the single WVG case. Therefore $B_i(x_1, ..., x_m) = B(x_1, ..., x_m)/(1 + x_1^{w_i^1}...x_m^{w_i^m})$. These generating functions can be encoded in the form of a coefficient array which gives a clear picture and make the computation of coefficients easier. We present the algorithm due to Algaba et al. with some modifications to avoid extra computations and also to compute the total number of winning coalitions and the set of vetoers.

Algorithm 7 PowerAnalysisOfMWVGs

Input:MWVG: $[q^t; w^t_1, ...w^t_n]$ for $1 \leq t \leq m$.
Output: Number of winning coalitions w, vetoplayerset and Banzhaf indices: $\{w, (\beta_1, \ldots, \beta_n)\}$.

```
vetoplayerset ← ∅
for i = 1 to n do
    isvetoplayer ← false
    for j = 1 to m do
        if w^j(N) − w_i^j < q^j then
            isvetoplayer ← true
        end if
    end for
    if isvetoplayer then
        vetoplayerset ← vetoplayerset ∪ {i}
    end if
end for
B(x_1, ..., x_m) ← ∏_{j=1}^{n} (1 + x_1^{w_j^1} ...x_m^{w_j^m})
coeff = Coeff(B(x_1, ..., x_m))
For k_t from q^t to w^t(N), 1 ≤ t ≤ m,
w ← Sum(coeff[k_1, ..., k_m]) where k_t is the range of weight values in the t-th constituent WVG
for i = 1 to n do
    if i ≠ 1 and w_i^t = w_{i−1}^t for t = 1, ..., m then
        η_i ← η_{i−1}
    else
        B_i(x_1, ..., x_m) ← B(x_1, ..., x_m) / (1 + x_1^{w_i^1} ...x_m^{w_i^m})
        coeff_i = Coeff(B_i(x_1, ..., x_m))
        For k_t from q^t − w_i^t + 1 to w^t(N \ i) + 1, 1 ≤ t ≤ m,
        s_1^i ← Sum(coeff_i[k_1, ..., k_m])
        For k_t from q^t + 1 to w^t(N \ i) + 1, 1 ≤ t ≤ m,
        s_2^i ← Sum(coeff_i[k_1, ..., k_m])
        η_i ← s_1^i − s_2^i
    end if
end for
η ← ∑_{i=1}^{n} η_i
for i = 1 to n do
    β_i ← η_i / η
end for
return {w, vetoplayerset, (β_1, ..., β_n)}
```

5.4.2 Analysis

The enumeration algorithm to compute Banzhaf indices of players in a MWVG has an exponential time complexity because of the need to compute and analyse each possible coalition. The generating function method can be more time efficient but involves more storage of data. It requires the computation of $B(x_1, ..., x_m)$ and $B_i(x_1, ..., x_m)$ for player i, for $1 \leq i \leq n$. The storage requirements increase even more if $B(x_1, ..., x_m)$ is encoded in a coefficient array. This makes the storage dependent on the sum of the weights in each component game.

The space complexity of the generating function method to compute Banzhaf indices of players in a MWVG is $c + \sum_{1\leq i\leq n} c_i + k$ where c is the number of terms of $B(x_1, ..., x_m)$ and c_i is the number of terms in $B_i(x_1, ..., x_m)$. Then,

$$c + \sum_{1\leq i\leq n} c_i \leq c + nc \leq (n+1) \prod_{1\leq t\leq m} (1 + w^t(N))$$

This follows from the fact that the generating function method requires the computation of $B(x_1, ..., x_m)$, the generating function of the overall game and $B_i(x_1, ..., x_m)$, the generating function of each player i. We can utilize Proposition 6.3 to limit the space requirements when the weights themselves may not be perfectly accurate as is the case in Chapter 7 where population figures are weights. This scaling of the WVGs into WVGs with smaller weights keeps the properties of the WVG invariant. Moreover, we have identified players with same voting weight to avoid re-computation of their generating functions and their underlying coefficient arrays. Whereas the Mathematica programs to compute Banzhaf indices of multiple weighted voting games with 2 or 3 games are available, the Mathematica code presented at the end of the chapter computes Banzhaf indices of an arbitrary number of players. In case the space complexity of the generating function method is high, the generating function and the corresponding coefficient array for each player can be computed, and then cleared after extracting the number of swings of that player. Appendix A shows the Mathematica code to compute Banzhaf indices of players in a MWVG. A variation of the program was used for analysis in Chapter 7.

6

Efficient algorithm to design weighted voting games

All animals are equal but some animals are more equal than others.
- George Orwell, Animal Farm

The passion of men for equality is ardent, insatiable, eternal, and invincible.
- De Tocqueville, 1860

Scientific and humanist approaches are not competitive but supportive, and both are ultimately necessary
- Robert C. Wood

Abstract The calculation of voting powers of players in a weighted voting game has been extensively researched in the last few years. However, the inverse problem of designing a weighted voting game with a desirable distribution of power has received less attention. We present an efficient algorithm which uses generating functions and interpolation to compute an integer weight vector for target Banzhaf power indices. This algorithm has better performance than any other known to us. It can also be used to design egalitarian two-tier weighted voting games and a representative weighted voting game for a multiple weighted voting game.

6.1 Introduction

6.1.1 Motivation

WVGs have been applied in various political and economic organizations for structural or constitutional purposes. Prominent applications include the United Nations Security Council, the Electoral College of the United States and the International Monetary Fund ([134], [4]). The distribution of voting power in the European Union Council of Ministers has received special attention in [1], [130], [129] and [83]. Voting power is also used in joint stock companies where each shareholder gets votes in proportion to the ownership of a stock ([5], [94]).

If one assumes that each coalition has the same probability of forming and all players are independent of each other, then Banzhaf indices are the most natural way to gauge the influence of each player. The calculation of voting powers of the voters, which is NP hard in all well known cases [152], has been extensively researched in the last few years. However, the inverse problem of designing a WVG with a desirable distribution of voting power has received less attention. In this chapter, we present an efficient algorithm to compute a corresponding integer weights vector to approximate a given vector of Banzhaf indices. This is a natural extension of the work on the method of generating functions to compute voting power indices. The algorithm is designed as a ready-made tool to be used by economists and political scientists in their analysis of WVGs.

This algorithm has better performance than any other known to us. We have looked at designing multiple weighted voting games and also proposed further directions for research. Experiments with variations of the algorithm also promise to give better insight into the nature of the relationship between voting weights and corresponding voting powers.

6.1.2 Outline

Section 6.2 provides a survey of approaches to designing voting games and WVGs in particular. It also includes our main algorithm to design WVGs and its analysis. Section 6.3 highlights an application of our algorithm which is to use the Penrose square-root law to design 'egalitarian' WVGs. Similarly, Section 6.4 shows how our algorithm can be used to find a 'representative' single WVG for a multiple weighted voting game, i.e., a single game for which the Banzhaf indices are approximately the same as those for the multiple games. Section 6.5 presents conclusions and open problems.

6.2 Designing weighted voting games

6.2.1 Outline and survey

By $P(w)$, we shall denote the vector of Banzhaf indices for weight vector w and some specified quota. The problem of designing WVGs can be defined formally as follows:

Definition 6.1. ComputeRealWeightsforGivenPowers*: Given a real number vector $P = (p_1, \ldots, p_n)$, the target Banzhaf indices for the n players, some appropriate Error function and a small positive real ϵ, compute real approximate weights $w = (w_1, \ldots, w_n)$ such that $Error(P, P(w)) \leq \epsilon$.*

The problem of designing WVGs was first discussed in [130] and [136]. Holler et al. [136] proposed an iterative procedure with a stopping criterion to approximate to a game which has a voting power vector almost equal to the target. The method was to choose an initial weight vector w^0 and use successive iterations to get a better approximation: $w^{r+1} = w^r + \lambda(d - P(w^r))$ where λ is a scalar and $P(w^r)$ is the power vector of w^r. The approach has been used to analyse the Council of European Union and the International Monetary Fund Board of Governors.

Not every power distribution (Banzhaf index vector) is feasible and might not have corresponding weights for it. For example, for any number of players, there are power distributions which cannot be satisfied even by simple games [33]. Tolle [204] shows that in any 4-player WVG with no vetoers, there are only five possible Banzhaf power distributions. There are some unexplored questions concerning the convergence of the vector, such as whether the iteration always converges to the right region. It is also critical to design systems with desirable properties.

Carreras [41] points out factors considered in designing simple games. The focus is different from the computation of powers and weights. The role of blocking coalitions is analysed in a simple game. Similarly, complexity results in designing simple games are provided in [201].

6.2.2 Algorithm to design weighted voting games

We provide a more effective *hill climbing* approach than the previous proposed algorithms. Our algorithm tackles a variation of the problem *ComputeRealWeightsforGivenPowers*:

Definition 6.2. IntegerWeightsforGivenPowers*: Given a real number vector* $P = (p_1, \ldots, p_n)$*, the target Banzhaf indices for the n players, some appropriate Error function and small positive real* ϵ*, compute integer approximate weights* $w = (w_1, \ldots, w_n)$ *such that* $Error(P, P(w)) \leq \epsilon$*.*

The reason we are computing integer voting weights is that we want to utilize the generating function method (outlined in Section4.5) in each iteration. The constraint of having integer weights is reasonable. Firstly, many WVGs naturally have integer weights. Secondly, some policy makers feel more comfortable dealing with integers. Thirdly, our algorithm is giving results to a high degree of accuracy even without using real or rational weights. Algorithm 8 provides a higher level sketch of the steps required to design a WVG for a target vector of Banzhaf indices. The Mathematica code is included at the end of the chapter. The algorithm is discussed in subsection 6.2.3.

6.2.3 Algorithmic and technical issues

In Algorithm 8, the normal distribution approximation has been used to get an initial estimate of the voting weights. Leech [135] also uses the multi-linear extension approach in approximation of voting powers.

Algorithm 8 IntegerWeightsforGivenBIs

.

Input: Target vector of Banzhaf indices, T.
Output: Corresponding vector of integer voting weights.

1: Use Normal Distribution Approximation to get an initial estimate of the weights.
2: Multiply the real voting weights by a suitable real number λ which minimizes the error while rounding to get new integer voting weights.
3: Use the Generating Function Method to compute new vector of voting powers.
4: Interpolate by using a best fit curve to get the new real voting weights.
5: Repeat Step 2 until the sum of squares of differences between the powers and the target powers is less than the required error bound.

The key step of the algorithm is to use the generating function method to compute voting powers of estimated voting weights in a limited number of iterations. Bilbao et al. [36] prove that the computational complexity of computing Banzhaf indices by generating functions is $O(n^2C)$ where C is the number of nonzero coefficients in $\prod_{1\leq i\leq n}(1+x^{w_j})$. Since for each player i, we check the coefficients of all terms in $B_i(x)$ ranging from x^{q-w_i} to x^{q-1}, we have limited the time complexity of Banzhaf index computation by using moderate sized integer weights. WVGs with big integer weights can be approximated by WVGs with smaller integer weights by simple scaling and rounding off. The scaling of the WVG's into WVG's with smaller weights keeps the properties of the WVG invariant.

Observation 6.3 *The power indices of players in WVG* $v = [q; w_1, \ldots, w_n]$ *are the same as the power indices in the WVG* $\lambda v = [\lambda q; \lambda w_1, \ldots, \lambda w_n]$.

Since the generating function method can only be applied on integer votes, in each iteration, Algorithm 1 rounds off interpolated values to integer values. This rounding off can lead to varying errors if different potential multiples of the same real voting weight vector are used. After every interpolation step, we find a real λ which is multiplied with the voting weight vector and tries to minimize the total error on rounding. To find an appropriate λ to multiply with the voting weights vector we minimize the sum of squares of the difference between new real voting weights and the rounded new voting weights. That is, if $w_1, w_2, \ldots, w_n$ are reals and $m_i = Round(\lambda w_i)\ \forall i \in N$, we want to minimize $\sum_{i\in N}\left(\frac{m_i}{M} - \frac{w_i}{W}\right)^2$ where $M = \sum_{i\in N} m_i$ and $W = \sum_{i\in N} w_i$. A reasonable alternative would be to minimize the sum of the differences between the normalized voting weights and their corresponding rounded normalized voting weights.

One extra degree of freedom which we have ignored is the variation in the quota. The same voting weights profile results in different Banzhaf indices according to the quota. The exact effect on the Banzhaf indices of changing the quota presents various open problems. One concern is the extra error induced when the interpolated weights are rounded off. Ideally, we will want the positive and negative differences in rounding to be balanced. The likelihood of this balance increases as we use more players.

6.2.4 Performance and computational complexity

As mentioned previously, the computational complexity of computing Banzhaf indices by generating functions is $O(n^2C)$ where C is the number of nonzero coefficients in $\prod_{1\le i\le n}(1 + x^{w_j})$. We can approximate the number of iterations required based on the number of significant figures required in our final solution. At the end of the chapter, the Mathematica code for Algorithm 8 is provided. In the example in Figure 6.1, the algorithm converges and achieves an error of 1.481×10^{-6} in a single iteration. Also, our algorithm is giving an error of less than 10^{-8} for 30 players after only 3 iterations.

6.3 Designing egalitarian voting games

Designing egalitarian voting games is a pertinent issue in resource allocation and also political bodies. Felsenthal and Machover [82] have obtained the following result for a two-tier voting system based on Penrose's seminal paper [177].

Theorem 6.4. *(*Penrose's Square-Root Law*) Let v be a 2-tier voting system in which representatives of m different countries with populations $\{n_1, \ldots, n_m\}$ vote in a WVG and opt for their country's majority decision. Then for sufficiently large*

n_i's, the indirect probabilities of citizens from different countries being critical in a decision in v are equal (with negligible error) if and only if the Banzhaf indices β_i of the representatives are proportional to the respective $\sqrt{n_i}$.

The theorem utilizes Stirling's approximation as the n_i's tend to infinity. It assumes that the 'yes' and 'no' decisions are equiprobable and the voters are independent. Based on this result, we can devise an algorithm to compute voting weights of countries so that every member of any country has approximately equal voting power.

Algorithm 9 FairIntegerWeightsforGivenPopulations

Input: Vector of state populations, $p = \{n_1, \ldots, n_m\}$.

Output: Corresponding vector of voting weights $w = \{w_1, \ldots, w_n\}$.

1: Let $B = \sum_{1 \le i \le m} \sqrt{n_i}$.
2: Target powers, $T = \{\sqrt{n_1}/B, \ldots, \sqrt{n_m}/B\}$.
3: Run Algorithm 1 with input T and return the output.

W. Slomczynski and K. Zyczkowski [197] have proposed that giving each nation a vote w_i proportional to $\sqrt{n_i}$, the square root of its population, and establishing a normalized quota rule equal to $(1 + \frac{\sqrt{\sum_{i=1}^{m} {w_i}^2}}{\sum_{i=1}^{m} w_i})/2$ makes the voting rule almost egalitarian. The method has been called the *Jagiellonian compromise* and will be discussed in detail in Chapter 7. Although this voting method appears to be elegant and transparent, Algorithm 10 provides an alternative in which we can change the quota to accommodate various levels of efficiency in making a decision.

6.4 Multiple weighted voting games

In Chapter 5, we outlined a generating function method to find the Banzhaf indices of players in a multiple weighted voting game. Algorithm 8 and Algorithm 7 can be used to produce an approximate WVG as a representative for a multiple weighted voting game.

Algorithm 10 SingleWVGForMultipleWVGs

Input: Multiple weighted voting game $(N, v_1 \wedge \cdots \wedge v_m)$.

Output: Corresponding approximate WVG.

1: Use Algorithm 7 to compute vector of Banzhaf indices, $T = \{\beta_1, \ldots, \beta_n\}$.

2: Run Algorithm 8 with target vector T to compute the corresponding WVG v.

3: Return v.

6.5 Conclusion & open problems

This chapter provides an algorithm which will be useful for practitioners in the voting power field. This has various applications because of the ubiquitous nature of WVGs, e.g., in reliability theory. Moreover we analyse computational considerations which will be of interest to computer scientists and engineers. We notice that our algorithm can be used to design egalitarian two-tier WVGs and also to find a representative WVG for multiple weighted voting games.

It is an interesting problem to analyse multiple voting games as a function of their constituent single WVGs. Moreover Slomczynski et al. [197] have created interest in the effect of the quota on WVGs. A deeper analysis of the effect is required. O' Donnell and Servedio [182] have examined the related the problem of designing WVGs for given Chow parameters. They provide a randomized PTAS to compute an approximate WVG. We mentioned the need for a deeper analysis on the convergence of our algorithm to to design a WVG. Alon and Edelman [3] recently examine our question and investigate which non-negative vectors of sum one can be closely approximated by Banzhaf vectors of simple voting games with n players.

```
(* Title:Compute Integer Weights for Given Powers (Mathematica Version:5.2) *)
(* Description:Illustrative Mathematica code for Algorithm 1  *)

Clear[RW]; Off[InterpolatingFunction::"dmval"];
g[v_] := Apply[Times, Map[(1 + x^#) &, v]];
s[r_] := Normal[Series[1 / (1 - x), {x, 0, r - 1}]];
h[r_, v_] := Expand[g[v] / (1 + x^r)];
p[r_, v_] := Coefficient[s[r] h[r, v], x^Round[Total[v] / 2 - 1]];
(* strict majority rule *)
Ind[v_] := Map[p[#, v] &, v]; (* raw index *)
NBI[v_] := (temp = Ind[v]; temp / N[Total[temp]]); (* normalized index *)
FirstEqual[u_, v_] := u[[1]] == v[[1]];
F[r_, RW_] := Total[(FractionalPart[RW r + 1 / 2] - 1 / 2)^2];
(* estimates the error caused by rounding RW.r *)

T = {0.0958, 0.0810, 0.0803, 0.0799, 0.0661, 0.0655, 0.0499, 0.0418, 0.0347, 0.0338,
  0.0337, 0.0335, 0.0333, 0.0313, 0.0302, 0.0299, 0.0245, 0.0243, 0.0239, 0.0204,
  0.0203, 0.0164, 0.0148, 0.0127, 0.0091, 0.0069, 0.0060}; (* target powers *)
P[n_] := NBI[V[n]]; DT[n_] := P[n] - T; Err[n_] := Total[DT[n]^2];
(* error function is the sum of squares of differences from the target powers *)
RW[n_] :=
 RW[n] = Map[Interpolation[Union[Transpose[{P[n - 1], V[n - 1]}], SameTest → FirstEqual],
    InterpolationOrder → 2], T];
Go[n_] := ( Print["Next real weights RW[", n, "] = ", NumberForm[RW[n], 4]];
   r = Minimize[F[s, RW[n]], 0.8 < s < 1.2, s][[2, 1, 2]]; Print["Good multiplier = ", r];
   V[n] = Round[RW[n] r]; Print["Next integer weights V[", n, "] = ", V[n]];
   Print["Error = ", NumberForm[Err[n], 4]];);

V[0] = Round[1000 Map[InverseErf, T] / Total[Map[InverseErf, T]]];
(* Initial approximation *)
Print["Initial integer weights V[0] = ", V[0]];
Print["Initial error = ", NumberForm[Err[0], 4]];

Initial integer weights V[0] = {96, 81, 80, 80, 66, 66, 50,
  42, 35, 34, 34, 33, 33, 31, 30, 30, 24, 24, 24, 20, 20, 16, 15, 13, 9, 7, 6}

Initial error = 0.00003265

Go[1]

Next real weights RW[1] =
 {92.42, 79.52, 78.89, 78.53, 65.8, 65.24, 50.25, 42.29, 35.22, 34.32, 34.22, 34.01, 33.81,
  31.81, 30.7, 30.4, 24.95, 24.75, 24.35, 20.8, 20.7, 16.74, 15.11, 12.97, 9.298, 7.051, 6.131}

Good multiplier = 0.853146

Next integer weights V[1] = {79, 68, 67, 67, 56, 56, 43,
  36, 30, 29, 29, 29, 29, 27, 26, 26, 21, 21, 21, 18, 18, 14, 13, 11, 8, 6, 5}

Error = 1.481×10^-6

Go[2]

Next real weights RW[2] =
 {78.75, 67.79, 67.25, 66.94, 56.11, 55.63, 42.86, 36.08, 30.05, 29.28, 29.19, 29.02, 28.85, 27.14,
  26.19, 25.94, 21.29, 21.12, 20.77, 17.75, 17.66, 14.27, 12.91, 11.06, 7.929, 6.015, 5.229}

Good multiplier = 1.0006

Next integer weights V[2] = {79, 68, 67, 67, 56, 56, 43,
  36, 30, 29, 29, 29, 29, 27, 26, 26, 21, 21, 21, 18, 18, 14, 13, 11, 8, 6, 5}

Error = 1.481×10^-6

(* The algorithm is seen to have converged after just one iteration. *)
```

Fig. 6.1. Mathematica program to design WVGs

7

EU: a case study

The vote is the most powerful instrument ever devised by man.
- Lyndon B. Johnson

Democracy is a process by which the people are free to choose the man who will get the blame.
- Laurence J. Peter

Abstract In this chapter, we analyse the real life case-study of the EU constitution. The Double Majority rule in the Reform Treaty agreed in Rome in September 2004 is claimed to be simpler, more transparent and more democratic than the existing rule. We examine these questions against the democratic ideal that the votes of all citizens in whatever member country should be of equal value, using voting power analysis. We also consider possible future enlargements involving candidate countries and then to a number of hypothetical future enlargements. We find the Double Majority rule fails to measure up to the democratic ideal in all cases. We find the Jagiellonian compromise to be very close to this ideal.

7.1 Introduction

The Reform Treaty agreed in Rome in September 2004 contains fundamental reforms to the voting system used by the Council of Ministers. The current triple-majority system would be replaced with a double-majority decision rule that is said to be simpler to understand, more democratic and more flexible. In this chapter we investigate these claims using voting power analysis in a number of enlargement scenarios.

The Council of Ministers, the senior legislature of the EU, is an intergovernmental body in which some matters are decided by unanimity but the most important voting rule is qualified majority voting (QMV), that is being used for an increasing number of decisions. Under QMV each country has a different number of votes that it can cast that is related in some way to its size. Under the Reform Treaty proposals they will be strictly proportional to population sizes but under the system determined by the Nice Treaty and under the previous system the voting weights were not directly based on populations in a transparently mathematical way.

The problem of the determination of the voting weights is an important one because under the rules of the council each country must cast its votes as a bloc; a country is not permitted to divide its votes for any reason, as it might, for example in order to reflect a division of public opinion at home in the country. Alternatively if, instead of a single representative with many votes, the country's representation were by numbers of elected individuals who would vote individually as representatives or delegates rather than as a national group acting en bloc, as for example members of the European Parliament are able to do, the problem addressed in this chapter would not exist.

In that case the voting power of the citizen of each country would be approximately the same. However in a body that uses weighted voting, there is not a simple relation between weight and voting power and each case must be considered on its merits by considering the possible outcomes of the voting process, making a voting power analysis. The proposed new Double Majority rule is that a decision taken by QMV should require the support of 55 percent of the member countries whose combined populations are at least 65 percent of the EU total. This contrasts with the system currently in use (the Nice system) under which each country has a given number of weighted votes, all of which were laid down in the Nice Treaty. Specifically the Nice system is a triple-majority rule that works as follows. For a vote to lead to a decision, three requirements must be met: (i) the countries voting in favour must constitute a majority of members; (ii) they must contain at least 62 percent of the population of the Union; and (iii) their combined weighted votes must exceed the specified threshold. The Nice Treaty specified a threshold that depended on the size of the membership: for the union of 15 countries it was about 71 percent of the total of the weighted votes, increasing gradually with enlargement to its present level, with 27 members, to almost 74 percent.

Studies using voting power analysis have concluded that the Nice system is broadly equitable in the sense that the resulting powers of individual countries are fair in relative terms (in an appropriately defined sense), with one or two exceptions, but that the threshold was set much too high for the Council to be able to deal with a greater range of decisions by qualified majority voting in an efficient manner [133] and [145].

Advocates of changing to the Double Majority rule argue, first, that it would be much simpler to understand than the Nice system (which has been described as 'fiendishly complex') which is lacking in transparency because of its use of artificially constructed voting weights. The Nice weights are criticized because they are not, even approximately, directly proportional to populations; the countries with larger populations are assigned larger weights than the smaller ones but the difference does not fully reflect relative populations. Superficially it appears that larger states are underrepresented, although it can be argued that such weights may well, in actual fact, be consistent with a reasonable degree of fairness in the distribution of voting power. But this argument by itself would not be decisive in favour of change given that the Nice system is already in place.

A second criticism of the Nice system is that the threshold is set too high, and moreover, increasing it as the membership increases, makes decisions harder by requiring a larger qualified majority, or making it easier for a blocking minority to form. Studies of the formal a priori decisiveness of the system have shown that the probability of a qualified majority emerging could be extremely small [144]. However, despite these fears, recent studies have found little evidence in practice of the sclerosis that was feared, and qualified majority voting appears to be working quite well [214].

The third argument for change is that the Nice system was designed for certain specified anticipated enlargements of the Union, which have now all occurred. It provided for a union of up to 27 members - the fifteen members at the time of the treaty, plus the ten countries that acceded in May 2004 followed by Bulgaria and Romania that joined in January 2007 - and further enlargement beyond that is therefore outside its scope. The formal position is that the accession of a new country would require a new treaty that included amendments to the system of qualified majority voting. But there are further candidates, including Turkey and the former republics of Yugoslavia; and there is also the remote possibility of further FSU countries, and perhaps also other European countries joining. It would clearly be impossibly inefficient to have to hold an Intergovernmental Conference every time further enlargement took place. So there is need for a system that embodies a principle that can be applied in a more or less routine manner each time a new member accedes. An example of such a voting system is the double-majority rule in the Reform Treaty.

It is this administrative simplicity that makes the double-majority voting rule most attractive. It enables us to know immediately how many votes a new member will have and in what ways the operation of qualified majority voting will be affected. All that it is necessary to know is the country's population.

7.2 Appraisal of voting rules by power analysis

It does not follow that we understand all the consequences of enlargement for the fairness and efficiency of the voting system. It has been claimed that a weighted voting rule, based directly on populations, will implement a desirable democratic principle of equality: that each country will have a voting power proportional to its population. That is undoubtedly a major factor in the thinking behind the proposal. However it is a serious mistake because in a weighted voting body like the council, where members cast all their votes as a single bloc, power in the sense of the ability to influence decisions is not related straightforwardly to weight. It is possible, for example, for a country to have voting weight that is not translated into actual voting power. It is therefore necessary to make a voting power analysis to establish the properties of this system and, in particular, the powers of the members.

In this chapter we do this by considering voting in the Council of Ministers as a formal two-stage democratic decision process that allows us to compare voting power of citizens of different countries. It is a fundamental principle of the EU that all citizens should have equal rights in whatever country they happen to live. This provides a natural criterion by which to judge the adequacy of the voting system, a benchmark against which to compare the fairness of the distribution of voting power. We use voting power analysis to do this, following the approach of Penrose [177, 178], treating the Council of Ministers as a delegate body on which individual citizens are represented by government ministers elected by them.

The voting power of a citizen is derived from two components: (i) power of his or her country in the council (a property of the system of weighted QMV in the council), and (ii) the power of the citizen in a popular election within the country. A citizen's voting power, as a structural property of the voting system, is measured by his or her Penrose power index, which is the product of these two voting power indices.

We make two analyses. First we compare the double-majority rule with the Nice system for the current EU of 27 countries. Secondly we investigate various scenarios for further enlargement. These begin with the expected accession of the known candidate countries and then become more and more speculative as further new members are presupposed. Our primary purpose is to test the claim that the Reform Treaty proposals are simple and transparent in the face of further enlargement. We also investigate the alternative voting rule that has recently been proposed, known as the Jagiellonian Compromise, and find it remarkably equitable. [197].

We report analyses of the following Scenarios for possible future enlargement of the EU:

- EU27: the current union. Member countries: Austria, Belgium, Bulgaria, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, United Kingdom.
- I EU29: as above plus Croatia, Macedonia.
- II EU30: with Turkey.
- III EU34: with Albania, Bosnia, Montenegro, Serbia.
- IV EU37: with Norway, Iceland, Switzerland.
- V EU40: with Belarus, Moldova, Ukraine.
- VI EU41: with Russia.

In the next section we describe the mathematics of the voting power approach that we employ to analyse these scenarios.

7.3 The Penrose index approach

The EU Council of Ministers at any time is assumed to consist of n member countries, represented by a set $N = \{1, 2, ..., n\}$, where each country is labelled by an integer $i = 1$ to n. Each country has a population (which we take, as a first approximation, to be the same as its electorate), represented for country i by m_i. The total population of the EU is $\sum_{i=1}^{n} m_i$.

Under the Reform Treaty, any normal decision requires a double majority in favour of the proposal: at least 55% of member countries whose combined populations are at least 65% of the total population. Suppose that in any vote concerning such a decision there are s countries in favour, represented by a set S, a subset of N. Then the decision is taken if $s \geq 0.55n$ and $\sum_{j \in S} m_j \geq 0.65m$. The double majority game is basically a MWVG of dimension 2 where a coalition S of countries is winning if and only if $s = |S| \geq 0.55n$ and $\sum_{j \in S} m_j \geq 0.65m$. The voting power of a country is the probabilistic Banzhaf index or Penrose index in the two dimensional MWVG. If the number of swings for country i is denoted by η_i, then the Penrose index for the Council of Ministers is defined as P_i^C

$$P_i^C = \frac{\eta_i}{2^{n-1}}$$

The study utilizes the Algorithm 7 outlined in Chapter 5. The Penrose index or Probabilistic Banzhaf Index is a measure of absolute voting power in the sense of a country's likelihood of being decisive when all voting outcomes are considered on an equal basis. The power of an individual citizen is defined formally by idealising the council as a representative body in which determination of how a country will cast its weighted votes follows a simple majority among its citizens. This requires finding a measure of power of a single citizen within a country.

For country i with m_i voters, Probabilistic Banzhaf index or the Penrose power index is the probability that the number of votes cast by the $m_i - 1$ voters, other than the single citizen under consideration, are precisely one vote short of a majority. Denote this power index for a single vote of any citizen in country i by P_i^S. Then,

$$P_i^S = Pr(\text{combined votes of } m_i - 1 \text{ voters} = \tfrac{m_i}{2}) = \binom{m_i - 1}{\lfloor m_i/2 \rfloor}(0.5)^{m_i - 1}.$$

When m_i is large, P_i^S can be approximated accurately by Stirling's formula:

$$P_i^S \approx \sqrt{\frac{2}{\pi m_i}}.$$

We can evaluate the indirect power index P_i for a citizen of country i as following:

$$P_i = P_i^C P_i^S$$

The value of P_i is of course rather small because P_i^S is small. However its value can vary enormously between countries, over changes in the membership of the Union and changes in the voting system. P_i provides a yardstick to use in the evaluation of the weighted voting system on a consistent basis of democratic legitimacy. Comparisons can be made using relative voting power indices to compare countries and therefore to test the extent to which the voting system is egalitarian.

P_i can be used on the basis of the Penrose Square Root rule for equalising voting power in all countries. The rule is that weighted voting be adopted in the council with a decision rule such that P_i is constant for all i. The power indices P_i^C should therefore be proportional to the square roots of the populations. This can be a decision rule with a single majority. An approximation to this that will be sufficient in many cases is to choose weights themselves in proportion to the population square roots. This has been applied recently in the so-called Jagiellonian compromise in which the decision rule is adjusted to improve the approximation [197]. We have investigated the performance of this proposed voting rule in equality of voting power and find it works very well indeed.

7.4 Analysis: voting power in EU27

We compare the three voting systems for the present day union EU27. The results are given in Figure 7.1 in which countries are in size order. The Penrose indices agree very closely with those of Falsenthal and Machover [84]. The generating function method requires that the weights be integers that are not too large. This means that it is more feasible in practice to replace population figures, which are mostly in millions, by much smaller proportional integers. This is a reasonable transformation considering that population figures themselves are estimates and also dynamic.

Besides the Penrose index for each country P_i^C, Figure 7.1 also shows the indirect citizen power indices P_i. These are presented in two ways, as absolute values and relative to the power of a citizen of Germany. The relative voting power indices show the inequality in the double majority voting rule, with all but those for the smallest group of countries being less than 1. Inequality is measured by the Gini coefficient of citizen power for the whole figure.

Figure 7.1 shows how much more unequal the proposed voting system would be compared with the existing Nice system, under which citizens of every country (with the slight exception of Latvia) have slightly greater voting power than those

Table 1: Voting Power Analysis of the EU27

		Nice			Reform Treaty				Jagiellonian			
Country	Population	Penrose Index	Czn Power	Rel Czn Power	Weight	Penrose Index	Czn Power	Rel Czn Power	Weight	Penrose Index	Czn Power	Rel Czn Power
Germany	82,437,995	0.03269	2.87E-06	1.000	824	0.20011	1.76E-05	1.000	9080	0.20080	1.76E-05	1.00000
France	62,998,773	0.03269	3.29E-06	1.144	630	0.15517	1.56E-05	0.887	7937	0.17597	1.77E-05	1.00246
UK	60,393,100	0.03269	3.36E-06	1.168	604	0.14932	1.53E-05	0.872	7771	0.17233	1.77E-05	1.00267
Italy	58,751,711	0.03269	3.40E-06	1.185	588	0.14587	1.52E-05	0.863	7665	0.16999	1.77E-05	1.00276
Spain	43,758,250	0.03116	3.76E-06	1.308	438	0.11252	1.36E-05	0.772	6615	0.14675	1.77E-05	1.00310
Poland	38,157,055	0.03116	4.02E-06	1.401	382	0.09816	1.27E-05	0.721	6177	0.13702	1.77E-05	1.00300
Romania	21,610,213	0.01789	3.07E-06	1.069	216	0.07139	1.23E-05	0.697	4649	0.10307	1.77E-05	1.00256
Netherlands	16,334,210	0.01669	3.29E-06	1.147	163	0.06006	1.19E-05	0.674	4042	0.08959	1.77E-05	1.00230
Greece	11,125,179	0.01547	3.70E-06	1.288	111	0.04933	1.18E-05	0.671	3335	0.07390	1.77E-05	1.00182
Portugal	10,569,592	0.01547	3.80E-06	1.322	106	0.04830	1.19E-05	0.674	3251	0.07203	1.77E-05	1.00184
Belgium	10,511,382	0.01547	3.81E-06	1.325	105	0.04810	1.18E-05	0.673	3242	0.07184	1.77E-05	1.00184
Czech	10,251,079	0.01547	3.86E-06	1.342	103	0.04768	1.19E-05	0.676	3202	0.07095	1.77E-05	1.00191
Hungary	10,076,581	0.01547	3.89E-06	1.354	101	0.04727	1.19E-05	0.676	3174	0.07031	1.77E-05	1.00156
Sweden	9,047,752	0.01299	3.45E-06	1.199	90	0.04500	1.19E-05	0.679	3008	0.06664	1.77E-05	1.00180
Austria	8,265,925	0.01299	3.60E-06	1.255	83	0.04356	1.21E-05	0.687	2875	0.06369	1.77E-05	1.00168
Bulgaria	7,718,750	0.01299	3.73E-06	1.299	77	0.04233	1.22E-05	0.691	2778	0.06154	1.77E-05	1.00149
Denmark	5,427,459	0.00916	3.14E-06	1.092	54	0.03758	1.29E-05	0.732	2330	0.05161	1.77E-05	1.00164
Slovakia	5,389,180	0.00916	3.15E-06	1.096	54	0.03758	1.29E-05	0.735	2321	0.05141	1.77E-05	1.00131
Finland	5,255,580	0.00916	3.19E-06	1.110	53	0.03738	1.30E-05	0.740	2293	0.05078	1.77E-05	1.00161
Ireland	4,209,019	0.00916	3.56E-06	1.240	42	0.03510	1.37E-05	0.776	2052	0.04544	1.77E-05	1.00145
Lithuania	3,403,284	0.00916	3.96E-06	1.379	34	0.03344	1.45E-05	0.822	1845	0.04086	1.77E-05	1.00143
Latvia	2,294,590	0.00525	2.77E-06	0.963	23	0.03116	1.64E-05	0.933	1515	0.03355	1.77E-05	1.00142
Slovenia	2,003,358	0.00525	2.96E-06	1.030	20	0.03053	1.72E-05	0.979	1415	0.03133	1.77E-05	1.00070
Estonia	1,344,684	0.00525	3.61E-06	1.257	13	0.02908	2.00E-05	1.138	1160	0.02568	1.77E-05	1.00148
Cyprus	766,414	0.00525	4.78E-06	1.666	8	0.02803	2.56E-05	1.453	875	0.01938	1.77E-05	1.00069
Luxembourg	459,500	0.00525	6.18E-06	2.151	5	0.02741	3.23E-05	1.835	678	0.01501	1.77E-05	1.00149
Malta	404,346	0.00396	4.97E-06	1.730	4	0.02720	3.41E-05	1.941	636	0.01408	1.77E-05	1.00126
Total	492,964,961	0.41999			4930	1.71869			95921	2.12556		
Quota					3205				59062			
Power to Act				0.020				0.129				0.163
Gini Coeff				0.059				0.080				9.08E-05

Czn Power: Citizen Power P_i

Fig. 7.1. Voting Power Analysis of the EU27

of Germany (a result due to the fact that Germany has no greater weight than France, Italy and the UK despite its much larger population): the Gini coefficient for Nice being 0.059, that for the Reform Treaty, 0.080. We have also reported the power to act of the Council of Ministers, which shows that the Reform Treaty voting rule is a very much more decisive voting rule than Nice, with a power to act of 0.129 compared to a very low value of 0.02.

The results for the Jagiellonian Compromise are quite impressive in showing that this method would lead to the equalisation of voting power throughout the union of 27 countries. There is almost no variation in the relative citizen voting power indices across countries, which indicates how good an approximation to the Penrose square root rule is obtained by using population square roots as weights.

7.5 Analysis: enlargement scenarios

Figure 7.2 presents the results for the enlargement scenarios I to VI. They are presented diagrammatically for existing members in Figures 7.4, 7.5 and 7.6. We also present a parallel analysis for the Jagiellonian Compromise in Figure 7.3. In all scenarios the same population figures have been used, the 2006 estimates taken from Eurostat.

They show that the inequality in citizen voting power under the Double Majority rule persists although there are sharp changes in relative voting power following changes in the membership. On the other hand, the Jagiellonian system turns out to be remarkably successful in creating a very equal distribution of citizen power in all scenarios, and to be quite robust to membership changes. The use of square root weights and adjustment of the quota gives an extremely good approximation to the Penrose square root rule.

The results for the Reform Treaty voting rule in Figure 7.2 show that citizen voting power is relatively unequal under all scenarios. The Gini coefficient for Scenario VI (41 countries including Russia) is the same as in Scenario O (EU27) although it falls below this in some scenarios. Citizen voting power is most equal following the accession of Turkey, Gini = 0.059, Scenario II, that may be largely due to the similarity in population of the two largest members, Germany and Turkey. Whereas having one country that is much larger than all the others creates an unequal power distribution, where there are two members with very large weight, a bipolar voting structure, there is a tendency for them to counteract one another. Thus the presence of Turkey would reduce Germany's power and increase the power of other members, making the distribution more equal. The accession of Turkey would substantially increase the voting power of citizens of Poland and Spain, from 0.718 and 0.772 to 0.822 and 0.815. There is a similar effect for medium sized countries, but their relative voting power remains much lower: for example, the index for Belgium goes from 0.661 to 0.760. The effects for small countries, whose citizen voting powers are already much larger than Germany's, are quite large: for example, Malta's goes from 1.859 to 2.442. The power of the council to act, declines more or less steadily as the union enlarges, from 12.9% for O (EU27) to 9.3% in VI, although it is always much greater than under the current system.

Our overall conclusion is that the Reform Treaty's Double Majority rule falls a long way short of the democratic ideal of ensuring that the votes of all members of the community are of equal value whatever country they are cast in. It is an endemic feature that citizens of middle sized countries have considerably less voting power than those in either large or small countries. This pattern persists under all the enlargement scenarios we have looked at.

Figure 7.3 shows the results for the Jagiellonian compromise under the same scenarios. For each scenario the weights, which are the population square roots, $\sqrt{m_i}$, are shown in the first column, and the quota is equal to:

$$q = \frac{1}{2}(1 + \frac{\sqrt{\sum_{i=1}^{n} m_i}}{\sum_{i=1}^{n} \sqrt{m_i}})$$

There is almost no variation in the relative citizen voting powers either between countries or over scenarios. We conclude that the method is therefore found to be extremely successful in equalising voting power in a wide range of circumstances.

Table 2: Citizen Indirect Power Indices Under All Scenarios: Reform Treaty

				Enlargement Scenarios			
Country	O	I	II	III	IV	V	VI
Albania				0.933	1.030	0.818	1.231
Austria	0.687	0.673	0.788	0.743	0.800	0.679	0.917
Belarus						0.670	0.886
Belgium	0.673	0.661	0.760	0.721	0.770	0.667	0.871
Bosnia & H				0.872	0.960	0.771	1.139
Bulgaria	0.691	0.676	0.796	0.749	0.808	0.682	0.931
Croatia	0.000	0.742	0.906	0.841	0.921	0.748	1.087
Cyprus	1.453	1.394	1.815	1.648	1.854	1.404	2.276
Czech	0.676	0.663	0.764	0.725	0.774	0.669	0.877
Denmark	0.732	0.712	0.861	0.802	0.875	0.718	1.025
Estonia	1.138	1.094	1.409	1.284	1.439	1.102	1.756
Finland	0.740	0.719	0.871	0.811	0.885	0.726	1.038
France	0.887	0.889	0.904	0.900	0.905	0.909	0.921
Germany	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Greece	0.671	0.659	0.755	0.718	0.765	0.665	0.862
Hungary	0.676	0.663	0.765	0.725	0.775	0.669	0.879
Iceland					2.876	2.152	3.554
Ireland	0.776	0.753	0.925	0.857	0.941	0.759	1.114
Italy	0.863	0.865	0.884	0.880	0.886	0.888	0.906
Latvia	0.933	0.900	1.139	1.044	1.161	0.907	1.401
Lithuania	0.822	0.796	0.989	0.913	1.007	0.803	1.201
Luxembourg	1.835	1.758	2.304	2.087	2.355	1.770	2.902
Macedonia		0.935	1.189	1.089	1.213	0.942	1.468
Malta	1.941	1.859	2.442	2.210	2.496	1.872	3.080
Moldova						0.780	1.156
Montenegro				1.842	2.076	1.564	2.555
Netherlands	0.674	0.666	0.739	0.711	0.746	0.671	0.822
Norway					0.911	0.741	1.074
Poland	0.721	0.718	0.822	0.804	0.817	0.788	0.846
Portugal	0.674	0.662	0.761	0.722	0.771	0.668	0.872
Romania	0.697	0.692	0.746	0.724	0.752	0.692	0.812
Russia							1.225
Serbia				0.727	0.778	0.670	0.883
Slovakia	0.735	0.715	0.864	0.805	0.878	0.721	1.029
Slovenia	0.979	0.943	1.200	1.098	1.224	0.951	1.481
Spain	0.772	0.772	0.815	0.802	0.814	0.815	0.854
Sweden	0.679	0.665	0.774	0.732	0.785	0.671	0.897
Switzerland					0.814	0.686	0.939
Turkey			0.959	0.958	0.959	0.968	0.969
UK	0.872	0.874	0.892	0.888	0.893	0.896	0.911
Ukraine						0.825	0.861
Power to Act	0.129	0.126	0.110	0.106	0.092	0.096	0.093
Gini Coefficient	0.080	0.082	0.059	0.067	0.061	0.079	0.080

Weights used are given in Table 1. Gini coefficients computed at the citizen level for the whole union.

Fig. 7.2. Citizen Indirect Power Indices Under All Scenarios: Reform Treaty

Table 3: Citizen Indirect Power Indices Under All Scenarios: Jagiellonian Compromise

		Enlargement Scenarios						
		O	I	II	III	IV	V	VI
Country	Weight	EU27	EU29	EU30	EU34	EU37	EU40	EU41
Albania	1786				1.0004	1.0005	1.0005	0.9987
Austria	2875	1.0017	1.0017	1.0006	1.0007	1.0008	1.0007	0.9989
Belarus	3113						1.0009	0.9991
Belgium	3242	1.0018	1.0019	1.0007	1.0008	1.0009	1.0008	0.9990
Bosnia & H	1984				1.0007	1.0007	1.0007	0.9990
Bulgaria	2778	1.0015	1.0016	1.0004	1.0006	1.0007	1.0006	0.9988
Croatia	2134		1.0014	1.0003	1.0004	1.0005	1.0005	0.9987
Cyprus	875	1.0007	1.0007	0.9996	0.9998	0.9999	0.9999	0.9981
Czech	3202	1.0019	1.0020	1.0008	1.0009	1.0010	1.0009	0.9991
Denmark	2330	1.0016	1.0017	1.0006	1.0007	1.0008	1.0008	0.9990
Estonia	1160	1.0015	1.0017	1.0004	1.0007	1.0008	1.0008	0.9990
Finland	2293	1.0016	1.0018	1.0007	1.0008	1.0009	1.0008	0.9991
France	7937	1.0025	1.0025	1.0015	1.0014	1.0014	1.0011	1.0001
Germany	9080	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Greece	3335	1.0018	1.0018	1.0006	1.0007	1.0008	1.0007	0.9989
Hungary	3174	1.0016	1.0018	1.0006	1.0007	1.0008	1.0007	0.9989
Iceland	549					1.0011	1.0011	0.9993
Ireland	2052	1.0015	1.0017	1.0006	1.0007	1.0008	1.0008	0.9990
Italy	7665	1.0028	1.0028	1.0016	1.0016	1.0015	1.0013	1.0002
Latvia	1515	1.0014	1.0015	1.0004	1.0006	1.0006	1.0006	0.9989
Lithaunia	1845	1.0014	1.0016	1.0004	1.0006	1.0006	1.0006	0.9989
Luxembourg	678	1.0015	1.0013	1.0011	1.0005	1.0006	1.0006	0.9988
Macedonia	1428		1.0014	1.0006	1.0007	1.0008	1.0008	0.9990
Malta	636	1.0013	1.0013	1.0011	1.0005	1.0005	1.0006	0.9989
Moldova	1948						1.0006	0.9989
Montenegro	773				0.9999	1.0000	1.0000	0.9982
Netherlands	4042	1.0023	1.0024	1.0012	1.0012	1.0012	1.0011	0.9994
Norway	2167					1.0004	1.0004	0.9986
Poland	6177	1.0030	1.0030	1.0017	1.0017	1.0017	1.0014	0.9999
Portugal	3251	1.0018	1.0019	1.0007	1.0008	1.0009	1.0008	0.9990
Romania	4649	1.0026	1.0027	1.0013	1.0014	1.0014	1.0012	0.9995
Russia	11937							0.9943
Serbia	3140				1.0009	1.0010	1.0009	0.9991
Slovakia	2321	1.0013	1.0014	1.0003	1.0004	1.0005	1.0004	0.9986
Slovenia	1415	1.0007	1.0011	1.0000	1.0001	1.0002	1.0002	0.9984
Spain	6615	1.0031	1.0031	1.0018	1.0018	1.0018	1.0015	1.0000
Sweden	3008	1.0018	1.0018	1.0007	1.0008	1.0009	1.0008	0.9990
Switzerland	2736					1.0009	1.0008	0.9990
Turkey	8653			1.0007	1.0007	1.0007	1.0005	1.0001
UK	7771	1.0027	1.0026	1.0015	1.0015	1.0014	1.0012	1.0001
Ukraine	6797						1.0014	1.0000
Total		95921	99483	108136	115,819	121271	133129	145066
Quota:		59062	60917	66052	70076	72929	79451	86735

Power indices computed using program ipgenf from website, Leech and Leech (2004).

Fig. 7.3. Citizen Indirect Power Indices Under All Scenarios: Jagiellonian Compromise

7.6 Conclusions

We have tested the suitability of the proposed Double Majority rule in the EU Reform Treaty by looking at its implications for voting power under various enlargement scenarios, some of which are realistic prospects, while some are no more than speculations. Our scenarios include the possibility of virtually all European countries, up to and even including Russia, acceding to membership. We have also tested the performance of the Jagiellonian compromise based on the Penrose Square Root rule whereby voting weights are determined by a simple formula as proportional to population square roots. In judging the voting rule we looked at two criteria: (i) equality of voting power as measured by the Penrose power index at the level of the citizen, assuming one-person-one-vote in national elections, and (ii) decisiveness of the Council of Ministers, as measured by the Coleman power to act. We found that for the present union of 27, the Reform Treaty voting rule gives a much more unequal distribution of citizen voting power than the existing voting rule, although it leads to the Council of Ministers being more decisive. The Jagiellonian compromise leads to the equalisation of citizen voting power in all countries.

In considering enlargement scenarios, the inequality of citizen voting power persists with each enlargement. The common pattern is for citizens of the smallest countries to have the greatest voting power, sometimes by a factor of as much as 2 or 3 times those of other countries, such as in the cases of Malta and Luxembourg. The medium sized countries have the smallest citizen voting power. That for Netherlands, for example, varies from about two-thirds that of Germany in EU27 to about four fifths of it following the accession of Russia. Our conclusion is that the Reform Treaty voting system is a flawed proposal that fails to reach the democratic ideal of equality of voting power of all citizens in the European Union. This ideal is reached by the Jagiellonian Compromise.

On a more general note, some experiments were also conducted to see if there is critical quota for which the Shapley-Shubik indices are proportional to the weights. The weights are integers randomly generated from a uniform distribution on [1, 100] with 30, 40 and 50 players. However, no sharp threshold was observed where the weights are proportional to the Shapley-Shubik indices.

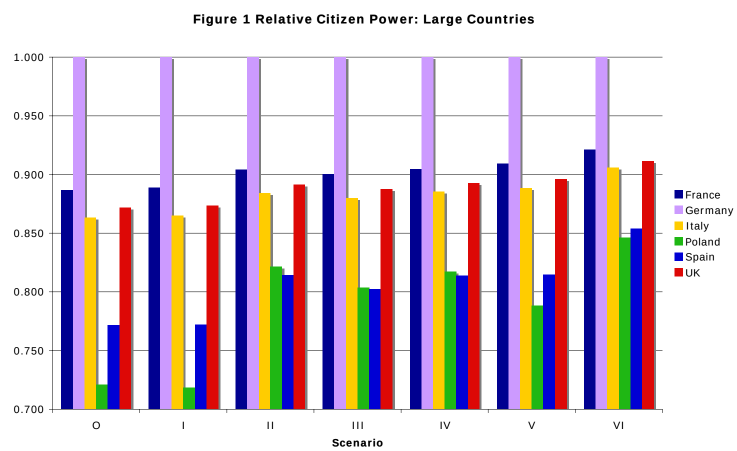

Fig. 7.4. Relative Citizen Power: Large Countries

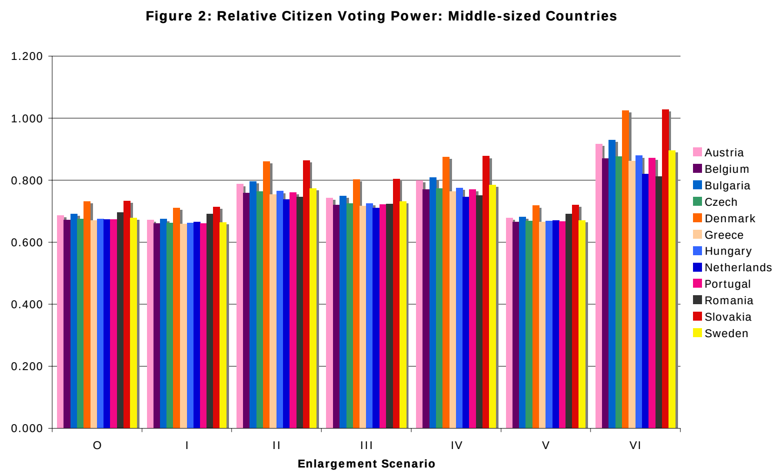

Fig. 7.5. Relative Citizen Voting Power: Middle-sized Countries

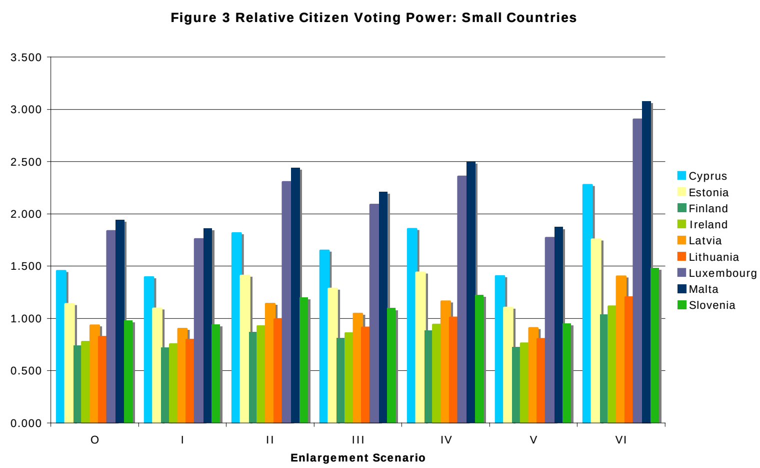

Fig. 7.6. Relative Citizen Voting Power: Small Countries

Part III

Complexity of manipulation

8

Complexity of control in weighted voting games

Complexity is not always a disease to be diagnosed; sometimes it is a resource to be exploited. But complexity turns out to be most elusive where it would be most welcome.
- Christos Papadimitriou

Would it then be possible to construct a hierarchy reflecting the difficulty of benefiting from strategic behavior?
- Hannu Nurmi [160]

Abstract An important aspect of mechanism design in social choice protocols and multiagent systems is to discourage insincere behaviour. Manipulative behaviour has received increased attention since the famous Gibbard-Satterthwaite theorem. We examine the computational complexity of manipulation in weighted voting games which are ubiquitous mathematical models used in economics, political science, neuroscience, threshold logic, reliability theory and distributed systems. It is a natural question to check how changes in weighted voting game may affect the overall game. Tolerance and amplitude of a weighted voting game signify the possible variations in a weighted voting game which still keep the game unchanged. We characterize the complexity of computing the tolerance and amplitude of weighted voting games. Tighter bounds and results for the tolerance and amplitude of key weighted voting games are also provided.

8.1 Introduction

In this section a general motivation to consider manipulation in WVGs is presented. WVGs have received increased interest in the artificial intelligence and agents community due to their ability to model various coalition formation scenarios [69]. Such games have also been examined from the point of view of susceptibility to manipulations [21, 220]. WVGs have been applied in various political and economic organizations [134, 130, 1]. Voting power is used in joint stock companies where each shareholder gets votes in proportion to the ownership of a stock [94].

WVGs and coalitional voting games are also encountered in threshold logic, reliability theory, neuroscience and logical computing devices [202, 208, 184]. There are many parallels between reliability theory and voting theory [184]. Parhami [171] points out that voting has a long history in reliability systems dating back to von Neumann [212]. Nordmann et al. [159] deal with reliability and cost evaluation of weighted dynamic-threshold voting-systems. Systems of this type are used in various areas such as target and pattern recognition, safety monitoring and human organization systems.

Elkind et al. [69] note that since WVGs have only two possible outcomes, they do not fall prey to manipulation of the type characterized by Gibbard-Satterthwaite [97]. The Gibbard-Satterthwaite theorem basically says that any reasonable voting system with three or more candidates is vulnerable to tactical voting. However, there are various ways WVGs can be manipulated and controlled. We examine some of the aspects. Tolerance and amplitude of WVGs signify the possible variances in a WVG which still keep the game unchanged. They are significant in mathematical models of reliability systems and shareholdings. For reliability systems, the weights of a WVG can represent the significance of the components, whereas the quota can represent the threshold for the overall system to fail. It is then a natural requirement to provide a framework which can help identify similar reliability systems. In shareholding scenarios [5], there is a need to check the maximum changes in shares which still maintain the status quo. In political settings, the amplitude of a WVG signifies the maximum percentage change in various votes which is possible without changing the voting powers of the voters. In this chapter, the computational aspects of amplitude and tolerance of WVGs are examined.

Section 8.2 provides a background of tolerance and amplitude. In Section 8.3, computational aspects of tolerance and amplitude are examined. It is seen that computing the amplitude and tolerance of a WVG is NP-hard. We give tighter bounds and results for the tolerance and amplitude of key WVGs such as uniform (symmetric) WVGs and unanimity WVGs.

The final section presents conclusions and ideas for future work.

8.2 Tolerance & amplitude: background

If the valuation function of a WVG v is same as another WVG v', then v' is called a *representation* of v. If the quota q' of v' is such that for all $S \subseteq N$, $\sum_{i \in S} w_i' \neq q'$, then v' is called a *strict representation* of v [89].

8.2.1 Background

The question we are interested in is to find the maximum possible variations in the weights and quotas of a WVG which still do not change the game. The two key references which address this question are [109] and [89]. Hu [109] worked within the theory of switching functions. He set forth the idea of *linearly separable switching functions* which are equivalent to each other. Freixas and Puente [89] extended the theory by framing it in the context of strict representations of WVGs, which are equivalent to linearly separable switching functions.

8.2.2 Tolerance

The setting of the problem is that we look at a transformation, $f_{(\lambda_1,\ldots,\lambda_n),\Lambda}$ which maps a WVG, $v = [q; w_1, \ldots, w_n]$ to $v' = [q'; w_1', \ldots, w_n']$ such that $w_i' = (1 + \lambda_i)w_i$ and $q' = (1 + \Lambda)q$. Let A be the maximum of $w(S)$ for all $\{S | v(S) = 0\}$. and let B be the minimum of $w(S)$ for all $\{S | v(S) = 1\}$. Then $A < q \leq B$ (and $q < B$ if the representation is strict). Moreover, let $m = \mathsf{Min}(q - A, B - q)$ and

$M = q + w(N)$. Hu [109] and then Freixas and Puente [89] showed that if for all $1 \leq i \leq n$, $|\lambda_i| < m/M$ and $|\Lambda| < m/M$, then v' is just another representation of v. They defined $\tau[q; w_1, \ldots, w_n] = m/M$ as the *tolerance* of the system. Freixas and Puente [89] also showed that the tolerance is less than or equal to $1/3$ for strict representations of a WVG and less than or equal to $1/5$ for a not necessarily monotonic[1] WVG.

8.2.3 Amplitude

Freixas and Puente defined the *amplitude* as the maximum μ such that $f_{(\lambda_1,\ldots,\lambda_n),\Lambda}$ is a representation of v whenever $\mathsf{Max}(|\lambda_1|, \ldots, |\lambda_n|, |\Lambda|) < \mu(v)$. For a strict representation of a WVG $[q; w_1, \ldots, w_n]$, for each coalition $S \subseteq N$, let $a(S) = |w(S) - q|$ and $b(S) = q + w(S)$.

Freixas and Puente [89] showed that the amplitude of a WVG is $\mu(v) = \underset{S \subseteq N}{\mathsf{Inf}} \frac{a(S)}{b(S)}$. Although both tolerance and amplitude have been used in the WVG literature to signify the maximum possible variation in the weights and the quota without changing the game, the amplitude is a more precise and accurate indicator of the maximum possible variation than tolerance.

8.3 Tolerance & amplitude: some results

8.3.1 Complexity

In all the complexity proofs in this section, we assume that the weights in a WVG are positive integers. We let WVG-STRICT be the problem of checking whether a WVG $v = [q; w_1, \ldots, w_n]$ is strict or not, i.e., WVG-STRICT = $\{v$: v is strict$\}$. Then we have the following proposition:

Proposition 8.1. *WVG-STRICT is co-NP-complete.*

Proof. Let WVG-NOT-STRICT = $\{v$: v is not strict$\}$. WVG-NOT-STRICT is in NP since a certificate of weights can be added in linear time to confirm that they

[1] Freixas and Puente also consider WVGs where players' weights can be negative.

sum up to q. Moreover v is not strict if and only if there is a subset of weights which sum up to q. Therefore the NP-complete problem SUBSET-SUM (see Garey and Johnson [95]) reduces to WVG-NOT-STRICT. Hence WVG-NOT-STRICT is NP-complete and WVG-STRICT is co-NP-complete. □

Corollary 8.2. *The problem of checking whether the amplitude of a strict WVG is zero is NP-hard.*

Proposition 8.3. *The problem of computing the amplitude of a WVG v is NP-hard.*

Proof. Let us assume that weights in v are even integers whereas the quota q is an odd integer $2k-1$. Then the minimum possible difference between q and A, the weight of the maximal losing coalition, or q and B, the weight of minimal winning coalition is 1. So $A \leq 2k-2$ and $B \geq 2k$. We see that $\mu(v) \leq 1/(4k-1)$ if and only if there exists a coalition C such that $w(C) = 2k$. Thus the problem of computing $\mu(v)$ for a WVG is NP-hard by a reduction from the SUBSET-SUM problem. □

A similar proof can be used to prove the following proposition:

Proposition 8.4. *The problem of computing the tolerance of a strict WVG is NP-hard.*

8.3.2 Uniform and unanimity WVGs

We show that the bound for the maximum possible tolerance can be improved when we restrict to strict representations of special cases of WVGs. We first look at uniform WVGs which are an important subclass of WVGs which model many multi-agent scenarios where each agent has the same voting power.

Proposition 8.5. *For a strict representation of a proper uniform WVG $v = [q; \underbrace{w, \ldots, w}_{n}]$, $\tau(v) \leq \frac{1}{3n}$.*

Proof. Since $\frac{q-A}{q+w(N)} = 1 - \frac{w(N)+A}{q+w(N)}$ is an increasing function of q and $\frac{B-q}{q+w(N)}$ is a decreasing function of q, the tolerance reaches its maximum when $q - A = B - q$, i.e. when q is the arithmetic mean $\frac{A+B}{2}$. We let the size of the maximal losing coalition be r and the size of the minimal winning coalition be $r+1$. Then the weight of a maximal losing coalition is rw and the weight of the minimal winning coalition is $(r+1)w$ and $m = w/2$. Since v is proper, $q \geq \frac{1}{2}(nw)$, and $M = q + w(N) \geq \frac{3nw}{2}$. Then,

$$\tau(v) = m/M \leq \frac{1}{3n}.$$

□

Proposition 8.6. *For a uniform WVG* $v = [q; \underbrace{w, \ldots, w}_{n}]$, *we have* $B = w\lceil \frac{q}{w} \rceil$ *and* $A = B - w$. *Then,*

$$\mu(v) = \begin{cases} \frac{q-A}{A+q}, \text{ if } q \leq \sqrt{AB} \\ \frac{B-q}{B+q}, \text{ otherwise.} \end{cases}$$

Proof. It is clear that B, the weight of the minimal winning coalition is $w\lceil \frac{q}{w} \rceil$ and A, the weight of the maximal losing coalition is $B-w$. Note that, $\frac{q-A}{q+A} \leq \frac{B-q}{q+B}$ if and only if $q \leq \sqrt{AB}$. For losing coalitions with weight w, $\frac{q-w}{q+w}$ is a decreasing function for w. For winning coalitions with weight w, $\frac{w-q}{q+w} = 1 - \frac{2q}{q+w}$ is an increasing function for w. Thus if $q \leq \sqrt{AB}$, $\mu(v) = \frac{q-A}{A+q}$. Otherwise, $\mu(v) = \frac{B-q}{B+q}$. □

Corollary 8.7. *The amplitude* $\mu(v)$ *of a uniform WVG* v *can be found in* $O(1)$*, i.e., constant, time.*

Proof. The corollary immediately follows from the previous theorem. □

We now look at unanimity WVGs, which are another important subclass of WVGs in which a coalition is winning if and only if it is the grand coalition N.

Proposition 8.8. *For a unanimity WVG* $v = [q; w_1, \ldots, w_n]$,

$$\tau(v) \leq \frac{w_n}{4w(N) - w_n} \leq \frac{1}{4n-1}.$$

Proof. We know that $B = w(N)$ and $A = w(N) - w_n$, which means that $w(N) - w_n < q \leq w(N)$. For maximum tolerance, $q = \frac{A+B}{2} = w(N) - \frac{w_n}{2}$. Therefore $m = w_n/2$ and $M = w(N) - \frac{w_n}{2} + w(N)$. Then the tolerance of v satisfies:

$$\tau(v) \leq \frac{m}{M} = \frac{w_n}{4w(N) - w_n} \leq \frac{1}{4n-1},$$

since $w_n \leq w(N)/n$. □

Note that we do not insist that $w_i^t \geq w_j^t$ for all $i < j$ and $1 \leq t \leq m$. Let $(N, v) = (N, v_1 \wedge ... \wedge v_m)$ be a multiple weighted voting game. Then we can see that $\mu(v) \geq \mathsf{Inf}(\mu(v_1), \ldots, \mu(v_m))$. The reason is that for v to change, at least one constituent game has to change. However it is not necessary that a change in any one game v_i changes v. As a simple example, suppose $v_1 = [2; 2, 1]$ and $v_2 = [2; 1, 2]$. Then $\mu(v_1 \wedge v_2) = 1/5$, as witnessed by the coalition $\{1, 2\}$. However, $\mu(v_i) = 0$, as witnessed by $\{i\}$, for $i = 1, 2$.

8.4 Conclusion and future work

We have examined the computational complexity of computing the tolerance and amplitude of WVGs. The tolerance and amplitude of uniform and unanimity games is also analysed. There is a need to devise approximation algorithms for computing the amplitude of a WVG. The analysis of amplitude and tolerance motivates the formulation of an overall framework to check the 'sensitivity' of voting games under fluctuations and susceptibility to control. It will be interesting to explore the limit of changes in WVGs in alternative representations of simple games.

9

False-name manipulations

...some voting procedures can be inherently resistant to abuse, while others are vulnerable. We base this distinction on a measure that is new to voting theory - computational complexity.
- Bartholdi, Tovey and Trick [114]

Life is not long, and too much of it should not be spent in idle deliberation.
- Samuel Johnson

Abstract We examine the computational complexity of false-name manipulation in *weighted voting games*. Weighted voting games have received increased interest in the multiagent community due to their compact representation and ability to model coalitional formation scenarios. Bachrach and Elkind in their AAMAS 2008 paper examined divide and conquer *false-name manipulation* in weighted voting games from the point of view of Shapley-Shubik index. We analyse the corresponding case of Banzhaf index and check how much the Banzhaf index of a player increases or decreases if it splits up into sub-players. A pseudo-polynomial algorithm to find the optimal split is also provided. Bachrach and Elkind also mentioned manipulation via merging as an open problem. In the chapter, we examine the cases where a player annexes other players or merges with them to increase their Banzhaf index or Shapley-Shubik index payoff. We characterize the computational complexity of such manipulations as well as providing limits to the manipulation. The Annexation Non-monotonicity paradox is also discovered

in the case of the Banzhaf index. The results give insight into coalition formation and manipulation.

9.1 Introduction

There are various ways WVGs can be manipulated and controlled. Splitting of a player into sub-players can be seen as a false-name manipulation by an agent where it splits itself into multiple agents so that the sum of the utilities of the split-up players is more than the utility of the original player [21]. We examine situations when splitting up into smaller players may be advantageous or disadvantageous to a player in the context of WVGs and Banzhaf indices. This gives a better idea of how to devise WVGs in which manipulation can be deterred. This may be done by keeping larger or non-integer weights. Moreover, we also examine the case of players merging to maximize their payoff in a WVG. This was mentioned as an unexplored question in [21].

The outline of the chapter is as following. Section 9.2 provides a brief literature survey. In Section 9.3, the case of players splitting up into sub-players in a WVG to increase their Banzhaf index is analysed. We examine the extent to which the Banzhaf index of a player can increase or decrease if it splits up into sub-players. From a computational perspective, it is #P-hard [179] to compute the payoff in the WVG. A prospective manipulator could still be interested in enabling a beneficial split even if he cannot compute the actual payoff. Moreover, this model is reasonable because the central authority which organizes the game is assumed to have much more computational resource than the players. In Section 9.4, we prove that it is NP-hard even to decide whether a split is beneficial or not. In the end a pseudo-polynomial algorithm is proposed which returns ‘no’ if no beneficial split is available and returns the optimal split otherwise. Section 9.5 is about the case of players annexing others or voluntarily merging into blocs to maximize their payoffs. It is shown that for both the Banzhaf index and the Shapley-Shubik index, it is NP-hard to find a beneficial merge and for the case of the Banzhaf index, it is NP-hard to decide a beneficial annexation. Limits to manipulation are also provided. The final section presents conclusions and ideas for future work.

9.2 Related work

WVGs have also been examined from the point of view of control and manipulation. Zuckerman et al. [220] analyse how the centre might control WVGs by changing the quota even if the weights are fixed. The most relevant work is by Bachrach and Elkind [21] where they examine false-name manipulation in WVGs from the point of view of the Shapley-Shubik index. In fact, this chapter answers problems posed by Bachrach and Elkind. Players forming blocs have been considered by political scientists and economists previously [82]. However, in this chapter, a complexity theoretic analysis of bloc forming manipulation has also been undertaken for WVGs. False name manipulations in open anonymous environments have been examined in different domains such as coalitional games [219, 162, 161] and auctions [218, 113]. The characteristic function by itself does not give enough information to analyze false-name manipulations especially if a player splits into sub-players. Therefore Yokoo et al. [219] introduced the model where each player has a subset of skills and the characteristic function assigns values to the subset of skills. We notice that false-name manipulations in WVGs can still be analyzed directly without considering more fine-grained representations.

9.3 Splitting

In the real world, WVGs may be dynamic. Players might have an incentive to split up into smaller players or merge into voting blocs. Payoffs of players in a coalitional games setting can be based on fairness, i.e., power indices, or they can be based on the notion of stability, which includes many cooperative game theoretic concepts such as core, nucleolus etc. We examine the situation when the Banzhaf indices of agents can be used as payoffs in a cooperative game theoretic situation. Falsenthal and Machover [146] refer to this notion of voting power as P-power since the motivation of agents is prize-seeking as opposed to influence-seeking. However Banzhaf indices have been considered as possible payments in cooperative settings [211, 21] and they satisfy desirable axioms [63]. Splitting of a player can be seen as a false-name manipulation by an agent, in which it splits itself into multiple agents so that the sum of the utilities of the split-up players is more than the utility of the original player [21].

Splitting is not always beneficial. We give examples where, if we use Banzhaf indices as payoffs of players in a WVG, splitting can be advantageous, neutral or disadvantageous.

Example 9.1. Splitting can be advantageous, neutral or disadvantageous:

- *Disadvantageous splitting*. In the WVG $[5; 2, 2, 2]$ each player has a Banzhaf index of $1/3$. If the last player splits up into two players, the new game is $[5; 2, 2, 1, 1]$. In that case, the split-up players have a Banzhaf index of $1/8$ each.
- *Neutral splitting*. In the WVG $[4; 2, 2, 2]$ each player has a Banzhaf index of $1/3$. If the last player splits up into two players, the new game is $[4; 2, 2, 1, 1]$. In that case, the split-up players have a Banzhaf index of $1/6$ each.
- *Advantageous splitting*. In the WVG $[6; 2, 2, 2]$ each player has a Banzhaf index of $1/3$. If the last player splits up into two players, the new game is $[6; 2, 2, 1, 1]$. In that case, the split-up players have a Banzhaf index of $1/4$ each.

We analyse the splitting of players in the unanimity WVG.

Proposition 9.2. *In a unanimity WVG with $q = w(N)$, if Banzhaf indices are used as payoffs of agents in a WVG, then it is beneficial for an agent to split up into several agents.*

Proof. In a WVG with $q = w(N)$, the Banzhaf index of each player is $1/n$. Let player i split up into $m + 1$ players. In that case there is a total of $n + m$ players and the Banzhaf index of each player is $1/(n + m)$. In that case the total Banzhaf index of the split up players is $\frac{m+1}{n+m}$, and for $n > 1$, $\frac{m+1}{n+m} > \frac{1}{n}$. □

An exactly similar analysis holds for Shapley-Shubik index. Players would only return to parity if they all split up into the same number of players.

We recall that a player is critical in a winning coalition if the player's exclusion makes the coalition losing. We will also say that a player is *critical for* a losing coalition C if the player's inclusion results in the coalition winning.

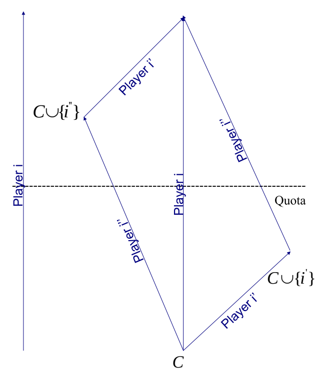

Fig. 9.1. Splitting of player i into i' and i'' .

Proposition 9.3. *Let WVG v be $[q; w_1, \ldots, w_n]$. If v transforms to v' by the splitting of player i into i' and i'', then*

$$\beta_{i'}(v') + \beta_{i''}(v') \leq 2\beta_i(v).$$

Moreover, this upper bound is asymptotically tight.

Proof. We assume that a player i splits up into i' and i'' and that $w_{i'} \leq w_{i''}$. We consider a losing coalition C for which i is critical in v (see Figure 9.1). The left hand vertical arrow shows the total weight of player i . We see that $w(C) < q \leq w(C) + w_i = w(C) + w_{i'} + w_{i''}$.

- If $q - w(C) \leq w_{i'}$, then i' and i'' are critical for C in v'.
- If $w_{i'} < q - w(C) \leq w_{i''}$, then i' is critical for $C \cup \{i''\}$ and i'' is critical for C in v'. (This case is shown in Figure 9.1.)

- If $q - w(C) > w_{i''}$, then i' is critical for $C \cup \{i''\}$ and i'' is critical for $C \cup \{i'\}$ in v'.

We see that for any swing for player i, there are is an addition of exactly two swings for the players i' and i''. Also if i is not critical for coalition, then neither can i' and i be critical for that coalition. Therefore we have $\eta_{i'}(v')+\eta_{i''}(v') = 2\eta_i(v)$ in each case.

Now we consider a player x in v which is other than player i. If x is critical for a coalition C in v then x is also critical for the corresponding coalition C' in v' where we replace $\{i\}$ by $\{i', i''\}$. Hence $\eta_x(v) \leq \eta_x(v')$. Of course x may also be critical for some coalitions in v' which contain just one of i' and i'', so the above inequality will not in general be an equality. Moreover,

$$\begin{aligned}
\beta_{i'}(v') + \beta_{i''}(v') &= \frac{2\eta_i(v)}{2\eta_i(v) + \sum_{x \in N(v') \setminus \{i', i''\}} \eta_x(v')} \\
&\leq \frac{2\eta_i(v)}{2\eta_i(v) + \sum_{x \in N(v) \setminus \{i\}} \eta_x(v)} \\
&\leq \frac{2\eta_i(v)}{\eta_i(v) + \sum_{x \in N(v) \setminus \{i\}} \eta_x(v)} = 2\beta_i(v)
\end{aligned}$$

We can prove that this coefficient of 2 is best possible asymptotically. We take a WVG $[n; 2, 1, \ldots, 1]$ with $n + 1$ players. We find that $\eta_1 = n + \binom{n}{2}$ and for all other x, $\eta_x = 1 + \binom{n-1}{2}$. Therefore

$$\beta_1 = \frac{n + \binom{n}{2}}{n + \binom{n}{2} + n(1 + \binom{n-1}{2})} = \frac{n+1}{(n-2)^2} \sim 1/n.$$

In case player 1 splits up into $1'$ and $1''$ with weights 1 each, then for all players $j, \beta_j = \frac{1}{n+2}$. Thus for large $n, \beta_{1'} + \beta_{1''} = \frac{2}{n+2} \sim 2\beta_1$. $\square$

It can be shown [67] that in the case of Banzhaf index, the disadvantage in splitting into two players cannot be more than by a factor of $1/(n + 1)$:

Proposition 9.4. *Let WVG v be $[q; w_1, \ldots, w_n]$. If v transforms to v' by the splitting of player i into i' and i'', then $\beta_{i'}(v') + \beta_{i''}(v') \geq \frac{\beta_i(v)}{n+1}$.*

Proof. Consider $S \subseteq N \setminus \{i, x\}$. If player x distinct from i is critical for S, then in the extreme case it may possibly be critical for S, $S \cup \{i'\}$, $S \cup \{i''\}$ and $S \cup \{i', i''\}$. Assume that x is not critical for S but x is critical for $S \cup \{i'\}$. This implies that $v'(S) = 0$, $v'(S \cup \{x\}) = 0$ and $v'(S \cup \{x, i'\}) = 1$ which means that i' is also critical for $S \cup \{x\}$. Then, new coalitions of the form $S \cup \{i'\}$ or $S \cup \{i''\}$ produced for which x is critical cannot be more than $\eta'_i(v') + \eta''_i(v') = 2\eta_i(v)$. Then we have,

$$
\begin{aligned}
\beta_{i'}(v') + \beta_{i''}(v') &= \frac{2\eta_i(v)}{2\eta_i(v) + \sum_{x \in N(v') \setminus \{i', i''\}} \eta_x(v')} \\
&\geq \frac{2\eta_i(v)}{2\eta_i(v) + 4\sum_{x \in N(v) \setminus \{i\}} \eta_x(v) + 2(n)\eta_i(v)} \\
&\geq \frac{2\eta_i(v)}{2\eta_i(v) + 2(n+1)\sum_{x \in N(v) \setminus \{i\}} \eta_x(v) + 2(n)\eta_i(v)} \\
&= \frac{\beta_i(v)}{n+1}
\end{aligned}
$$

□

It is believed that the lower bound proved in Proposition 9.4 is not tight. We now show that splitting into two players can decrease the Banzhaf index payoff by as much as a factor of almost $\sqrt{\frac{\pi}{2n}}$:

Example 9.5. Disadvantageous splitting. We take a WVG v on n players where $v = [3n/2; 2n, 1, \ldots, 1]$. For the sake of simplicity, we assume that n is even. It is easy to see that player 1 is a dictator. Now we consider the case where v changes into v' with player 1 splitting up into $1'$ and $1''$ with weight n each. For player $1'$ to be critical for a losing coalition in v', the coalition much exclude $1''$ and have from $n/2$ to $n-1$ players with weight 1 or it must include $1''$ and have from 0 to $(n/2 - 1)$ players with weight 1. So $\eta_{1'}(v') = \eta_{1''}(v') = \sum_{i=0}^{n} \binom{n-1}{i} = 2^{n-1}$. Moreover, for a smaller player x with weight 1 to be critical for a coalition in v', the coalition must include only one of $1'$ or $1''$ and $(n-2)/2$ of the $n-2$

other smaller players. So, $\eta_x(v') = 2\binom{n-2}{(n-2)/2}$. By using Stirling's formula, we can approximate $\eta_x(v')$ by $\sqrt{\frac{2}{\pi(n-2)}}2^{n-1}$. We see that:

$$\begin{aligned}
\beta_{i'}(v') = \beta_{i''}(v') \\
&\approx \frac{2^{n-1}}{2^{n-1} + 2^{n-1} + (n-1)\sqrt{\frac{2}{\pi(n-2)}}2^{n-1}} \\
&= \frac{1}{2 + \frac{(n-1)}{\sqrt{n-2}}\sqrt{\frac{2}{\pi}}} \\
&\sim \sqrt{\frac{\pi}{2n}}.
\end{aligned}$$

We notice that the bounds on the effect of splitting on the Banzhaf index are quite similar to those in the Shapley-Shubik case (see Table 9.2).

9.4 Complexity of finding a beneficial split

From a computational perspective, it is #P-hard for a manipulator to find the ideal splitting to maximize his payoff. This is because even computing Banzhaf values once for a WVG is #P-complete.

An easier question is to check whether a beneficial split exists or not. We define a Banzhaf version of the BENEFICIAL SPLIT problem defined in [21].

Name: BENEFICIAL-BZ-SPLIT
Instance: (v, i) where v is the WVG $v = [q; w_1, \ldots, w_n]$ and player $i \in \{1, \ldots, n\}$.
Question: Is there a way for player i to split his weight w_i between sub-players $i_1, \ldots, i_m$ so that, in the new game v', $\sum_{j=1}^{m} \beta_{i_j}(v') > \beta_i(v)$?

Proposition 9.6. *BENEFICIAL-BZ-SPLIT is NP-hard, and remains NP-hard even if the player can only split into two players with equal weights.*

Proof. We prove this by a reduction from an instance of the classical NP-hard PARTITION problem to BENEFICIAL-BZ-SPLIT.

Given an instance of PARTITION $\{a_1, \ldots, a_k\}$, we can transform it to a WVG $v = [q; w_1, \ldots, w_n]$ with $n = k + 1$ where $w_i = 8a_i$ for $i = 1$ to $n - 1$, $w_n = 2$ and $q = 4\sum_{i=1}^{k} a_i + 2$. After that, we want to see whether it can be beneficial for player n with weight 2 to split into two sub-players n and $n + 1$ each with weight 1 to form a new WVG $v' = [q; w_1, \ldots, w_{n-1}, 1, 1]$. Note that, since the weights are integral, it is certainly not possible to split up a weight of 2 other than into 1 and 1.

If A is a 'no' instance of PARTITION, then we see that no subset of the weights $\{w_1, \ldots, w_{n-1}\}$ can sum to $4\sum_i a_i$. This implies that player n is a dummy. We see that even if player n splits into sub-players, the sub-players are also dummies. Therefore (v, n) is a 'no' instance of BENEFICIAL-BZ-SPLIT.

Now let us assume that A is a 'yes' instance of PARTITION. In that case, let the number of subsets of weights $\{w_1, \ldots w_{n-1}\}$ summing to $4\sum_i a_i$ be x. Then $\eta_n(v) = x$. For $i \leq n - 1$, player i can be critical in winning coalition with weight exactly q or more than q. We note that exactly half of the x subsets of $\{w_1, \ldots w_{n-1}\}$ summing to $4\sum_i a_i$ contain w_i. If player i is critical in a coalition C which is a subset of $\{w_1, \ldots w_{n-1}\}$ then i is also critical in $C \cup \{w_n\}$. Therefore for $i \leq n - 1$, $\eta_i(v) = \frac{x}{2} + 2y_i$ where y_i is the number of subsets of $\{w_1, \ldots w_{n-1}\}$ in which i is critical. We see that

$$\beta_n(v) = \frac{x}{x + \frac{kx}{2} + 2y} \text{ where } \sum_{i \leq n-1} y_i = y.$$

However, in the new game v', $\eta_n(v') = \eta_{n+1}(v') = x$ and for $i \leq n - 1$, $\eta_i(v') = \frac{x}{2} + 4y_i$, since there are now 4 coalitions, $C, C \cup \{w_n\}, C \cup \{w_{n+1}\}, C \cup \{w_n, w_{n+1}\}$, corresponding to each C. So

$$\beta_n(v') + \beta_{n+1}(v') = \frac{2x}{2x + kx/2 + 4y} > \beta_n(v),$$

since $x > 0$. Thus, a 'yes' instance of PARTITION implies a 'yes' instance of BENEFICIAL-BZ-SPLIT. □

In terms of minimizing chances of manipulation, we see that computational complexity acts as a barrier. This idea of using computational complexity to model bounded rationality is well explained by Papadimitriou and Yannakakis [170]. In the context of complexity of voting, it was a series of groundbreaking papers by Bartholdi, Orlin, Tovey, and Trick [32, 111, 112, 114] that showed how important computationally complexity consideration is in terms of ease of computing winners and difficulty of manipulation.

9.4.1 Pseudo-polynomial algorithm

It is well known that, although computing Banzhaf indices of players in a WVG is NP-hard, there are polynomial time algorithms using dynamic programming [151] or generating functions [36] to compute Banzhaf indices if the weights of players are polynomial in n. Let this pseudo-polynomial algorithm be $\mathsf{BanzhafIndex}(v, i)$ which takes a WVG v and an index i as input and returns $\beta_i(v)$, the Banzhaf index of player i in v. We use a similar argument as in [21] to show that a polynomial algorithm exists to find a beneficial split if the weights of players are polynomial in n and the player i in question can split into up to a constant k number of sub-players with integer weights. Algorithm 11 takes as input a WVG v and player i which can split into a maximum of k number of players. The algorithm returns 'no' if no beneficial split exists and returns the optimal split otherwise. Whenever player i in WVG v splits according to a split s, we denote the new game by $v_{i,s}$.

Proposition 9.7. *Algorithm 11 is polynomial in n if the weights are polynomial in n.*

Proof. Since the weight values are polynomial in n and we consider splits into a constant number of players, a brute force method is sufficient. We see that the total number of splits for player i is equal to $q(w_i, k)$ where $q(n, k)$ is the partition function which gives the number of partitions of n with k or fewer addends. It is

Algorithm 11 BeneficialSplitInWVG

Input: (v, i) where $v = [q; w_1, \ldots, w_n]$ and i is the player which wants to split into a maximum of k sub-players.
Output: Returns NO if there is no beneficial split. Otherwise returns the optimal split $(w_{i_1}, \ldots, w_{i_{k'}})$ where $k' \leq k$, and $\sum_{j=1}^{k'} w_{i_j} = w_i$.

BeneficialSplitExists = **false**
BestSplit = $\emptyset$
BestSplitValue = $-\infty$
β_i = BanzhafIndex(v, i)
for $j = 2$ to k **do**
  for Each possible split s where $w_i = w_{i_1} + \ldots + w_{i_j}$ **do**
    SplitValue = $\sum_{a=1}^{j}$ BanzhafIndex$(v_{i,s}, i_a)$
    if SplitValue $> \beta_i$ **then**
      BeneficialSplitExists = **true**
      if SplitValue > BestSplitValue **then**
        BestSplit = s
        BestSplitValue = SplitValue
      end if
    end if
  end for
end for
if BeneficialSplitExists = **false then**
  return false
else
  return BestSplit
end if

clear that for a constant k, the number of splits of player i is less than $(w_i)^k$ which is a polynomial in n. Since the computational complexity for each split is also a polynomial in n, therefore Algorithm 11 is polynomial in n if the weights are polynomial in n. □

9.5 Merging and annexation

For the case of players merging to gain advantage, we examine two cases. One is annexation where one voter takes the voting weight of other players. The annexation is advantageous if the payoff of the new merged coalition in the new game is greater than the payoff of the annexer in the original game. The other case is voluntary merging where players merge to become a bloc for which their new payoff is more than the sum of their individual payoffs. For every game (N, v), the result of the merging of players in coalition S is another game $((N \setminus S) \cup \{\&S\}, v_{\&S})$.

We define the problem of checking a beneficial voluntary merge or annexation:

Name: BENEFICIAL-BZ-MERGE
Instance: (v, S) where v is the WVG $v = [q; w_1, \ldots, w_n]$ and $S \subset N$.
Question: Suppose coalition S merges to form a new game $((N \setminus S) \cup \{\&S\}, v_{\&S})$. Is $\beta_{\&S}(v_{\&S}) > \sum_{i \in S} \beta_i(v)$?

Name: BENEFICIAL-BZ-ANNEXATION
Instance: (v, S, i) where v is the WVG $v = [q; w_1, \ldots, w_n]$, i is the ith player in v and $S \subset (N \setminus \{i\})$.
Question: If i annexes coalition S to form a new game $((N \setminus (S \cup \{i\})) \cup \{\&(S \cup \{i\})\}, v_{\&(S \cup \{i\})})$, is $\beta_i(v_{\&(S \cup \{i\})}) > \beta_i(v)$?

If Shapley-Shubik indices are used as payoffs in place of Banzhaf indices, then the corresponding problems are defined with BZ replaced by SS so that BENEFICIAL-SS-MERGE corresponds to BENEFICIAL-BZ-MERGE. Felsenthal and Machover [82] prove that if a player annexes other players, then it cannot be the case that the annexation is disadvantageous if the Shapley-Shubik indices are used as payoffs. We provide a clear and simple proof of this theorem. Let player i be critical for a coalition S in WVG v. Then the contribution to $\phi_i(v)$ from this is $\frac{(|S|-1)!(n-|S|)!}{n!}$. We consider a game $v_{\&\{i,j\}}$ where i annexes j. For every S for which i is critical in v, the contribution to $\phi_{\&\{i,j\}}(v_{\&\{i,j\}})$ is either $\frac{(|S|-2)!(n-|S|)!}{(n-1)!}$ or $\frac{(|S|-1)!(n-|S|-1)!}{(n-1)!}$. For either case we see that $\phi_{\&\{i,j\}}(v_{\&\{i,j\}}) > \phi_i(v)$. However Felsenthal and Machover [82] show that, for the case of the Banzhaf index, annexation could be disadvantageous. They provide a 13-player WVG for which annexation is disadvantageous, which is the simplest example they could find. We provide an 8-player WVG where annexation is disadvantageous:

Example 9.8. In WVG $[13; 7, 6, 1, 1, 1, 1, 1, 1]$, player 1 has Banzhaf index 0.48507. If player 1 annexes one of the small players, the new game is $[13; 8, 6, 1, 1, 1, 1, 1]$ and the Banzhaf index becomes 0.47826.

For the case where the merging is voluntary instead of an annexation, for both the Banzhaf index and Shapley-Shubik index, merging can be advantageous or disadvantageous. As in the case of splitting, we expect it to be hard to find a beneficial merge:

Proposition 9.9. *BENEFICIAL-BZ-MERGE is NP-hard.*

Proof. Given an instance of PARTITION $\{a_1, \ldots, a_k\}$, we can transform it to a WVG $v = [q; w_1, \ldots, w_n]$ where $n = k + 3$, $w_i = 8a_i$ for $i = 1$ to $n - 3$, $w_{n-2} = w_{n-1} = w_n = 1$, and $q = 4\sum_{i=1}^{k} a_i + 2$.

If A is a 'no' instance of PARTITION, then we see that a subset of weights $\{w_1, \ldots w_{n-3}\}$ cannot sum to $4\sum_i a_i$. This implies that players $(n-2), (n-1)$ and n are dummies. Even if players n and $(n-1)$ merge together, the new player $\&\{n-1, n\}$ remains a dummy in the new game $v_{\&\{n-1,n\}}$.

Now let us assume that A is a 'yes' instance of PARTITION. In that case, let the number of subsets of weights $\{w_1, \ldots w_{n-3}\}$ summing to $4\sum_i a_i$ be x. For $i \leq n-3$, player i can be critical in winning coalitions with weight q or $q+1$ or more than $q+1$. The number of coalitions for the first two cases are $3x/2$ and $x/2$, respectively, corresponding to the participation of either 2 or 3 of the unit players. The third case corresponds to coalitions in which the three unit players are dummies. Therefore for $i \leq n-3$, $\eta_i = \frac{4x}{2} + 8y_i$ where y_i is the number of subsets of $\{w_1, \ldots, w_{n-3}\}$ in which i is critical. Moreover, $\eta_{n-2}(v) = \eta_{n-1}(v) = \eta_n(v) = 2x$, since each unit player is critical only when exactly one other of these is in the coalition. Then

$$\beta_n(v) = \frac{2x}{6x + \frac{4kx}{2} + 8y}, \text{ where } \sum_{i \leq n-3} y_i = y.$$

In the new game $v_{\&\{n-1,n\}}$, $\eta_{\&\{n-1,n\}}(v_{\&\{n-1,n\}})$ is $2x$ but $\eta_{n-2}(v_{\&\{n-1,n\}})$ is 0. For $i \leq n-3$, $\eta_i(v_{\&\{n-1,n\}})$ is $\frac{2x}{2} + 4y_i$. We see that

$$\beta_{\&\{n-1,n\}}(v_{\&\{n-1,n\}}) = \frac{2x}{2x + 2kx/2 + 4y}.$$

Therefore,

$$\beta_{\&\{n-1,n\}}(v_{\&\{n-1,n\}}) > \beta_n(v) + \beta_{n-1}(v),$$

which means that n and $(n-1)$ had a beneficial merge. It has been shown that a 'yes' instance of PARTITION implies a 'yes' instance of BENEFICIAL-BZ-MERGE. □

Proposition 9.10. *BENEFICIAL-BZ-ANNEXATION is NP-hard.*

Proof. Given an instance of PARTITION, $\{a_1, \dots, a_k\}$, we can transform it to a WVG $v = [q; w_1, \dots, w_n]$ where $n = k + 2$, $w_i = 8a_i$ for $i = 1$ to $n-2$, $w_{n-1} = 1$, $w_n = 1$ and $q = 4\sum_{i=1}^{k} a_i + 2$. Just as in Proposition 9.9, we see that a 'no' instance of partition implies that w_{n-1} and w_n are dummies even if n annexes $(n-1)$. However, a 'yes' instance of partition implies that player n benefits by annexing player $(n-1)$. □

Proposition 9.11. *BENEFICIAL-SS-MERGE is NP-hard*

Proof. Given an instance of PARTITION $\{a_1, \dots, a_k\}$, we can transform it to a WVG $v = [q; w_1, \dots, w_n]$ where $n = k + 3$, $w_i = 8a_i$ for $i = 1$ to $n-2$, $w_{n-2} = w_{n-1} = w_n = 1$, and $q = 4\sum_{i=1}^{k} a_i + 2$.

If A is a 'no' instance of PARTITION, then we see that a subset of weights $\{w_1, \dots w_{n-3}\}$ cannot sum to $4\sum_i a_i$. This implies that players $(n-2)$, $(n-1)$ and n are dummies. Even if player n and $(n-1)$ merge together, the new player $\&\{n-1, n\}$ remains a dummy in the new game $v_{\&\{n-1,n\}}$.

Now let us assume that A is a 'yes' instance of PARTITION. For each partition (P_1, P_2) where $|P_1| = p_1$ and $|P_2| = p_1$, we check the number of permutations corresponding to (P_1, P_2). In the original game v, the contribution to the Shapley-Shubik payoff for either player n or $(n-1)$ by the permutations corresponding to (P_1, P_2) is

$$\frac{2(p_1+1)!(p_2+1)!}{n!} = \frac{p_1!p_2!}{n!}2(p_1+1)(p_2+1).$$

If players n and $n-1$ merge into bloc $\&\{n-1,n\}$, then the contribution to the Shapley-Shubik payoff to bloc $\&\{n-1,n\}$ by the permutations corresponding to (P_1,P_2) is

$$\frac{p_1!(p_2+1)!+(p_1+1)!p_2!}{(n-1)!} = \frac{p_1!p_2!}{n!}(n(p_1+1+p_2+1)).$$

For the merge to be beneficial, it is required that the sum of the Shapley-Shubik indices of $(n-1)$ and n in the original game v is less than the Shapley-Shubik index of $\&\{n-1,n\}$ in the game $v_{\&\{n-1,n\}}$, i.e., $4(p_1+1)(p_2+1) < n(p_1+1+p_2+1)$. Since $(p_1+1)+(p_2+1) = n-1$, we have

$$4(p_1+1)(p_2+1) \leq 4\left(\frac{n-1}{2}\right)^2 < n(n-1) = n(p_1+1+p_2+1),$$

and so $\phi_{n-1}(v)+\phi_n(v) < \phi_{\&\{n-1,n\}}(v_{\&\{n-1,n\}})$. $\square$

We examine the limits of advantage or disadvantage for the case of the annexation of another player to increase the Banzhaf index.

Proposition 9.12.

$$\frac{\beta_i(v)+\beta_j(v)}{2} \leq \beta_i(v_{\&(\{i,j\})}) \leq 1.$$

Proof. Let v be WVG $[q; w_1,\ldots,w_n]$. Suppose i annexes or merges with player j and v' is $v_{\&(\{i,j\})}$. Then the new game is $((N\setminus\{j\})\cup\{\&(\{i,j\})\}, v')$. From the proof of Proposition 9.3, we see that $\eta_{\&(\{i,j\})}(v')$ equals $\frac{1}{2}(\beta_i(v)+\beta_j(v))$.

Now consider a player x which is other than player i or player j. Let S be coalition such that $S \subseteq N\setminus\{i,j,x\}$. If x is critical for S in v then x is critical for S in v'. If x is critical for $S\cup\{i,j\}$ in v then x is critical for $S\cup\&(\{i,j\})$ in v'. However, x may also be critical for $S\cup\{i\}$ or $S\cup\{j\}$ in v. So $\eta_x(v) \geq \eta_x(v')$. We see that:

$$\begin{aligned}
\beta_{\&(\{i,j\})}(v') &= \frac{\eta_{\&(\{i,j\})}(v')}{\eta_{\&(\{i,j\})}(v') + \sum_{x\in(N\setminus\{i,j\})} \eta_x(v')} \\
&= \frac{\frac{1}{2}(\eta_i(v) + \eta_j(v))}{\frac{1}{2}(\eta_i(v) + \eta_j(v)) + \sum_{x\in(N\setminus\{i,j\})} \eta_x(v')} \\
&\geq \frac{\frac{1}{2}(\eta_i(v) + \eta_j(v))}{\eta_i(v) + \eta_j(v) + \sum_{x\in(N\setminus\{i,j\})} \eta_x(v)} \\
&= \frac{1}{2}(\beta_i(v) + \beta_j(v)).
\end{aligned}$$

The upper bound is tight and easy to observe. If player i is a dummy and j is a dictator then $\beta_i(v) = 0$ whereas $\beta_i(v') = 1$. The upper bound can also be achieved by two big enough players joining forces. □

We have seen that annexation can be disadvantageous in the case of the Banzhaf index. One would at least expect the Banzhaf index payoff after annexing another player to be monotone in the power of the annexed player. Surprisingly, this is not the case. Suppose $w_i \geq w_j \geq w_k$ in a WVG v. We provide an example where $\beta_{i,k} > \beta_{i,j}$. We call this the *annexation non-monotonicity paradox*:

Example 9.13. In the WVG $[9; 3, 3, 2, 1, 1, 1]$ we see that player 2 has more weight than player 3. However if player 1 annexes player 2 to form game $[9; 6, 2, 1, 1, 1]$, its Banzhaf index is 0.4, whereas if player 1 annexes player 3 to form game $[9; 5, 3, 1, 1, 1]$, its Banzhaf index is $7/17 \approx 0.411765$.

Proposition 9.14. *For any coalition, $S \subset N \setminus \{i\}$,*

$$\phi_i(v) \leq \phi_i(v_{\&(\{i\}\cup S)}) \leq 1.$$

Proof. The lower bound follows from the result by Felsenthal and Machover [82] that annexation cannot decrease the Shapley-Shubik index of a player. Moreover, the upper bound is tight and easily attainable if $\{i\} \cup S$ is big enough. □

Proposition 9.15. *For the unanimity game, for both the Shapley-Shubik index and Banzhaf index:*

Table 9.1. Complexity of false name manipulations in WVGs

	Banzhaf index	Shapley-Shubik index
SPLITTING	NP-hard	NP-hard [21]
MERGING	NP-hard	NP-hard
ANNEXATION	NP-hard	advantageous [82]
SPLITTING in unanimity game	advantageous	advantageous [21]
MERGING in unanimity game	disadvantageous	disadvantageous
ANNEXATION in unanimity game	advantageous	advantageous

1. it is disadvantageous for a coalition to merge;

2. it is advantageous for a player to annex.

Proof. We check each case separately:

1. This is expected considering Proposition 9.2. If k players merge, then the payoff of the new coalition is $1/(n-k+1)$. It is easy to see that $1/(n-k+1) < k/n$.
2. For a unanimity WVG with n players, the payoff of each player is $1/n$. If a player annexes $k-1$ other players, its payoff is $1/(n-k+1)$ which is more than $1/n$.

□

In a WVG, if player i annexes a dummy, then there is no difference to the Banzhaf index payoff of each player. This is because the Banzhaf value of each player reduces to half of the original Banzhaf value. Moreover, it follows from Proposition 9.12 that if a player annexes a player bigger than itself, its Banzhaf index can only increase. Thus annexation could only be disadvantageous, if a player annexes a smaller player. Although deciding a beneficial merge or annexation is computationally difficult, it may often be easier in practice. We propose a simple heuristic to get beneficial annexations or at least to avoid disadvantageous annexations. It appears to be a better strategy to annex fewer players with some total weight than more players with the same total weight. This is because, while annexing, the annexer does not want to increase the payoff of other players significantly.

Bounds	Reference
$\frac{1}{n+1}\beta_i(v) \leq \beta_{i'}(v') + \beta_{i''}(v') \leq 2\beta_i(v)$	Prop 9.4 and 9.3
$\frac{\beta_i(v)+\beta_j(v)}{2} \leq \beta_i(v_{\&(\{i,j\})}) \leq 1.$	Prop 9.12
$\phi_i(v) \leq \phi_i(v_{\&(\{i\}\cup S)}) \leq 1.$	Prop 9.14
$\frac{2}{n+1}\phi_i(v) \leq \phi_{i'}(v') + \phi_{i''}(v') \leq \frac{2n}{n+1}\phi_i(v)$	[21]

Table 9.2. Bounds of false-name manipulations in WVGs

9.6 Conclusions

We have investigated the impact on the Banzhaf power distribution due to a player splitting into smaller players in a weighted voting game. We have also considered the case of manipulation via annexation and voluntary merging when the payoff is according to the Banzhaf index or the Shapley-Shubik index. Both the complexity of manipulation and the limits of manipulation are examined. The complexity results are summarised in Table 9.1. In the table, whenever, the complexity of manipulation is NP-hard, there is possibility of the manipulation proving advantageous or disadvantageous. Table 9.2 summarizes the bounds of false-name manipulations in WVGs. The Shapley-Shubik index appears to be a more desirable solution for resource allocation because annexation does not decrease the payoff of a player. It is seen that manipulation may be discouraged by keeping weights which are large or non-integers. The finer, more detailed, analysis for players splitting into more than two players or merging into bigger blocs is still unexplored. Although it is NP-hard to evaluate different false-name manipulations, it may be the case that certain instances of WVGs are more susceptible to manipulation [16]. A careful investigation of heuristics for false-name manipulation is also a promising area of research. There is scope to analyse such false-name manipulations with respect to other cooperative game-theoretic solutions. A particularly suitable solution to consider could be the nucleolus which not only always exists but is also unique. Further examination into various aspects of manipulation in weighted voting games promises to give better insight into designing fairer and manipulation-resistant systems. Another interesting question is to what extent can the results be applied to more general cooperative games.

10

Complexity of length, duality and bribery

The people who cast the votes don't decide an election, the people who count the votes do.
- Joseph Stalin

Theory is to practice as rigor is to vigor.
- Donald Knuth

Abstract Coalitional voting games, especially simple games have received increased interest within the agents community recently. Length and width are important characteristics of coalitonal voting games which indicate efficiency of making a decision. Duality theory also plays an important role in artificial intelligence. In this chapter, the complexity of problems concerning the length, width and minimal winning coalitions of simple games is analysed. The complexity of questions related to duality of simple games such as DUAL, DUALIZE and SELF-DUAL is also examined. The possible representations considered are simple games represented by winning coalitions, minimal winning coalitions, a weighted voting game or a multiple weighted voting game. Since susceptibility to manipulation is a major issue in multiagent systems, it is observed that the results obtained have direct bearing on susceptibility to optimal bribery in simple games.

10.1 Introduction

10.1.1 Background

In this chapter we utilize the concepts of length and width of simple games to study the bribery manipulation in simple games. Length and width of threshold function were first considered in the electrical engineering and threshold logic literature. Duality theory also plays an important role in artificial intelligence. We analyse the computational complexity of certain questions related to length, width and duality of simple games. It is seen that answers to the questions have direct bearing on complexity results on bribery in simple games. Here, bribery is considered as buying loyalties of players to either enable a decision or prevent a decision. For reference to various notions and classes of computational complexity, please see [166].

We now present key definitions needed in this chapter.

Definition 10.1. *A coalition S is* blocking *if $N \setminus S$ is losing. We denote by $B(v)$ the set of blocking coalitions of v. For a simple game $v = (N, W)$, there is a* dual *game $v^d = (N, W(v^d))$ where $W(v^d)$ is equal to $B(v)$. A game v is* self-dual *if $v = v^d$.*

We can say that a coalition is winning in a simple game if and only if it is blocking in the dual of the game.

Definition 10.2. *A minimal blocking coalition (MBC) is a blocking coalition such that removal of any player makes it a non-blocking coalition. The set of MBCs of a simple game v is denoted by B^m.*

It is easy to see that the set of MWCs for v is equal to the MBCs for v^d. For self-dual simple games, there is an easy characterization: A simple game is self-dual if and only if for any coalition $S \subseteq N$, either S is a winning coalition or $(N \setminus S)$ is a winning coalition but not both. The reasoning for this is as follows. If v is self-dual, then coalition S is winning if and only if S is blocking which is equivalent to $(N \setminus S)$ not being winning. If S is a winning coalition, then it is known that $(N \setminus S)$ is losing. This means that S is blocking. Similarly if S is blocking, then this implies that S is winning. This clarifies that self-dual games and decisive games are equivalent.

We give a brief outline of the chapter. In Section 10.2, computational aspects of computing the length of simple games are considered. Section 10.3 examines the complexity of questions related to duality of games. In Section 10.4, complexity of bribery in simple games is explored in the light of previous sections. In Section 10.5, a summary of results and future directions of study are given. Throughout, we assume that the voting games have integer weights.

10.2 Computing length of games

Now, game theoretic version of definitions from Ramamurthy's book [184] are provided:

Definitions 10.3. *The set $W^k(v)$ is the set of winning coalitions of size k of the simple game v. Moreover, $B^k(v)$ is the set of blocking coalitions of v of size k.*

These definitions can be used to define the length and width of simple games:

Definitions 10.4. *The* length *of a simple game is the smallest integer k such that $W^k(v) \neq \emptyset$. The* width *of a simple game is the smallest integer k such that $B^k(v) \neq \emptyset$*

The length is an important indicator of a game which signifies in a sense the ease with which the status quo can be changed. We examine the complexity of computing the length of a simple game.

Name: LENGTH
Instance: Simple game v
Output: Length of v

Name: WIDTH
Instance: Simple game v
Output: Width of v

10.2.1 Complexity of computing length

It is evident that LENGTH(v) is equivalent to WIDTH(v^d). Moreover, for some special types of simple games, it is easy to observe their length and the width:

Observation 10.5 *For a simple game v,*

1. *If v is a unanimity game, $LENGTH(v) = n$ and $WIDTH(v) = 1$.*
2. *If v is a singleton game, $LENGTH(v) = 1$ and $WIDTH(v) = n$.*
3. *If v is a majority game, $LENGTH(v) = \lceil n/2 \rceil$ and $WIDTH(v) = \lceil (n+1)/2 \rceil$*

Proof. (Follows from the definitions). □

Now the complexity of computing LENGTH for a simple game represented by (N, W), (N, W^m), WVG or MWVG is analysed:

Observation 10.6 *The problem LENGTH for a simple game represented by (N, W), (N, W^m) or WVG is in P.*

Proof. For a simple game v represented by (N, W) or (N, W^m), LENGTH(v) can be computed in linear time by scanning the winning coalitions and identifying the smallest k such that coalition S is in W or W^m and $|S| = k$.

For the case of WVG, the weights of the players are already sorted. So start off with w_1 and keep adding more players with decreasing weights until $\sum_{i=1}^{k} w_i \geq q$. It is then claimed that LENGTH(v) is k. It is easy to see this since any other approach, apart from the greedy approach to pick up weights, will require at least k weights for the sum of the weights to be more than q. The greedy method outlined for LENGTH(v) for WVGs also computes the coalition which has the smallest feasible length. □

Proposition 10.7. *The problem LENGTH for a simple game represented by a MWVG is NP-hard.*

Proof. We provide a reduction from a special case of the minimization version of multidimensional 0-1 knapsack problem (MKP) [92].

Name: MKP
Instance: A collection of n items and m knapsacks where the capacity of the ith knapsack is b_i, the jth item requires a_{ij} units of resource consumption in the ith knapsack and has corresponding profit c_j
Output: Maximize $\sum_{j=1}^{n} c_j x_j$ such that $\sum_{j=1}^{n} a_{ij} x_j \leq b_i$, $i \in M = \{1, 2, \ldots m\}$, and $x_j \in \{0, 1\}$, $j \in N = \{1, 2, \ldots n\}$.

The goal in MKP is to find a subset of items that yields maximum profit without exceeding the resource capacities. MKP is equivalent to the minimization version of the problem (MIN-MKP) since maximizing the profit of a set of items is equivalent to minimizing the profit of items not in the set. The transformations needed are $y_j = 1 - x_j$ for $j \in N$ and $d_i = (\sum_{j=1}^{n} a_{ij}) - b_i$ for $j \in N$ and $i \in M$. Therefore the following problem is as hard as MKP:

Name: MIN-MKP
Instance: A collection of n items and m knapsacks where each knapsack i should have at least d_i capacity filled and the jth item has corresponding profit c_j and requires a_{ij} units of resource consumption.
Output: Minimize $\sum_{j=1}^{n} c_j y_j$ such that $\sum_{j=1}^{n} a_{ij} y_j \geq d_i$, $i \in M = \{1, 2, \ldots m\}$ and $y_j \in \{0, 1\}$, $j \in N$.

Gens and Levner [96] point out that Dinic and Karzanov [61] proved that even the special case of MIN-MKP where $m = 2$ and $c_j = 1$ for all $j = 1$ to n is NP-hard. It is easy to see that by renaming some variables (a_{ij} to w_j^i and d_i to q^i), the NP-hard special case of MIN-MKP is equivalent to computing the length of a MWVG of dimension 2. $\square$

10.2.2 Approximating the length of a MWVG

Although this may seem a paradox, all exact science is dominated by the idea of approximation.
- Bertrand Russell

Although all NP-complete problems share the same worstcase complexity, they have little else in common. When seen from almost any other perspective, they resume their healthy, confusing diversity. Approximability is a case in point.
- Christos Papadimitriou (1993)

Although the length of a MWVG cannot be computed efficiently, it is observed that it can be approximated efficiently:

Proposition 10.8. *For a MWVG, v with dimension m, there exists a polynomial time approximation algorithm which computes LENGTH(v) with an absolute error of* $m - 1$.

Proof. This result uses the same approach as in [34] where the authors use LP-relaxation to provide an $m-1$ absolute approximation algorithm for the *Safe Deposit Boxes (SDB) problem*:

Name: Safe Deposit Boxes (SDB) problem
Instance: $a_{ji} \geq 0$ for $i = 1, \dots n$ and $j = 1, \dots m$.
Output: Minimize $\sum_{i=1}^{n} x_i$ such that $\sum a_{ji} x_i \geq A_j$, $j = 1, \dots m$; $x \in \{0, 1\}$, $i = 1, \dots, n$.

A complete proof is given as follows. Let v be a MWVG with n players and m constituent WVGs $[q^t; w_1^t, \dots, w_n^t]$ for $1 \leq t \leq m$. We assume that $m < n$. The problem of computing LENGTH(v) is an integer program. An LP-relaxation changes it into a problem where we want to minimize $\sum_{i=1}^{n} x_i$ where $\sum_{i=1}^{n} x_i w_i^t \geq q^t$ for all $1 \leq t \leq m$ and $0 \leq x_i \leq 1$ for all $i \in N$. The inequalities of the linear program can be changed into equalities by introducing $n+m$ slack variable where one slack variable is used for each inequality.

$$
\begin{aligned}
\min\ & \textstyle\sum_{i=1}^{n} x_i \\
\text{s.t.}\ & \textstyle\sum_{i=1}^{n} x_i w_i^t + s_i = q^t && \text{for } \text{i=1,}\ldots,\text{ m,} \\
& x_i + s_{n+i} = 1 && \text{for } \text{i=1,}\ldots,\text{ n,} \\
& x_i \geq 0 && \text{for } \text{i=1,}\ldots,\text{ n} \\
& s_i \geq 0 && \text{for } \text{i=1,}\ldots,\text{ m+n}
\end{aligned}
\tag{10.1}
$$

The resultant LP has a total of $n + m$ constraints and $2n + m$ variables (all of which are non-negative) where $n + m$ is the number of slack variables. Any extreme point in the feasible region of formulation requires n binding constraints. It follows that any basic feasible solution contains at least n zero values. Out of these n zero values, a maximum of m values can be attributed to slack variables related to the quota constraints. Out of the remaining $n-m$ zero values, either one of the original variables is zero, or a slack variable related to the inequality $x_i \leq 1$ is zero. In either of the cases, the original variable is non-fractional. Therefore, there are at most m original variables(x_is) which may have fractional values in the LP solution. The LP solution is of course solvable in polynomial time [44].

If none of the x_is are fractional, then the LP solution is also the length of the MWVG. If not, then let l be the number of x_is equal to one in the LP solution. Then, the length of the MWVG is at least $l + 1$. If we round up every fractional values of the LP solution, then all the constraints are still satisfied. Moreover, the maximum value of the sum of the ceilings of values of the LP solution is $l + m$. Therefore the maximum error between the length of the MWVG and the sum of the ceilings of values of the LP solution is $m - 1$. □

Although computing the length of an MWVG has a PTAS, there is no FP-TAS (fully polynomial time approximation scheme) [96]. The argument is that if there is an ϵ-approximation algorithm polynomial in n and $1/\epsilon$ to approximate the length of a MWVG, then this implies that there is a polynomial algorithm to compute the length of a MWVG.

10.3 Complexity of duality questions

Now some key problems on duality of simple games are defined.

Name: X-DUALIZATION
Instance: Simple game v in representation X.
Output: Dual game v^d in representation X

Name: DUAL
Instance: Simple games v and v'
Question: Are v and v' dual of each other?

Name: SELF-DUAL
Instance: Simple games v
Question: Is v equivalent to v^d?

It is noticed that in the dualization of a simple game $v = (N, W)$, the output $(N, W(v^d))$ may be exponential in terms of the input. For example take the unanimity simple game v in which only the grand coalition is winning. Since every coalition apart from the empty set is a blocking coalition, the dual of v has $2^n - 1$ winning coalition. However, it is easy to decide DUAL for two simple games represented by their winning coalitions:

Proposition 10.9. *DUAL for two simple games $v = (N, W)$ and $v' = (N, W')$ is in P.*

Proof. If $v^d = v'$, then $|W| = 2^n - |W'|$. This is because for each losing coalition in v, its complement is winning in v'. In case $|W| = 2^n - |W'|$, then we consider the bigger of the two sets $|W|$ and $|W'|$. Without loss of generality, let us assume that $|W| \geq |W'|$. Then for each coalition $S \notin W$, we check if $N \setminus S$ is a member of W' or not. If $N \setminus S$ is not a member of W', then return 'no'. The total number of such operations involved is $(2^n - |W|)(|W'|) = (|W'|^2)$. $\square$

It follows from the proposition that deciding the problem SELF-DUAL for a simple game (N, W) is in P.

Proposition 10.10. *WVG-DUALIZATION is in P.*

Proof. We denote by $\langle q; w_1, \ldots, w_n\rangle$ a WVG where a coalition $S \subset N$ is winning if and only if $w(S) > q$. For a WVG $v = [q; w_1, \ldots, w_n]$ it is easy to see that $v^d = \langle w(N) - q; w_1, \ldots, w_n\rangle$. This observation has been made as early as in [63]. The argument is that $v^d(S) = 1$ if and only if $v(N \setminus S) = 0$. Take any coalition S such that $v^d(S) = 1$. This means that $w(S) > w(N) - q$ which is equivalent to $w(N \setminus S) = w(N) - w(S) < q$. If WVG v is represented by integers only, then $v^d = [w(N) - q + 1; w_1, \ldots, w_n]$. □

Corollary 10.11. *For a WVG v, WIDTH(v) is in P.*

Proof. For any simple game v, WIDTH(v) is equivalent to LENGTH(v^d). It is already known that WVG-DUALIZATION is in P and that for a WVG v, LENGTH(v) is in P. Therefore, computing the width of a WVG is in P. □

Now, we examine the problem MWVG-DUALIZATION. Let v be a MWVG where its constituent WVGs are $v_t = [q^t; w_1^t, \ldots, w_n^t]$ for $1 \leq t \leq m$. Also, $v^d = \bigvee_{t=1}^m {v_t}^d$. We know that ${v_t}^d = [(\sum_i^n w_i^t) - q^t + 1; w_1^t, \ldots, w_n^t]$. Therefore while dualizing a MWVG, it easy to get a disjunction of WVGs but not easy to get a MWVG. A dual of MWVG is also a simple game and therefore can be represented by a MWVG. However, it will be interesting to check if the dual of a MWVG with dimension m can be represented by a MWVG with a dimension polynomial in m.

It is interesting to see that although computing the length of a MWVG is difficult, the computing the width of a MWVG is computationally easy.

Proposition 10.12. *For a MWVG v, WIDTH(v) is in P.*

Proof. Let v be a MWVG where its constituent WVGs are $v_t = [q^t; w_1^t, \ldots, w_n^t]$ for $1 \leq t \leq m$. Then, $v^d = \bigvee_{t=1}^m {v_t}^d$. We know that ${v_t}^d = [(\sum_i^n w_i^t) - q^t + 1; w_1^t, \ldots, w_n^t]$. Now we know that WIDTH(v) is equal to LENGTH(v^d).

It is proved that $\text{LENGTH}(v^d) = \texttt{Inf}_{t=1}^{n}\{\text{LENGTH}({v_t}^d)\}$. Choose $i \in \{1, \ldots m\}$, such that length of ${v_i}^d$ is the smallest among ${v_t}^d$s for all t. Then assume for contradiction that there exists some coalition $C \subseteq N$ such that C is winning in v_d and $|C|$ is less than the length of ${v_i}^d$. Then then there exists a $j \in \{1, \ldots m\}$ other than i such that ${v_j}^d(C) = 1$. This implies that the length of ${v_j}^d$ is less than the length of ${v_i}^d$ which is a contradiction. □

There is no known polynomial time algorithm for W^m-DUALIZATION. Neither are there any hardness results for the problem. However it is known that W^m-DUALIZATION is polynomially time equivalent to the problems DUAL and SELF-DUAL for simple games represented by their MWCs [37, 86]. It is also known that linear simple games represented by MWCs can be dualized in polynomial time [66]. Now we prove that the complexity of DUAL for WVGs is NP-hard:

Proposition 10.13. *DUAL for WVGs is co-NP-complete.*

Proof. We first define another problem EQUIVALENT-WVGs:

Name: EQUIVALENT-WVGs
Instance: WVGs $v = [q; w_1, \ldots, w_n]$ and $v' = [q'; {w_1}', \ldots, {w_n}']$
Question: Is $v = v'$?

We first prove that this problem is NP-hard. We provide a reduction from the problem of checking whether a player is a dummy or not. Let v by a WVG $[10q+1; 10w_1, \ldots, 10w_{n-1}, 1]$ where q and w_is are all integers. It is known that it is NP-hard to verify whether player n with weight 1 is a dummy or not [151]. This is equivalent to asking whether v is equivalent to WVG $[10q; 10w_1, \ldots, 10w_{n-1}, 0]$. Moreover, it is easy to see that EQUIVALENT-WVGs is in co-NP because any coalition S such that $v(S) \neq v'(S)$ is a ‘no’ certificate.

Now that we know that EQUIVALENT-WVGs is NP-hard we show that any instance of EQUIVALENT-WVGs is equivalent to the problem DUAL for WVGs. We take an instance of EQUIVALENT-WVGs where we want to check whether WVGs $v = [q; w_1, \ldots, w_n]$ and $v' = [q'; w_1', \ldots, w_n']$ are equivalent. This is the same as asking whether the dual of v' is $[w(N) - q + 1; w_1, \ldots, w_n]$. □

From the proof it follows that SELF-DUAL for WVGs is also co-NP-complete. Moreover, DUAL and SELF-DUAL for MWVGs are co-NP-complete. For a pair of simple games represented by their winning coalitions, both games have to be exactly similar for them to be equivalent. The same rule holds for simple games represented by their minimal winning coalitions.

10.4 Bribery in WVGs

Manipulation, control and bribery in elections and social choice protocols have been examined both in political science and multiagent systems. In [75], a comprehensive analysis of manipulation in elections was undertaken. Different kinds of manipulations considered are insincere behaviour by voters, bribery of voters and control by mechanism designers. Manipulations in voting systems have received interest in many recent papers, for instance [104, 180, 50]. In this section, a similar approach is used to analyse bribery in simple games.

Winning coalitions with the smallest length are the following set: $W^e = \{S : S \subset W, |S| \leq |S'|\ \forall S' \in W\}$. For the specific case of WVGs, another variation is winning coalitions with the minimum weight: $W^s = \{S : S \subset W, w(S) \leq w(S')\ \forall S' \in W\}$. It is evident that W^e and W^s are subsets of W^m. These kinds of minimal winning coalitions have also been considered in [51]. Both W^s and W^e appear to be useful concepts especially from the point of view of bribery and manipulation in social choice protocols. If a manipulator wants to control or manipulate a coalition, he would prefer a coalition which is winning but barely, with minimal weight or number of players. Minimizing the number of players bribed has the motivation of maximizing confidentiality and minimizing costs. Minimizing the weight of the bribed coalition has motivation in the assumption that players may price themselves according to their perceived importance, which is their contributed weight. For example, if each player has unit cost of being influenced, the cost of bribery is the length of the game. The general problem is defined as follows:

Name: SVG-WIN-BRIBERY
Instance: Simple game v with cost c_i for each player $i \in N$.
Output: Minimize the bribery cost while ensuring a win.

10.4.1 Under no information

We now examine the complexity of optimal bribery to ensure the decision when there is no information on the players' preferences (whether they want to vote 'yes' or 'no').

Influencing a decision

Proposition 10.14. *SVG-WIN-BRIBERY is in P for a simple game represented by a WVG,* (N, W) *or* (N, W^m) *where each player has unit cost and the manipulator has no knowledge of player preferences.*

Proof. This follows directly from Observation 10.6) that computing a winning coalition with the minimum number of players can be computed in polynomial time for all the three representations. □

Proposition 10.15. *SVG-WIN-BRIBERY is NP-hard for a MWVG where each player has unit cost and the manipulator has no knowledge of player preferences.*

Proof. This follows from the result that computing LENGTH(v) of MWVGs is NP-hard. □

Since we noticed that LENGTH(v) for MWVG has an absolute error approximation algorithm, this means that SVG-WIN-BRIBERY also be approximated efficiently. As the approximation algorithm always overestimates the length of the MWVG, it is guaranteed that a big enough coalition of players has been bribed.

Proposition 10.16. *SVG-WIN-BRIBERY is NP-hard for WVGs where each player has cost proportional to its weight and the manipulator has no knowledge of player preferences.*

Proof. We use a reduction from the optimization version of the SUBSET-SUM Problem where the instance is $A = \{a_1, \ldots, a_k\}$ and we want to minimize $\sum_{i=1}^{k} x_i a_i$ such that $\sum_{i=1}^{k} x_i a_i \geq Q$ where $x_i \in \{0, 1\}$. This problem is equivalent to SVG-WIN-BRIBERY which involves minimizing $\sum_{i=1}^{k} x_i(Cw_i)$ such that $\sum_{i=1}^{k} x_i w_i \geq Q$ where $x_i \in \{0, 1\}$, $w_i = a_i$ for all $i = 1$ to k and C is a constant. □

Maintaining the status quo

If a manipulator wants to maintain the status quo and prevent the formation of a winning coalition, he might want to control a blocking coalition with the least cost. This could again be a coalition which is winning in the dual of the game but either has the smallest number of players or the least amount of weight.

Name: SVG-VETO-BRIBERY
Instance: Simple game v with cost c_i for each player $i \in N$.
Output: Ensure a no-win (so that a decision cannot be taken) while minimizing cost.

It is easy to see that SVG-VETO-BRIBERY for WVG v is equivalent to SVG-WIN-BRIBERY for v^d. Therefore, we automatically arrive at the following propositions:

Proposition 10.17. *If the manipulator has no knowledge of player preferences,*

1. *SVG-VETO-BRIBERY is in P for a WVG where each player has unit cost.*
2. *SVG-VETO-BRIBERY is in P for a MWVG where each player has unit cost.*
3. *SVG-VETO-BRIBERY is NP-hard for a WVG where each player has cost proportional to its weight.*

For a simple game $v = (N, W^m)$, it was shown that SVG-WIN-BRIBERY, is in P. However, it is not clear whether, SVG-VETO-BRIBERY is in P or not since the complexity of dualization of v is an open question.

10.4.2 Manipulation under full or partial information

Under full or partial information, we get the same results on complexity of bribery as in the last subsection. Let us say that we have a simple game in which we know that a coalition $S \subset N$ of players wants to win. If the coalition S can win on its own, then there is no motivation for bribery. If the coalition S is not enough to effect a win, then the problem of bribing to implement an overall win transforms into a smaller optimization problem of bribing enough players among $N \setminus S$ to effect a win. A similar argument holds for the situation where we want to effect a veto and we have partial or full information of a coalition of players who also want to veto the decision.

10.5 Conclusion

Table 10.1. Complexity of dualization, length and bribery

Input	(N, W)	(N, W^m)	WVG	MWVG
DUALIZATION	Exp	?	P	?
DUAL	P	?	co-NPC	co-NPC
SELF-DUAL	P	?	co-NPC	co-NPC
LENGTH	P	P	P	NP-hard
WIDTH	?	?	P	P
EQUIVALENT	P	P	co-NPC	co-NPC
SVG-WIN-BRIBERY with uniform costs	P	P	P	NP-hard
SVG-WIN-BRIBERY with weight proportional costs	N/A	N/A	NP-hard	N/A
SVG-VETO-BRIBERY with uniform costs	?	?	P	P
SVG-VETO-BRIBERY with weight proportional costs	N/A	N/A	NP-hard	N/A

In the chapter, the complexity of key questions concerning length, width, and duality of simple games is examined. Moreover, the complexity of identifying the ideal coalition to bribe has been considered for different cost patterns. A summary of results is presented in Table 10.1. The question marks indicate problems which are open. It is seen that since simple games involve binary decisions, the bribery process is not as complex as general elections with a range of alternatives. However, optimal bribery has different complexity for different representations of games and approaches to bribery. The idea of optimal bribery may be a realistic consideration if there are multiple decisions to be made via simple games, and the briber does not want to overuse his resources. In analyzing bribery, it will be more realistic to consider probabilistic models where there are probabilities for players to vote 'yes' or 'no'. In this context, results from information theory might shed more light. Another direction is to examine simple games where instead of deciding a binary outcome, the players vote for a list of candidates. Moreover, characterizing the complexity of dualization of simple games represented by MWCs is a long-standing open question.

Part IV

Resource Allocation & Networks

11

Cooperative game theory & simple games

The value or worth of a man is, as of all other things, his price; that is to say, so much as would be given for the use of his power, and therefore is not absolute, but a thing dependent on the need and judgement of another.
- Thomas Hobbes (Leviathan)

In seeking private interests, we fail to secure greater collective interests. The narrow rationality of self-interest that can benefit us all in market exchange can also prevent us from succeeding in collective endeavors.
- Russell Hardin (Collective Actions)

The classes of problems which are respectively known and not known to have good algorithms are of great theoretical interest ... I conjecture that there is no good algorithm for the traveling salesman problem. My reasons are the same as for any mathematical conjecture: (1) it is a legitimate mathematical possibility; and (2) I do not know.
- Edmonds (1966)

Abstract Simple coalitional games are not only a type of voting games but also a fundamental class of cooperative games. In this chapter, cooperative games and cooperative game solutions are introduced. Cooperative game theory is concerned with analyzing which coalitions will form and how should the coalitions divide the payoff between their members. The trend of using computational tractability

as a criterion for cooperative game solutions is recent and is prevalent in the mathematics of operations research and theoretical computer science. In this chapter, the computational aspects of various cooperative game solutions in simple games are examined. Questions considered include the following: 1) for solution set X and simple game v, is X of v empty or not, 2) compute an element in X of v and 3) verify if a payoff is in X of v. Some representations taken into account are simple games represented by W, W^m, WVGs and MWVGs. The cooperative solutions considered are the core, ϵ-core, least-core, nucleolus, prekernel, kernel, bargaining set and stable sets. A complexity of checking the stability of the core of simple games is also examined. Structural results of the least core and nucleolus payoffs of simple games are presented. A theorem from the paper "The nucleolus and kernel for simple games or special valid inequalities for $0-1$ linear integer programs" by L.A Wolsey is corrected. It is proved that an oracle to compute a least core payoff for a simple game in any *passer-reasonable* representation can be used to compute the worst excess of a least core payoff. Finally, the relation between *cost of stability* and the least core is examined. A natural and desirable solution called the super-nucleolus is also proposed.

11.1 Introduction and background

Cooperative game theory models problems where a group of players cooperate to make a profit or investment and the profit or cost has to be allocated among the players in a fair and stable way. If a subset of players in N cooperate and work together, they form a coalition. Cooperative game theory is used to analyse which coalitions will form and how the coalitions should divide the payoff among their members. Solution concepts in cooperative theory measure profit allocations of players while considering the profit of each coalition of players. The foundations of cooperative game theory were laid by von Neumann and Morgenstern. The biggest focus in the development of cooperative game theory has been devising various solution concepts to explain equilibrium in different systems. This also involved axiomatic characterization of the properties of the solution concepts and their relations with each other.

11.2 Related work

Cooperative game theory has seen tremendous growth in the last few decades with several textbooks written on it [196, 163, 62, 176]. Concepts from the area have then been used in various combinatorial optimization games in operations research which involve resource allocation among multiple players [54, 38]. Although algorithms to compute different solutions have been considered in the mathematics of operations research literature, Deng and Papadimitriou [60] undertook one of the earliest computational complexity investigation of different solution concepts. There have been new developments in the computational complexity of solution concepts of weighted voting games [68, 70]. Deng and Fang have surveyed the developments in algorithmic cooperative game theory in a recent detailed article [58]. Cooperative game theory has also been widely used by the artificial intelligence and multiagent community, especially for multiagent resource allocation [46].

11.3 Preliminaries

We define different kinds of cooperative games. The definitions are to provide context and to show how simple games are related to other classes of games. The relations between the classes of cooperative games are further highlighted in Figure 11.1 and Figure 11.2.

Definitions 11.1. *A cooperative game is:*

- zero-normalized *if* $v(\{i\}) = 0$ *for all* $i \in N$,
- monotonic *if* $S \subseteq T \subseteq N$ *implies that* $v(S) \leq v(T)$,
- superadditive *if for all* $S, T \subset N$, *if* $S \cap T = \emptyset$, *then* $v(S \cup T) \geq v(S) + v(T)$,
- cohesive *if* $v(N) \geq \sum_{k=1}^{K} v(S_k)$ *for every partition* $\{S_1, \ldots, S_k\}$ *of* N,
- additive *if for all* $S, T \subset N$, *if* $S \cap T = \emptyset$, *then* $v(S \cup T) = v(S) + v(T)$,
- constant-sum *if for all* $S \subset N$, $v(S) + v(N \setminus S) = v(N)$,
- convex *if* $v(S \cup T) \geq v(S) + v(T) - v(S \cap T)$ *for all* $S, T \subset N$.

Among cooperative games, simple coalitional games are a fundamental subclass of games because any cooperative game can be mapped into a corresponding simple game by introducing a threshold which is a value between zero and the value of the grand coalition. In the simple game variation, a coalition which has a value greater than or equal to the threshold may be considered winning. Well-studied games such as threshold network flow games [26] fit into this framework.

Definition 11.2. *For each cooperative game* (N, v) *and each threshold* $t \in \mathbb{R}^+$*, the* corresponding threshold game or version *is defined as the cooperative game* (N, v^t)*, where*

$$v^t(S) = \begin{cases} 1 & \textit{if } v(S) \geq k, \\ 0 & \textit{otherwise.} \end{cases}$$

It is easily verified that, for any threshold t, if a game (N, v) is monotone, so is its threshold version (N, v^t), in which case (N, v^t) is a simple game.

A weighted voting game is *homogeneous* if it can have a *homogeneous representation*. A WVG is a *homogeneous representation* if all minimal winning coalitions have the same weight. This implies that in a homogeneous representation, players with equal power get the same weight.

Observation 11.3 *We recall that a decisive simple game is one which is both proper and strong (Definition 2.5). Therefore, it easy to see that a constant-sum game which is simple is equivalent to a decisive simple game.*

A *solution concept* assigns for each game a set of payoffs or allocations. The Banzhaf index can also be considered as a solution concept specifically in the context of simple games. We will generally denote a payoff by a vector $x = (x_1, \ldots, x_n)$ where x_i is the payoff of player i for $i \in N$. The choice of a specific solution concept depends on the notion of fairness, stability and certain desirable properties. We write $x(S)$ for $\sum_{i \in S} x_i$

Definitions 11.4. *Some desirable properties of payoffs for solution concepts are:*

- Efficiency*: For any game* (N, v)*,* $\sum_{i \in N} x_i = v(N)$.

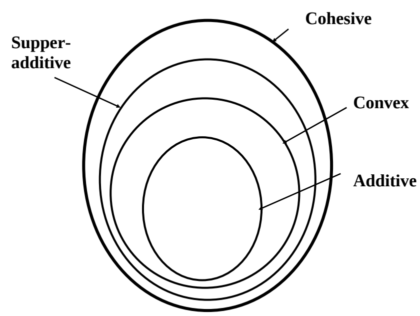

Fig. 11.1. Relations of cooperative games I

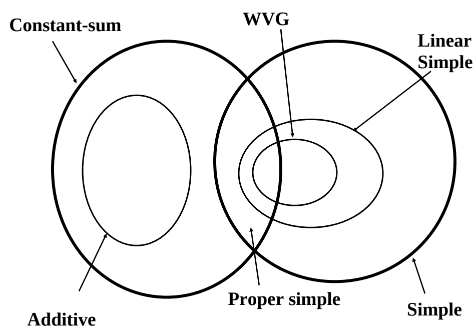

Fig. 11.2. Relations of cooperative games II

- Individual rationality*: $x_i \geq v(i)$ for all $i \in N$. This means the payoff of a player is at least the amount which it can get by acting alone in the game.*
- Coalitional rationality*: $x(S) \geq v(S)$ for all $S \subseteq N$.*
- Nonemptiness*: The set of payoffs according to the solution concept is non-empty for any game (N, v).*
- Computationally feasible*: The solution concept can be computed efficiently.*
- Symmetry*: The solution concept is not influenced by renumbering of the player set. The condition is also referred to as anonymity in the literature.*

- Additive: *For any two v_1 and v_2, $x_i(v_1 + v_2) = x_i(v_1) + x_i(v_2)$ where game $(N, v_1 + v_2)$ is such that $(v_1 + v_2)(S) = v_1(S) + v_2(S)$ for any coalition S.*
- Zero allocation to Dummies: *If i is a dummy player, then $x_i(v) = v(\{i\}) = 0$.*
- Pareto optimal: *If x, y are two payoffs such that $x_i > y_i$ for all $i \in N$, then y is not in the solution.*

We define different types of payoffs depending on which desirable properties they satisfy:

Definitions 11.5. *Different kind of payoffs in cooperative games are:*

- Feasible payoff: *For a cooperative game (N, v), a* feasible payoff *is $x \in R^n$ such that $x(N) \leq v(N)$. The set of feasible payoffs is denoted by $X^*(N, v)$.*
- Preimputation: *For a cooperative game (N, v), a* preimputation *is $x \in R^n$ such that $x(N) = v(N)$. Preimputations are* efficient *feasible payoffs. The set of preimputations for a game v is denoted by $I^*(v)$.*
- Imputation: *For a cooperative game (N, v), an* imputation *x is a preimputation which satisfies individual rationality, i.e. such that for all $i \in N$, $x(i) \geq v(i)$. The set of imputations for a game v is denoted by $I(v)$.*

11.4 Cooperative game theory solutions

11.4.1 Introduction

In cooperative game theory, the goal is to distribute the payoffs fairly among the players and encourage cooperation. Solution concepts formalize the notions of fair and stable payoffs. For a payoff $x = (x_1, ..., x_n)$, the *excess* $e(x, S)$ of a coalition S under x is $x(S) - v(S)$. The *excess vector* of a payoff x, is the vector $(e(x, S_1), ..., e(x, S_{2^n}))$ where $e(x, S_1) \leq e(x, S_2) \leq e(x, S_{2^n})$. We denote the distinct values in the excess vector by $-\epsilon_1(x, v), -\epsilon_2(x, v), \ldots, -\epsilon_m(x, v)$ where $-\epsilon_i(x, v) < -\epsilon_j(x, v)$ for $i < j$. For a payoff x and game v, the set of coalitions that get the i-th distinct worst excess $-\epsilon_i(x, v)$ will be denoted by $A^i_x(v)$ and a member of $A^i_x(v)$ will be called an ϵ_i-coalition. Many of the following definitions are from [68] and [163]. A payoff x is called an *S-feasible payoff vector* if $x(S) = v(S)$.

An N-feasible payoff vector which is a preimputation is called a *feasible payoff profile*.

We consider the following cooperative game solutions for simple game: Shapley value, core, ϵ-core, nucleolus [149], stable set [141], bargaining set [149], kernel [149], prekernel and τ-value. The core of a game is one of the most fundamental solutions in cooperative game theory. The idea of the core goes back to von Neumann and Morgenstern [213]. The modern definition and name were first used in [98].

Definition 11.6. ***Core****: A payoff x is in the* core *if and only $\forall S \subset N$, $e(x, S) \geq 0$ or in other words $x(S) \geq v(S)$. The core of a game (N, v) is denoted by $C(v)$ and $C(v) \subset I(v)$.*

A core imputation guarantees that each coalition gets at least what it could gain on its own. The core is not unique and is a set which satisfies a system of weak linear inequalities, so it is closed and convex. Moreover, the core is well-defined, but can be empty. Those games which have non-empty cores are called *balanced*. Although the core is a desirable solution concept, it may be empty for many games. This led to the development of *ϵ-core* [193] and *least core* [142].

Definitions 11.7. *A preimputation x is in the* ϵ-core *if $\forall S \subset N$, $e(x, S) \geq -\epsilon$. The ϵ-core is denoted by $C_\epsilon(v)$. The preimputation x is in the* least core *if it is in the ϵ-core for the smallest possible ϵ. We will denote by $-\epsilon_1(v)$, the worst excess of any least core payoff of (N, v).*

Therefore, the least core is the intersection of all the ϵ-cores. The least core is not unique and may contain many payoffs. One may want to find fairest payoffs among the payoffs within the least core. This led to the idea of the *prenucleolus* and *nucleolus* [189].

Definition 11.8. ***Prenucleolus****: A preimputation x that has lexicographically the largest excess vector is called the* prenucleolus.

Definition 11.9. ***Nucleolus****: An imputation x that has lexicographically the largest excess vector is called the* nucleolus.

The prenucleolus always exists and is unique as long as $v(S) = 0$ for all one person coalitions [189]. It is easy to see that the least core is always non-empty and always contains the prenucleolus. For any cooperative game v for which $I(v)$ is non-empty, the nucleolus also exists and coincides with the prenucleolus. In terms of the computational complexity of problems concerning nucleolus and prenucleolus, there is no difference.

Definition 11.10. ***Stable Set****: An imputation x* dominates *imputation y if there is a non-empty coalition S such that $x_i > y_i$ for all $i \in S$ and $x(S) \leq v(S)$. Two imputations can dominate each other. A* stable set *of a game (also known as the von Neumann-Morgenstern solution [213]) is a set of imputations which satisfies the following two properties:*

1. Internal stability: *No imputation in the stable set is dominated by another imputation in the set.*
2. External stability: *All imputation outside the set are dominated by at least one alternative in the set.*

A stable set may or may not exist [140]. Moreover, just like the core, even if it exists it is not necessarily a singleton. [139]. For zero-normalized simple games, stable sets always exist. It is also known that the core is a subset of any stable set and if the core is stable it is the unique stable set [62].

The bargaining set models stability of payoffs where if player i has an objection against player j to imputation x, then j has a counter-objection.

Definition 11.11. ***Bargaining set****: The pair (y, S), where S is a coalition, is an* objection of i against j to x *if S includes i but not j, $y(S) = v(S)$ and $y_k > x_k$ for all $k \in S$. A pair (z, T) where T is a coalition is a* counter-objection to the objection (y, S) of i against j *if T includes j, but not i, $z(T) = v(T)$, $z_k \geq x_k$ for all $k \in T \setminus S$ and $z_k \geq y_k$ for all $k \in T \bigcap S$. An imputation x belongs to the* bargaining set $M(v)$ *of game v, if for any objection (y, S) of any player i against player j to x, there is a counter-objection to (y, S) by j to x.*

The kernel is a subset of the bargaining set with similar concept of objections and counter-objections.

Definition 11.12. ***Kernel****: A coalition, S is an* kernel-objection of i against j to x *if S includes i but not j and $x_j > v(\{j\})$. A coalition, T is the* kernel-counter-objection to the objection S *of i against j if T includes j but not i and $e(x, T) \leq e(x, S)$. The* kernel *of a coalitional game with transferable payoffs is the set of all imputations x with the property that for every objection S of any player i against any other player j to x there is a counter-objection of j to S. The kernel of a simple game (N, v) is denoted by $K(v)$. There is also an alternative way to define the kernel. We let $s^v_{ij}(x)$ be the* maximum surplus *of player i over player j with respect to x, i.e.,*

$$s^v_{ij}(x) = \max\left\{v(S) - x(S) | S \subseteq N \setminus \{j\}, i \in S\right\}.$$

Then the kernel is the set of imputations x such that

$$(s^v_{ij}(x) - s^v_{ji}(x))(x_j - v(\{j\})) \leq 0$$

and

$$(s^v_{ji}(x) - s^v_{ij}(x))(x_i - v(\{i\})) \leq 0.$$

We say that i outweighs *j according to payoff x if $s^v_{ij}(x) > s^v_{ji}(x)$ and $x_j > v(\{j\})$. This is equivalent to saying that i has an objection against x to j for which j has no counter-objection.*

Definition 11.13. ***Prekernel****: The* prekernel *of a game (N, v) is the set of preimputations $x \in I^*(v)$ such that:*

$$s^v_{ij}(x) = s^v_{ji}(x)$$

for all i, j. The prekernel of the game (N, v) is denoted by $PK(v)$.

We see that simple games with no passers are both zero-normalized and also zero-monotonic.

In the last five decades, properties of cooperative game solutions have been widely studied. We state some of the well known and relevant facts of many of these solutions. Since the nucleolus always exists and nucleolus $\subset$ kernel $\subset$ bargaining set, therefore, the kernel and bargaining set are always non-empty. Figure 11.3 provides the relations with some of the cooperative game solutions.

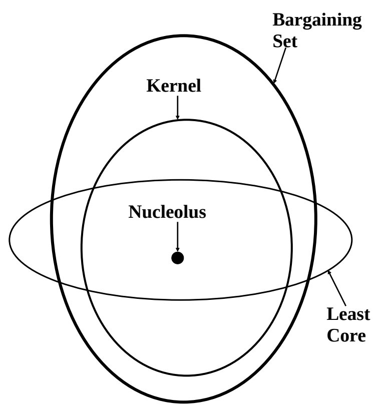

Fig. 11.3. Relations of cooperative game solutions

Every constant-sum game which is not additive has an empty core. For a convex game, the core is non-empty and the Shapley value is in the core.

The prenucleolus, prekernel and the least core exist for any cooperative game and the nucleolus, bargaining set and kernel exist for zero-normalized games. On the other hand, the core, stable set and ϵ-core may be empty for a game. Shapley values, prenucleolus and the nucleolus (if it exists) are unique. Moreover, if the core is non-empty, the nucleolus is in the core. For simple games, bargaining sets are equivalent to the core if the core is non-empty, but it may be the case that the core is empty but the bargaining set is non-empty [65].

11.4.2 Desirability relation and cooperative game solutions

For simple games, we consider respecting the desirability relation in a simple game as another useful criterion for a solution concept. A payoff x obeys desirability relations if when i is more desirable than j, then $x_i \geq x_j$. Table 11.1 lists cooperative game solutions and whether they respect the desirability relation.

Table 11.1. Cooperative game solutions and desirability relation in simple games

Cooperative game solution	Desirability relation
Banzhaf index	✓
Shapley-Shubik index	✓
Holler index	✓
Deegan-Packel Index	×
Nucleolus	✓
Prekernel	✓
Kernel	✓
Bargaining Set	×
Core	×
Least core	×

11.4.3 Generalized problems

We define some natural problems in cooperative game theory. For the sake of consistency and continuity, we have used the same names for the problems as in [68] where single WVG are analysed.

Name: EMPTY-X
Instance: Simple game v
Question: Is X empty?

Name: IN-X
Instance: Simple game v and payoff p
Question: Is p in solution X of v?

Name: CONSTRUCT-X
Instance: Simple game v
Output: A payoff p which is in solution X of v?

Name: ISZERO-X
Instance: Simple game v and player i
Question: Is payoff of player i in game v zero according to solution X?

For a solution X which is unique, the problem IN-X is equivalent to checking if a payoff is X and the problem CONSTRUCT-X is equivalent to computing X. If a solution X is not unique, then the problem ISZERO-X is not precise and we will ignore it. Moreover if X is unique, then we will consider CONSTRUCT-X simply as computing X.

11.5 Core

Proposition 11.14. *For a simple game (N, v), $I(v)$ is non-empty if and only if there is at most one passer.*

Proof. $I(v)$ is non-empty if and only if there exists a payoff x such that $x(N) = 1$ and $x_i \geq v(\{i\})$ for all $i \in N$. This is true if and only if either $v(\{i\}) = 0$ for all $i \in N$, or there exists a unique $j \in N$ such that $v(\{j\}) = 1$. If j is the unique passer, then x is an imputation if and only if $x_j = 1$. □

If there is a passer, then no other player can be a vetoer. Elkind et al. [68] showed that EMPTY-CORE is in P for WVGs. We notice that this observation holds true for simple games in any representation:

Proposition 11.15. *EMPTY-CORE is in P for simple games in any representation where the value of a coalition is obtained in polynomial time.*

Proof. We first observe that EMPTY-CORE for a simple game v is equivalent to verifying that IDENTIFY-VETOERS is an empty set. The argument for this is as follows. We want to prove that the core of a simple game is non-empty if and only if a vetoer exists.

($\Rightarrow$) Let us assume that the core is non-empty and there is a feasible imputation p which is in the core. Let us also assume that no vetoer exists. Take any player j with non-zero imputation according to x. Since j is not a vetoer, $v(N \setminus \{j\}) = 1$. However $p(N \setminus \{j\}) < 1$ is a contradiction since x is in the core.

($\Leftarrow$) We now prove that if a vetoer exists, the core is non-empty. We know that $v(N) = 1$ and $p(N) = 1$. If a vetoer i exists, our imputation p is such that we can give all the payoff to i, and none to the other players. In that case for all $S \subseteq N \setminus \{i\}$, $p(S) = 0$ and $v(S) = 0$. Therefore p is in the core and the core is non-empty.

Identifying vetoers is easy for any representation of a simple game for which value of each coalition can computer in polynomial time. A player i is a vetoer if and only if $v(N \setminus \{i\}) = 0$. A simple game even with an implicit characteristic function can return $v(N \setminus \{i\})$. Therefore IDENTIFY-VETOERS can be used in a black-box manner to solve EMPTY-CORE. □

Elkind et al. [68] prove that if the core of simple game v is non-empty, the nucleolus of v is given by $x_i = 1/k$ if i is vetoer and $x_i = 0$ if i is not a vetoer, where k is the total number of vetoers. The result also holds even if we consider any element of the kernel and any simple game [9]. Therefore, for a simple game v represented by (N, W), (N, W^m), WVG, MWVG or any other representation, if the core is non-empty, the nucleolus can be computed in polynomial time.

11.6 Core stability

It is already known that the core is a subset of every stable set, no stable set is a subset of another stable set and if the core is a stable set then it is the only stable set (Prop 279.2a [163]). The following is a characterization of the stable set of a simple game:

Proposition 11.16. *For a simple game v, a set of imputations is a stable set if and only if for some minimal winning coalition T, all players not in T get zero payoff.*

Proof. ($\Leftarrow$)

If T be a minimal winning coalition. Then the claim is that $X_T = \{x|x(T) = 1\}$ is a stable set. The set is internally stable because there is no imputation $y \in X_T$ such that there exists a non-empty coalition S such that $x_i < y_i$ for $i \in S$. Moreover, let z be an imputation not in X_T. Then $z(T) < x(T)$, which implies there exists an $x \in X_T$ which dominates z. Therefore, constructing a stable set of a simple game simply requires identifying a minimal winning coalition.

($\Rightarrow$) Let us assume that Y is a stable set of imputations of game v such that there exists at least one imputation $y \in Y$ which does not distribute all its payoff exactly to players in a minimal winning coalition. This means that that y distributes its payoff to a coalition T'' such that $T'' \supset T$ where T is a minimal winning coalition. However, it is easy to see that y is dominated by an imputation in X_T. □

Corollary 11.17. *CONSTRUCT-STABLE-SET and IN-STABLE-SET are in P for any representation of a simple game where the value of a coalition is obtained in polynomial time.*

The stability of the core is an important question in algorithmic cooperative game theory. Core stability has been examined for several classes of games including assignment games [183], minimum coloring games [35], network flow games [200] and vertex cover games [79]. However for simple games, a stable set always exists. The following is a characterization of the core stability of simple games.

Proposition 11.18. *The core of a simple game is stable if and only if there is only one minimal winning coalition of the game.*

Proof. If coalition S is the only MWC of the simple game v, then all players in S are vetoers. Any imputation which distributes the payoff among players in S in the core. Moreover, from Proposition 11.16, we see that that the core is also a stable set.

There is an alternative way to see this. A cooperative game is *convex* if $v(S \cup T) \geq v(S) + v(T) - v(S \cap T)$ for all $S, T \in 2^N$ where $S \neq T$. Moreover, there is a well known result due to Shapley [191] that for convex games, the core is stable. We see that a simple game is not convex if and only if there exist coalitions S and T, $S \neq T$ such that $v(S) = 1$, $v(T) = 1$ and $v(S \cap T) = 0$. Then it follows that $v(S \cup T) = 1$. If S and T are mutually exclusive then this means that there are two MWCs. If S and T are not mutually exclusive then this means that $S \cap T$ does not contain a MWC but S and T both contain MWCs. □

Corollary 11.19. *CORE-STABILITY is in P for any representation of a simple game where the value of a coalition is obtained in polynomial time.*

Proof. The problem is to check whether the simple game has only one MWC or not. This is trivial for (N, W^m). For other representations, one needs to identify the vetoers and then verify if the coalition of vetoers forms a winning coalition. This can be checked in polynomial time. □

11.7 Least core

Elkind et al.(Theorem 5, [68]) proved that the problems EMPTY-ϵ-CORE, IN-LEASTCORE and CONSTRUCT-LEASTCORE are NP-hard for WVGs. Since EMPTY-ϵ-CORE is NP-hard, then it follows that CONSTRUCT-ϵ-CORE is NP-hard. Also, from the proof in Theorem 5, [68], it is easy to see that IN-ϵ-CORE is NP-hard.

It is easy to see from the definition of the least core, that it is the solution of the following LP:

$$\begin{aligned} &\min \epsilon \\ &\text{s.t.} \quad x(S) \geq v(S) - \epsilon \,, \text{ for all } S \subset N \,, \\ &\qquad\; x_i \geq 0 \,, \text{for all } i \in N, \\ &\qquad\; \textstyle\sum_{i=1,\ldots,n} x_i = v(N) \,. \end{aligned} \tag{11.1}$$

We now show that the for any simple games and least core payoff, every player is in one coalition which gets the worst excess. Proposition 11.20 is specially for simple games but as we shall see later, it can be generalized to monotone cooperative games.

Proposition 11.20. *For any simple coalitional game* (N, v)*, suppose that* $x = (x_1, \ldots, x_n)$ *is an element in the least core, where the minimum excess is* $-\epsilon$*. Then for any player* $i \in N$ *there exists a coalition* T *such that* $i \in T$ *and* $e(x, T) = -\epsilon$*.*

Proof. Let A be the set of players such that for every $j \in A$, we have that j is contained in some coalition M with $e(x, M) = -\epsilon$. Consider a player $i \in N \setminus A$. We must have $x_i > 0$, since if $x_i = 0$, then for any coalition S such that $i \notin S$ and $e(x, S) = -\epsilon$, the excess $e(x, S \cup \{i\}) = -\epsilon$. Let δ be half of the minimum of the non-zero differences between successive components of the excess vector of x. Then we can obtain a new imputation y such that $y_i = x_i - \mathsf{Min}(x_i, \delta)$, and $y_j = x_j + \frac{\mathsf{Min}(x_i, \delta)}{|A|}$ for $j \in A$, and $y_k = x_k$ for $k \notin A \cup \{i\}$. Since the smallest excess for y is more than $-\epsilon$, this means that x is not in the least core which is a contradiction. □

Corollary 11.21. *Suppose* $x = (x_1, \ldots, x_n)$ *is the nucleolus of a simple game. Let* $e(x, S)$ *be the first element of the excess vector of* x*. Then for any player* i *, there exists a coalition* T *such that* $i \in T$ *and* $e(x, T) = e(x, S)$*.*

Proof. This follows from the fact that the nucleolus is a member of the least core.
□

We see that Proposition 11.20 can be generalized to monotone cooperative games:

Proposition 11.22. *For any monotone cooperative game* (N, v)*, suppose that* $x = (x_1, \ldots, x_n)$ *is an element in the least core, where the minimum excess is* $-\epsilon$*. Then for any player* $i \in N$ *there exists a coalition* T *such that* $i \in T$ *and* $e(x, T) = -\epsilon$*.*

Proof. Let A be the set of players such that for every $j \in A$, we have that j is contained in some coalition M with $e(x, M) = -\epsilon$. Let P_1 be the set of those coalitions which get an excess of $-\epsilon$. Consider a player $i \in N \setminus A$.

Consider the case $x_i = 0$. Choose a coalition $S \in P_1$. Then we consider the coalition $S \cup \{i\}$. If $v(S \cup \{i\}) = v(S)$, then $i \in A$. If $v(S \cup \{i\}) > v(S)$, then $e(x, S \cup \{i\}) < -\epsilon$ which is a contradiction.

Now consider the case when $x_i > 0$. Let δ be half of the minimum of the non-zero differences between successive components of the excess vector of x. If there exists a coalition $S \in P_1$ such that $x(S \cup \{i\}) - v(S \cup \{i\}) < -\epsilon$, then this is a contradiction. If there exists a coalition $S \in P_1$ such that $x(S \cup \{i\}) - v(S \cup \{i\}) = -\epsilon$, then $i \in A$. If there exists no coalition $S \in P_1$ such that $x(S \cup \{i\}) - v(S \cup \{i\}) \leq -\epsilon$, then we can obtain a new payoff y such that $y_i = x_i - \mathsf{Min}(x_i, \delta)$, and $y_j = x_j + \frac{\mathsf{Min}(x_i,\delta)}{|A|}$ for $j \in A$, and $y_k = x_k$ for $k \notin A \cup \{i\}$. Since the smallest excess for y is more than $-\epsilon$, this means that x is not in the least core which is a contradiction. $\square$

Proposition 11.23. *Let (N, v) be a simple game with no vetoers and let $x = (x_1, \ldots, x_n)$ be a member of the least core of (N, v). Then, there is no player which is present in every coalition which gives the minimum excess for imputation x.*

Proof. Let P_1 be the set of coalitions which get the minimum excess $-\epsilon$. We already know that every player is a member in at least one element of P_1. Let δ be half of the minimum of the non-zero differences between successive components of the excess vector of x. Assume there is a player j which is a member of each coalition in P_1. Then there are three possibilities:

1. *There exist a player i such that $x_i > 0$ and i is not in every member of P_1.* If j features in all coalitions in P_1, then players i other than j such that $x_i > 0$ can donate $\frac{\delta}{n}$ weight to j which increases the payoffs of all coalitions in P_1. This is a contradiction as x is a least core payoff.
2. *Any player i other than j such that $x_i > 0$ is in every member of P_1.* Let the set of such players be J'. Then we prove that j is a vetoer which is equivalent to saying that $v(N \setminus \{j\}) = 0$. For the sake of contradiction, assume that $v(N \setminus \{j\}) = 1$. Then $x(N \setminus \{j\}) = x(J')$. Since we have that $J' \subseteq S$ for all $S \in P_1$ and since $v(N \setminus \{j\}) = 1$, we know that $N \setminus \{j\} \in P_1$. This is a contradiction as there exists a coalition $N \setminus \{j\}$ which also gets the worst excess
3. *There exists no player i other than j such that $x_i > 0$.* But if this happens, $x_j = 1$. This implies $x(N \setminus \{j\}) = 0$. Also $v(N \setminus \{j\}) = 0$ because if $v(N \setminus \{j\}) = 1$, then $N \setminus \{j\}$ has the minimum possible excess but does not include j. Therefore, there exists a coalition $N \setminus \{j\}$ which also gets the worst excess (0 in this case).

□

Proposition 11.24. *The following problems can be solved in polynomial time for simple games represented by* (N, W^m)*:*

1. *EMPTY-ϵ-CORE*
2. *IN-ϵ-CORE*
3. *CONSTRUCT-ϵ-CORE*
4. *IN-LEAST-CORE*
5. *CONSTRUCT-LEAST-CORE*

Proof. Although the solution lies in solving the single LP (11.1), the constraints include 2^n constraints concerning the coalitions. However, for simple games, it is sufficient to only consider the minimal winning coalitions in the LP. This is because for a losing coalition S, $v(S) = 0$, so $x(S) \geq v(S)$. Moreover, if S is a minimal winning coalition and $x(S) \geq v(S) - \epsilon$, then for any $S' \supset S$, $x(S') \geq v(S') - \epsilon$. □

Corollary 11.25. *The following problems can be solved in polynomial time for simple games represented by* (N, W)*:*

1. *EMPTY-ϵ-CORE*
2. *IN-ϵ-CORE*
3. *CONSTRUCT-ϵ-CORE*
4. *IN-LEAST-CORE*
5. *CONSTRUCT-LEAST-CORE*

Denote the length of a game v by $l(v)$. We present a proposition which relates the length of a simple game to the least core elements of the game.

Proposition 11.26. *Let* $x = (x_1, \ldots x_n)$ *be an element of the least core of* v *where* $x_1 \geq x_2 \ldots \geq x_n$*. Let* $-\epsilon_1(v)$ *be the worst excess of* x*. Then,*

$$l(v) \geq \frac{1 - \epsilon_1(v)}{x_1}.$$

Proof. Let S be a winning coalition of length $l(v)$. Since $x(C) - v(C) \geq -\epsilon_1(v)$ for all $C \subseteq N$, then $x(S) - 1 \geq -\epsilon_1(v)$. This implies that there exists at least one player $i \in S$ such that $x_i \geq \frac{1-\epsilon_1(v)}{l}$. Therefore, $x_1 \geq x_i \geq \frac{1-\epsilon_1(v)}{l(v)}$. Thus,

$$l(v) \geq \frac{1 - \epsilon_1(v)}{x_1}.$$

□

For certain classes of simple games, the inequality in Proposition 11.26 turn out to be equalities. If v is singleton game, then $l(v) = \frac{1-\epsilon_1(v)}{x_1} = 1$. Similarly, if v is a unanimity game, $l(v) = \frac{1-\epsilon_1(v)}{x_1} = n$.

Proposition 11.27. *If computing the length of a simple game* (N, v) *is NP-hard, then IN-*ϵ*-CORE for* (N, v) *is NP-hard.*

Proof. Consider the payoff $x = (\frac{1}{n}, \ldots, \frac{1}{n})$ for (N, v). Denote the length of (N, v) by $l(v)$. The payoff of the smallest winning coalition is $\frac{l(v)}{n}$. The worst excess of (N, v) for payoff x is $\frac{l(v)}{n} - 1$.

The payoff x is in the ϵ-core if and only if $\frac{l(v)}{n} \geq 1 - \epsilon$. If there is an oracle to compute IN-ϵ-CORE in polynomial time, then by using different values of ϵ, binary search can be used to compute the $l(v)$. Therefore computing $l(v)$ reduces to solving IN-ϵ-CORE. Since $l(v)$ is NP-hard to compute, IN-ϵ-CORE is NP-hard.
□

For many important simple coalitional games such as simple coalitional skill games, threshold graph games and threshold network flow games, computing the length is NP-hard [12]. Proposition 11.27 implies that if the length of a simple game is NP-hard and unless P=NP, then there is no polynomial time separation oracle to solve the least core LP. This means that, if a polynomial time algorithm does exist, one needs to make extra use of the combinatorial structure of the underlying game.

Let (N, v) be a simple game and x be any payoff of (N, v) we will denote $1 - \epsilon_1(x, v)$ by $\delta_1(x, v)$ and $1 - \epsilon_1(v)$ by $\delta_1(v)$. The value $\delta_1(x, v)$ is the payoff of any coalition with the worst excess.

Let (N, v) be a simple game and x be any payoff of (N, v). We will denote $1 - \epsilon_1(x, v)$ by $\delta_1(x, v)$ and $1 - \epsilon_1(v)$ by $\delta_1(v)$. The value $\delta_1(x, v)$ is the payoff of any coalition with the worst excess.

Lemma 11.28. *Let (N, v) be a simple game and x be any efficient payoff of (N, v). Consider the game $(N \cup \{n+1\}, v')$ which is obtained by adding a passer player $n+1$ to the game (N, v). For any efficient payoff x' for $(N \cup \{n+1\}, v')$, if $x'_{n+1} = a$ and $x'_i = (1-a)x_i$ for $i \in N$, then*

1. $\delta_1(x', v') = \mathsf{Min}(a, (1-a)\delta_1(x, v))$.
2. *If x' is a least core payoff of $(N \cup \{n+1\}, v')$, then $a = (1-a)\delta_1(x, v)$ and $x'_{n+1}(v') = \delta_1(v')$.*

Proof. Since $\{n+1\}$ is a winning coalition, the worst excess of $\{n+1\}$ for payoff x' is $a - 1$ which implies that $\delta_1(x', v') \leq a$.

For $S \subseteq N$, any coalition $S \cup \{n+1\}$ is not a minimal winning coalition. Therefore, to examine other coalitions with the worst excess in $(N \cup \{n+1\}, v')$ for payoff x', we look at subsets of N. The worst payoff for winning coalitions among N is then $(1-a)\delta_1(x, v)$. This implies that $\delta_1(x', v') \leq (1-a)\delta_1(x, v)$. Since all subsets of $N \cup \{n+1\}$ have been considered, $\delta_1(x', v') = \mathsf{Min}(a, (1-a)\delta_1(x, v))$.

We now prove that payoff x' is a least core payoff of $(N \cup \{n+1\}, v')$, only if $a = (1-a)\delta_1(x, v)$ and $x'_{n+1}(v') = \delta_1(v')$.

The value $\delta_1(x', v')$ is maximized only when $a = (1-a)\delta_1(x, v)$. Also, $\delta_1(x', v')$ is maximum only when the optimum payoff $\delta_1(v')$ is given to player $n+1$, i.e., when $x'_{n+1} = a = \delta_1(v')$. $\square$

We define a representation of a simple game as *passer-reasonable* if, for any simple game, the game with a newly added passer can also be represented and there is at most polynomial blowup. WVGs, MWVGs, and coalitional skill games are examples of passer-reasonable representations.

Deng and Fang [58] note that *"the most natural problem is how to efficiently compute the value ϵ_1 for a given cooperative game. The catch is that the computation of ϵ_1 requires one to solve a linear program with exponential number* *of constraints.*" It is not clear that the least core worst excess can be computed efficiently even if a least core payoff is given. However, we have the following result.

Proposition 11.29. *An oracle to compute a least core payoff for a simple game in any passer-reasonable representation can be used to compute the worst excess of a least core payoff.*

Proof. Consider a game (N, v) in a passer-reasonable representation. Use the oracle to compute $x = (x_1, \ldots, x_n)$, a least core payoff of (N, v).

Denote the worst excess (as yet unknown) of x by $-\epsilon_1(v)$ and $1 - \epsilon_1(v)$ by $\delta_1(v)$. Then, we know that $\delta_1(x, v) = \delta_1(v)$. Form a new game $(N \cup \{n+1\}, v')$ by adding a passer player $n+1$ to the game such that $v'(\{n+1\}) = 1$ and $v'(S) = 1$ if and and only if $v(S) = 1$ for all $S \subseteq N$. Since (N, v) is in a passer-reasonable representation, $(N \cup \{n+1\}, v')$ can also be represented by a passer-reasonable representation.

Use the oracle to compute $x' = (x'_1, \ldots x'_{n+1})$, a least core payoff of $(N \cup \{n+1\}, v')$. From Lemma 11.28, we know that $x'_{n+1}(v') = \delta_1(v')$ and $x'_{n+1} = (1 - x'_{n+1})\delta_1(x, v))$. This means that,

$$\frac{x'_{n+1}(v')}{1 - x'_{n+1}(v')} = \delta_1(v) = 1 - \epsilon_1(v) \tag{11.2}$$

From (11.2), we know that $\epsilon_1(v) = 1 - \delta_1(v)$ can be computed by adding a passer to (N, v) to form game $(N \cup \{n+1\}, v')$ and then computing $x'_{n+1}(v')$. $\square$

11.8 Nucleolus

Elkind et al. [68] showed that CONSTRUCT-NUCELULOS is NP-hard for WVGs. We check the complexity of CONSTRUCT-NUCLEOLUS for a simple game represented by (N, W) and (N, W^m). It is well-known that the nucleolus is unique [189]. The nucleolus can be computed by solving a series of linear programs [78]:

Initially, $J_0 = \{\emptyset, N\}$ and $\epsilon_0 = 0$. The value ϵ_r is the optimal value of LP_r and $J_r = \{S \in 2^N : x(S) = v(S) - \epsilon_r$ for every $x \in X_r\}$ where $X_r\{x \in I(v) : (x, \epsilon_r)$ is an optimal solution of $LP_r\}$. Kopelowitz [122] showed that a maximum of $n - 1$ LP program iterations need to be run before one arrives at the solution x^* which is the nucleolus. The solution to LP_1 is the least core of the game. Since, nucleolus is a least-core payoff, if computing the least core of a game is NP-hard, then it implies that computing the nucleolus is NP-hard. The following is the description of LP_k in the series of LPs.

$$\begin{aligned} &\min \epsilon \\ &\text{s.t. } \; x(S) = v(s) - \epsilon_r \;, \text{ for all } S \in J_r, r = 0, \ldots, k-1 \;, \\ &\qquad x(S) \geq v(S) - \epsilon \;, \text{for all } S \in 2^N \setminus \textstyle\bigcup_{r=0}^{r=k-1} J_r, \\ &\qquad x \in I(v) \;. \end{aligned} \tag{11.3}$$

The nucleolus is a complex solution and no polynomial algorithm is known for computing the nucleolus in general. The computation of the nucleolus for different cooperative games has attracted much attention [126]. Only for some special classes of cooperative game can the nucleolus be computed efficiently, for example, standard tree games [102], convex games [125], weighted voting games with small weights [70] and assignment games [198]. Computing the nucleolus is NP-hard for min-cost spanning tree games [73], general flow games and linear production games [59]. We consider using the following meta-theorem to compute the nucleolus.

Theorem 11.30. *(Elkind and Pasechnik: Theorem 5 in [70]) Given a coalitional game G, suppose that we can, for any payoff vector p, identify the top n distinct deficits under p, as well as the number of coalitions that have these deficits in polynomial time. Then we can compute the nucleolus in polynomial time.*

However, Theorem 11.30 cannot be used to compute the nucleolus for simple games represented by minimal winning coalitions or winning coalitions. For example, if $|W|$ is small enough, then computing the condition in Theorem 11.30 reduces to a knapsack problem.

If CONSTRUCT-NUCLEOLUS can be solved efficiently, this also means that IN-NUCLEOLUS can be solved efficiently by first computing the nucleolus and then comparing it with the payoff in consideration. This also implies that CONSTRUCT-BARGAINING-SET and CONSTRUCT-KERNEL are easy. It is well known that in a simple game, if a player is a dummy, it gets zero payoff not only in the nucleolus but in every member of the kernel. We now give an example of a simple game where there are no vetoers and no dummies but still there is a player who gets zero nucleolus payoff:

Example 11.31. Let $N = \{1, 2, 3, 4\}$. Let

$$W^m(v) = \{\{1, 2\}, \{1, 3\}, \{2, 3, 4\}\}.$$

The game has no vetoer and no dummy. Let $x = (x_1, x_2, x_3, x_4)$ be the nucleolus payoff of the simple game v. From the relative powers of the players, we can see that $x_1 \geq x_2 = x_3 \geq x_4$. Since the sum of the nucleolus payoffs of all players is equal to one, $x_1 + 2x_2 + x_4 = 1$. The two candidates for the value of the first element of the excess vector of x are $(x_1 + x_2) - 1$ and $(2x_2 + x_4) - 1$. For x to be the nucleolus, the excess vector of x must lexicographically be the greatest. This is the case when $(x_1 + x_2) - 1 = (2x_2 + x_4) - 1$. This means that $x_2 = x_3 = \frac{1}{3} - \frac{2}{3}x_4$ and $x_1 = \frac{1}{3} + \frac{1}{3}x_4$. It is easy to work out that the excess vector of x is maximum when $x_4 = 0$.

Peleg [175] proved that for certain WVGs, the nucleolus is simply equal to the normalized weights of the players:

Proposition 11.32. *(Peleg [175]) Consider a constant sum homogeneous weighted voting game $v = [q; w_1, \ldots, w_n]$ in which each dummy gets zero weight. Then the nucleolus is equal to* $(w_1/w(N), \ldots, w_n/w(N))$.

We use some examples and comments to clarify certain concepts which have not always been used clearly in the game theory literature. For example, on page 5 in [68], it is mentioned that Peleg's proposition holds for any constant sum weighted voting games. This may not be true if the dummy players are not given zero weight. Moreover, in some places in the literature, Peleg's proposition is assumed to hold for any homogeneous game. A homogeneous game does not have to be constant sum: for example $[2; 2, 2, 2]$. Also, a homogeneous representation can have a dummy with non-zero weight: for example $[4, 2, 2, 1]$. Proposition 11.32 does not necessarily hold if all the conditions mentioned are not satisfied. The proposition says that there is at least one representation of a constant sum, homogeneous weighted voting game which coincides with the nucleolus. On the face of it, it seems that the proposition may provide an easy method to compute a normalized WVG representation for a target nucleolus value. However, for any nucleolus vector, it is not certain whether the nucleolus payoff is achievable. Moreover, even if the nucleolus payoff is feasible for some WVG, we need to find a suitable quota which is the weight of all minimal winning coalitions and the resultant WVG is constant-sum. The following example shows that finding a suitable quota may not be possible.

Example 11.33. Consider the target nucleolus $(4/10, 3/10, 2/10, 1/10)$. Then the WVG in consideration is $v = [q; 4, 3, 2, 1]$ where $6 \leq q \leq 10$. However, for $q = 6, 7, 8, 9, 10$, it is easy to check that the following two conditions are not met: v is homogeneous and dummies get zero weight.

Definition 11.34. *The nucleolus-like payoffs are those least core payoffs for which the number of coalitions with the worst excess is minimum possible.*

Proposition 11.35. *For any monotone cooperative game (N, v) and nucleolus-like payoff x, assume that there exists a player i such that for all $S \in A_x^1(v)$, we have that $i \in S$. Then, for player j other than i, either for all $S \in A_x^1(v)$, $j \in S$ or we have that $x_j = 0$.*

Proof. Assume that there exists a coalition $S \in A_x^1(v)$ such that player $j \notin S$. Let $\delta = \frac{\epsilon_1 - \epsilon_2}{2}$. Then if j donates δ amount of its payoff to i, this reduces the number of $-\epsilon_1$-coalitions. This cannot be since x is nucleolus-like. □

Proposition 11.36. *For simple games represented by (N, W^m), CONSTRUCT-PRENUCLEOLUS and CONSTRUCT-NUCLEOLUS can be solved in polynomial time.*

Proof. It follows from Theorem 7 of [185] that, instead of examining 2^n coalitions, it is sufficient to examine a small collection of coalitions $\mathbb{B}$ to compute the nucleolus. The set $\mathbb{B}$ consists of MWCs plus certain other winning coalitions. From the definition of $\mathbb{B}$ in [185], it is easy to see that $\mathbb{B} \subset \mathbb{C}$ where

$$\mathbb{C} = W^m \cup \{S \cup \{i\} | i \in N, S \in W^m\}.$$

Therefore, the standard series of linear programs can be used to compute the nucleolus where in place of 2^n coalitions, only $|\mathbb{C}|$ coalitions are considered. □

This implies that for a simple game represented by (N, W^m), IN-PRENUCLEOLUS and IN-NUCLEOLUS can be solved in polynomial time and CONSTRUCT-PRENUCLEOLUS, CONSTRUCT-NUCLEOLUS, IN-PRENUCLEOLUS and IN-NUCLEOLUS can be solved in polynomial time for simple game represented by (N, W).

11.9 Kernel and bargaining set

In this section we examine the computational complexity of questions related to the kernel and bargaining set of simple games. We also consider the prenucleolus. The relation between the kernel and prekernel is intricate. Maschler et al. [142] note that the kernel is not a subset of the prekernel. Prekernel does not have to be individually rational so is not a subset of the kernel. However the intersection of the prekernel with the set of imputations is a subset of the kernel. Moreover, the parts of the kernel and the prekernel inside any ϵ-core always coincide [142]. Both solution concepts are closely related and an imputation that is in the prekernel is also an imputation of the kernel. However, the prekernel also contains payoffs which are not imputations. For a non-zero-normalized simple game, a kernel may not exist. In fact, the prekernel satisfies individual rationality on the class of zero-monotonic games.

If there are vetoers in a simple game, then the kernel of a simple game is simply a uniform distribution among the vetoers and no payoff for the vetoers [9].

A naive algorithm to solve IN-KERNEL will take $2^n\binom{n}{2}$ time, as for each pair of players, their maximum surpluses over each other are compared.

Proposition 11.37. *For simple games represented by* (N, W^m)*, IN-KERNEL and IN-PREKERNEL can be solved in polynomial time.*

Proof. Let $x = (x_1, \ldots, x_n)$ be a potential payoff. Then in order to check whether x is in the kernel or the prekernel, it is sufficient to compute all the $n(n-1)$ surpluses: $s^v_{ij}(x) = \max\{v(S) - x(S) | S \subseteq N \setminus \{j\}, i \in S\}$. One can compute $s^v_{ij}(x)$ simply by examining MWCs which exclude j because for any MWC S and a coalition $S \subset S'$, $v(S) - x(S) = 1 - x(S) \leq 1 - x(S') = v(S') - x(S')$. A problem may arise if it is the case that $\forall S \in W^m$, $j \in S$. Then we consider the singleton coalition of i to compute $s^v_{ij}(x)$ because it gives the maximum surplus of $-x_i$. Another problem may happen if $\forall S' \in W^m$ such that $j \notin S'$, we find that $i \notin S'$. In that case, we consider the coalition S' which has the minimum $x(S)$. Therefore $S' \cup \{j\}$ gives us $s^v_{ij}(x)$. □

Corollary 11.38. *For simple games represented by* (N, W)*, IN-KERNEL and IN-PREKERNEL can be solved in polynomial time.*

Proposition 11.39. *For simple games represented by* (N, W^m)*, CONSTRUCT-PREKERNEL and CONSTRUCT-KERNEL can be solved in polynomial time.*

Proof. From [74], we know that an element in the so-called *lexicographic prekernel* can be computed in polynomial time if the surpluses $s_{ij}(x)$ corresponding to a given allocation x can be computed in polynomial time. From Prop 11.37, it then follows that an element in the lexicographic kernel can be computed in polynomial time. Since the lexicographic prekernel is a subset of the intersection of the prekernel and the least core, we can compute an element x in the prekernel of the (N, W^m). Moreover, as long as $I(v)$ is non-empty, $K(v)$ is non-empty and x is also in the kernel of (N, W^m), because the intersection of the kernel and the least core coincides with the intersection of the prekernel and the least core for games with non-empty $I(v)$. □

Corollary 11.40. *The following statements follow from Proposition 11.39:*

1. *For simple games represented by* (N, W)*, CONSTRUCT-KERNEL and CONSTRUCT-PREKERNEL can be solved in polynomial time.*
2. *For simple games represented by* (N, W^m) *and* (N, W)*, CONSTRUCT-BARGAINING-SET can be solved in polynomial time.*

For simple games, if the core is non-empty, the bargaining set is equivalent to the core [65]. This means that for a game in which the core is non-empty, problems associated with the core are equivalent to problems associated with the bargaining set. For a unanimity game, any feasible imputation is in the bargaining set. This is because there is no scope for a valid objection. Let (N, v) be a simple game with no passers. Then symmetric players get equal kernel payoff. Kernel payoffs obey desirability relations, i.e., if i is more desirable than j, then $x_i \geq x_j$.

If two players are symmetric, then they get equal payoff in the kernel and prekernel. However, if two players get equal payoff in the prekernel or kernel, it does not imply that the players are symmetric, even if the game is linear and has no vetoers:

Example 11.41. Take WVG $[5; 3, 2, 2, 1]$. The imputation, $(1/3, 1/3, 1/3, 0)$ is a prekernel imputation.

11.9.1 Wolsey's theorem

Consider WVGs with non-increasing weights. A payoff vector $x = (x_1, \ldots, x_n)$ is *homogeneous* if each player receives either 0 or a fixed amount $1/r$ for some $r \leq n$. Moreover, for set $R = \{1, \ldots, r\} \subseteq N$,, we call the set $T(R) = \{1, \ldots, k\} \cup \{r + 1, \ldots, n\}$, the *cover* of R where k is the maximum with the property that $\sum_{j \in T(R)} w_j < q$. Then Wolsey claimed the following:

Claim. (Wolsey [217]) Let $x = (x_1, \ldots, x_n)$ be a homogeneous payoff in WVG $[q; w_1, \ldots, w_n]$ where $q \geq w_1 \geq \ldots w_n$. Then the payoff with $x_i = 1/r$ for $i \in R$ and $x_i = 0$ for $i \in N \setminus R$ is in the kernel if and only if

1. $w(T(R)) + w_r - w_{r+1} \geq q$ and
2. $w(T(R)) - w_1 + w_{k+1} + w_r \geq q$.

The paper has been cited in [124, 70, 68, 153]. The following counter example shows that at least the left to right implication does not hold in general.

Example 11.42. We consider the WVG $v = [12; 8, 4, 2, 1]$. The WVG v meets the condition of the claim that $q \geq w_1$. This means that $v(\{i\}) = 0$ for all $i \in N$. We see that players 1 and 2 are vetoers. Player 1 is a vetoer because $\{2, 3, 4\}$ is a losing coalition as $w(\{2, 3, 4\}) = 7 < 12$, and player 2 is a vetoer because $\{1, 3, 4\}$ is a losing coalition as $w(\{1, 3, 4\}) = 11 < 12$. Since players 1 and 2 are vetoers, the nucleolus payment of the game is $(1/2, 1/2, 0, 0)$. Since the nucleolus is a member of the kernel, the claim implies that both the conditions in the claim should be satisfied for $r = 2$.

We now consider the claim for the conditions that $r = 2$. If $r = 2$, then $T(R) = \{1, 3, 4\}$ with $k = 1$. Therefore $w(T(R)) = 11$. We consider the first condition. Since $w(T(R)) + w_r - w_{r+1} = 11 + 4 - 2 = 13 > 12$, the first condition is satisfied. However, the second condition is not satisfied: $w(T(R)) - w_1 + w_{k+1} + w_r = 11 - 8 + 4 + 4 = 11 \ngeq 12$.

From the proof in [217], it is evident that the case, when players i and j are vetoers, is ignored in proving both left-to-right and right-to-left implications. Therefore, the required added condition for Wolsey's theorem would be that there are no vetoers:

Theorem 11.43. *Let $x = (x_1, \ldots, x_n)$ be a homogeneous payoff in WVG $[q; w_1, \ldots, w_n]$ where there are no vetoers and $q \geq w_1 \geq \ldots w_n$. Then the payoff with $x_i = 1/r$ for $i \in R$ and $x_i = 0$ for $i \in N \setminus R$ is in the kernel if and only if*

1. $w(T(R)) + w_r - w_{r+1} \geq q$ *and*
2. $w(T(R)) - w_1 + w_{k+1} + w_r \geq q$.

11.10 Cost of stability

Unlike normal form games, where a mixed-strategy Nash equilibrium always exists, coalitional games can be unstable if the core is empty. If the core of a coalitional game is empty, it is hard to ensure that players do not break off from the grand coalition to maximize their payoff. One recent proposal [23] to take care of this problem is the idea of an external authority paying a *supplemental payment* to incentivize the players to cooperate in a stable manner. This payment is denoted by $\triangle$ and distributed in some way among the players. We use the same definitions as introduced in [23].

Definitions 11.44. *For a given coalitional game $G = (N, v)$, the* adjusted coalitional game $G(\triangle) = (N, v')$ *is obtained by setting $v'(S) = v(S)$ for $S \subset N$ and $v'(N) = v(N) + \triangle$. Any payoff which is in the core of $G(\triangle) = (N, v')$ is a* superimputation*. The* cost of stability (CoS) *of a game is the minimum supplemental payment $CoS(G)$ such that $G(CoS(G))$ has a nonempty core.*

If the core of a game is nonempty, then CoS is 0. It is easy to see that $CoS(G)$ is the solution of the following LP:

$$\begin{aligned} &\min \triangle \\ &\text{s.t.}\ \ x(S) \geq v(S)\ ,\ \text{for all } S \subset N\ , \\ &\qquad x_i \geq 0\ , \text{for all } i \in N, \\ &\qquad x(N) = v(N) + \triangle\ . \end{aligned} \tag{11.4}$$

We now the consider the following natural computational problems related to the cost of stability:

Name: CoS
Instance: Coalitional game (N, v)
Question: Compute CoS

Name: SUPERIMP
Instance: Coalitional game $G = (N, v)$, supplement payment $\triangle$ and superimputation $(x_1, \ldots, x_n)$ in $G(\triangle)$
Question: Is $x \in CORE(G(\triangle))$?

It is known that solving CoS and SUPERIMP for WVGs is co-NP-hard [23]. However, for other representations, the problems are tractable:

Proposition 11.45. *For a simple game represented by* (N, W^m)*, solving SUPER-IMP and CoS is in P.*

Proof. The solution for both general problems seems to require considering 2^n constraints concerning the coalitions. However, for simple games, it is sufficient to consider only the minimal winning coalitions. This is because for a losing coalition S, $v(S) = 0$, so $x(S) \geq v(S)$. Moreover, if S is a minimal winning coalition and $x'(S) \geq v(S)$, then for any $S' \supset S$, $x'(S') \geq v(S')$. □

Propositions 11.46 draws the connection between the cost of stability and the least core. The similarity in LP (11.1) for the least core and LP (11.4) leads us to the following proposition:

Proposition 11.46. *For a monotone cooperative game* (N, v)*, if the separation oracle O for a least core LP can be constructed and be solved in polynomial time, then for* (N, v)*, SUPERIMP and CoS are in P.*

Proof. Consider payoff $(x_1, \ldots, x_n)$. Then oracle O, can check in polynomial time whether $x(S) - v(S) \geq -\epsilon$ for all $S \subset N$, or find a violated constraint otherwise. Then O can be used to solve SUPERIMP for (N, v). Also, O can be used as a separation oracle to solve LP (11.4). □

Corollary 11.47. *Solving SUPERIMP and CoS are in P for the following games:*

1. *WVG with weight values polynomial in n,*
2. *Weighted matching games.*

Proof. The separation oracle O for a least core LP can be constructed and solved in polynomial time for WVG with weight values polynomial in n [68] and weighted matching games [118]. □

Is it easy to notice the following relation between the cost of stability and the worst excess of an element in the least core: $\epsilon_1(G) \leq CoS(G) \leq n(\epsilon_1(G))$. In order to change G to a balanced $G(\triangle)$, we need to at least add ϵ_1 amount to S_1 so that $x'(S_1) \geq 1$. Therefore $\epsilon_1 \leq \triangle$. Take an element $x = (x_1, \ldots, x_n)$ such that

$x \in LC(G)$. Then $x(S) - v(S) \geq -\epsilon_1$. Consider $x' = (x'_1, \ldots x'_n)$ where $x'_i = x_i + \epsilon_1$. Take any coalition $S \subset N$. Then, $x'(S) - v(S) = x(S) + |S|\epsilon_1 - v(S) \geq 0$.

There may be multiple of ways of distributing the payoffs after the cost of stability has been paid. We propose a natural and desirable solution for any cooperative game called the *super-nucleolus*. The super-nucleolus is the nucleolus of a cooperative game G if the core is nonempty and is the nucleolus of $G(CoS(G))$ if the core of G is empty. Since the core of $G(CoS(G))$ is nonempty, it may be easier to compute the super-nucleolus than the nucleolus of G in certain games.

11.11 Conclusion and open problems

One conclusion is that (N, W^m) is a comprehensive representation of simple games which allows the efficient computation of almost all solutions. Similarly, Deng [58] states that, in all problems known, the concepts of the core, the bargaining set, and the stable set should be in increasing order of complexity. It will again be useful to characterize this for any cooperative game. In this context, it is shown that computing the Shapley-Shubik index is at least as hard as computing the Banzhaf index. It will be interesting to prove that for any representation of a monotone coalitional game, computing a least-core payoff has the same computational complexity as computing the cost of stability. Another conjecture is that if computing the length of a simple game is NP-hard, then computing a least core payoff is NP-hard.

Table 11.2. Complexity of cooperative game solutions in simple games

	(N, W)	(N, W^m)	WVG	MWVG
EMPTY-SHAPLEYVALUE	always nonempty for any cooperative game			
IN-SHAPLEYVALUE	P	?	NP-hard	NP-hard
CONSTRUCT-SHAPLEYVALUE	P	#P-complete	#P-complete [60]	#P-complete
ISZERO-SHAPLEYVALUE	P	P	NP-hard	NP-hard
EMPTY-CORE	P	P	P [68]	P
IN-CORE	P	P	P [68]	P
CONSTRUCT-CORE	P	P	P [68]	P
ISZERO-CORE	P	P	P [68]	P
EMPTY-ϵ-CORE	P	P	NP-hard [68]	NP-hard
IN-ϵ-CORE	P	P	NP-hard	NP-hard
CONSTRUCT-ϵ-CORE	P	P	NP-hard	NP-hard
ISZERO-ϵ-CORE	N/A			
EMPTY-LEAST-CORE	always nonempty for any cooperative game			
IN-LEAST-CORE	P	P	NP-hard [68]	NP-hard
CONSTRUCT-LEAST-CORE	P	P	NP-hard [68]	NP-hard
ISZERO-LEAST-CORE	N/A			
EMPTY-NUCLEOLUS	always nonempty if no passer			
IN-NUCLEOLUS	P	P	?	?
CONSTRUCT-NUCLEOLUS	P	P	NP-hard [68]	NP-hard
ISZERO-NUCLEOLUS	P	P	NP-hard [68]	NP-hard
EMPTY-STABLE-SET	always nonempty in simple games			
IN-STABLE-SET	P	P	P	P
CONSTRUCT-STABLE-SET	P	P	P	P
ISZERO-STABLE-SET	N/A			
EMPTY-BARGAINING-SET	always nonempty if no passer			
IN-BARGAINING-SET	?	?	?	?
CONSTRUCT-BARGAINING-SET	P	P	?	?
ISZERO-STABLE-SET	N/A			
EMPTY-KERNEL	always nonempty if no passer			
IN-KERNEL	P	P	?	?
CONSTRUCT-KERNEL	P	P	?	?
ISZERO-KERNEL	N/A			
EMPTY-PREKERNEL	always nonempty in any cooperative game			
IN-PREKERNEL	P	P	?	?
CONSTRUCT-PREKERNEL	P	P	?	?
ISZERO-PREKERNEL	N/A			
EMPTY-PRENUCLEOLUS	always nonempty in any cooperative game			
IN-PRENUCLEOLUS	P	P	?	?
CONSTRUCT-PRENUCLEOLUS	P	P	NP-hard	NP-hard
ISZERO-PRENUCLEOLUS	P	P	NP-hard	NP-hard
CORE-STABILITY	P	P	P	P
CoS	P	P	coNP-hard [23]	coNP-hard
SUPERIMP	P	P	coNP-complete [23]	coNP-complete

12

Power indices of spanning connectivity games

> *The study of networks pervades all of science, from neurobiology to statistical physics.*
> - *Steven Strogatz* [199]

> *If the Internet is the next great subject for Theoretical Computer Science to model and illuminate mathematically, then Game Theory, and Mathematical Economics more generally, are likely to prove useful tools.*
> - *Christos Papadimitriou* [167]

Abstract We consider a simple coalitional game, called the spanning connectivity game (SCG), based on an undirected, unweighted multigraph, where edges are players. We examine the computational complexity of computing the voting power indices of edges in the SCG. It is shown that computing Banzhaf values is #P-complete and computing Shapley-Shubik indices or values is NP-hard for SCGs. Interestingly, Holler indices and Deegan-Packel indices can be computed in polynomial time. Among other results, it is proved that Banzhaf indices can be computed in polynomial time for graphs with bounded treewidth.

12.1 Introduction

In this chapter, we study the natural problem of computing the influence of edges in keeping an unweighted and undirected multigraph connected. Game theorists have studied notions of efficiency, fairness and stability extensively. Therefore, it is only natural that when applications in computer science and multiagent systems require fair and stable allocations, social choice theory and cooperative game theory provide appropriate foundations. For example, a network administrator with limited resources to maintain the links in the network may decide to commit resources to links according to their connecting ability. A spy network comprises communication channels. In order to intercept messages on the channels, resources may be utilized according to the ability of a channel to connect all groups. In a social network, we may be interested in checking which connections are more important in maintaining connectivity and hence contribute more to social welfare.

Our model is based on undirected, unweighted and connected multigraphs. All the nodes are treated equally, and the importance of an edge is based solely on its ability to connect all the nodes. Using undirected edges is a reasonable assumption in many cases. For example, in a social network, relations are usually mutually formed.

We use a multigraph as a succinct representation of a simple coalitional game called the *spanning connectivity game (SCG)*. The players of the game are the edges of the multigraph. The importance of an edge is measured by computing its voting power index in the game.

The whole chapter is concerned with computing solutions for SCGs. In Section 12.2, a summary of related work is given. In Section 12.3, preliminary definitions related to graph theory and coalitional games are given, and we define SCGs. Section 12.4 presents hardness results for computing Banzhaf values and Shapley-Shubik indices. In Section 12.5, positive computational results for Banzhaf values and Shapley-Shubik indices are provided for certain graph classes. Section 12.6 presents a polynomial-time algorithm to compute Holler indices and Deegan-Packel indices. In Section 12.7, a summary of results is given and future work is discussed.

12.2 Related work

Power indices such as the Banzhaf and Shapley-Shubik indices have been used to gauge the power of a player corporate networks [52]. These indices have recently been used in network flow games [24], where the edges in the graph have capacities and the power index of an edge signifies the influence that an edge has in enabling a flow from the source to the sink. Voting power indices have also been examined in vertex connectivity games [27] on undirected, unweighted graphs; there the players are nodes, which are partitioned into primary, standard, and backbone classes.

The study of cooperative games in combinatorial domains is widespread in operations research [38, 54]. Spanning network games have been examined previously [101, 210] but they are treated differently, with weighted graphs and *nodes* as players (not *edges*, as here). The SCG is related to the all-terminal reliability model, a non-game-theoretic model that is relevant in broadcasting [209, 29]. Whereas the reliability of a network concerns the overall probability of a network being connected, this chapter concentrates on resource allocation to the edges. A game-theoretic approach can provide fair and stable outcomes in a strategic setting. The hardness results in this chapter are a strengthening of the hardness results for the more general, min-base games, introduced in [155].

12.3 Preliminaries

12.3.1 Graph theory

Definition 12.1. *A* multigraph $G = (V, E, s)$ *consists of a simple underlying graph* (V, E) *with a multiplicity function* $s : E \mapsto \mathbb{N}$ *where* $\mathbb{N}$ *is the set of natural numbers excluding* 0*. Let* $|V| = n$ *and* $|E| = m$*. For every underlying edge* $i \in E$*, we have* s_i *edges in the multigraph. The multigraph has a total of* $M = \sum_{i \in E} s_i$ *edges and the set of all of these edges is* $\mathbb{M}$*.*

We note that $G = (V, E, s)$ is a compact representation of multigraphs which can contain exponential number of parallel edges.

Definition 12.2. *A* subgraph $G' = (V', \mathbb{M}')$ *of a graph* $G = (V, \mathbb{M})$ *is a graph where* V' *is a subset of* V *and* $\mathbb{M}'$ *is a subset of* $\mathbb{M}$ *in which case the vertex set of* $\mathbb{M}'$ *is a subset of* V'. *A subgraph* H *is a* connected spanning subgraph *of a graph* G *if it is connected and has the same vertex set as* G.

12.3.2 Spanning connectivity game

For each connected multigraph (V, E, s) where s_i is the multiplicity of underlying edge $i \in E$, we define the SCG, *spanning connectivity game*, $(\mathbb{M}, v)$ with M players (one player corresponding to each parallel edge) and valuation function v, defined as follows for $S \subseteq E$:

$$v(S) = \begin{cases} 1, \text{if there exists a spanning tree } T = (V, E') \text{ such that } E' \subseteq S \\ 0, \text{otherwise} \end{cases}$$

It is easy to see that for a connected graph with more than one vertices, the SCG $(\mathbb{M}, v)$ is a simple game because the outcome is binary, v is monotone, $v(\emptyset) = 0$ and $v(E) = 1$. We consider power indices and cooperative game solutions for the edges in the SCG.

12.4 Complexity of computing power indices

We define the problems of computing the power indices of the edges in the SCG. For any power index X (e.g. Banzhaf value, Banzhaf index, Shapley-Shubik index etc.) we define the problem SCG-X as follows:

Problem: SCG-X
Instance: Multigraph G
Output: For the SCG corresponding to G, compute X for all the edges.

Computation of power indices of SCGs has relations with computation of reliability in networks. We will use these connections in some of our computational results. We will now present the prerequisite background of reliability computation. We represent a communication network as a multigraph, where an edge represents a connection that may or may not work. An edge is said to be *operational* if it works. For a given graph G, the reliability $Rel(G, \{p_i\})$ of G is the probability that the operational edges form a connected spanning subgraph, given that each edge is operational with probability p_i for $i = 1, \ldots m$.

Problem: Rational Reliability Problem
Instance: Multigraph G and $p_i \in \mathbb{Q}$ for all i, $1 \leq i \leq m$
Output: Compute $Rel(G, \{p_i\})$.

A special case of the reliability problem is when every edge has the same probability p of being operational. This is called the *Functional Reliability Problem*.

Definition 12.3. *Let N_i be the number of connected spanning subgraphs with i edges. Then the required output of the Functional Reliability Problem is the* reliability polynomial

$$Rel(G, p) = \sum_{i=0}^{m} N_i p^i (1 - p)^{m-i}.$$

Problem: Functional Reliability Problem
Instance: Multigraph G
Output: Compute the coefficients N_i of the reliability polynomial for all i, $1 \leq i \leq m$.

Ball [29] points out that an algorithm to solve the Rational Reliability Problem can be used as a sub-routine to compute all the coefficients for the Functional Reliability Problem. Moreover he proved that computing the general coefficient N_i is NP-hard and therefore computing the rational reliability of a graph is NP-hard. As we will see in Section 12.5, reliability problems have connections with computing power indices of SCG. We first prove that SCG-BANZHAF-VALUE is #P-complete.

Proposition 12.4. *SCG-BANZHAF-VALUE is #P-complete even for simple, bipartite and planar graphs.*

Proof. We present a reduction from the problem of counting connected spanning subgraphs. SCG-BANZHAF-VALUE is clearly in #P because a connected spanning subgraph can be verified in polynomial time. It is known that counting the total number of connected spanning subgraphs is #P-complete even for simple, bipartite and planar graphs ([28], p. 305). We now reduce the problem of computing the total number of connected spanning subgraphs to solving SCG-BANZHAF-VALUE. Take $G = (V, E)$ with n nodes and m edges. Transform graph G into $G' = (V \cup \{n+1\}, E \cup \{m+1\})$ by taking any node and connecting it to a new node via a new edge. Then the number of spanning subgraphs in G is equal to the Banzhaf value of edge $m+1$ in graph G'. This shows that SCG-BANZHAF-VALUE is #P-complete. □

Similarly, SCG-SS is NP-hard.

Proposition 12.5. *SCG-SS is #P-complete even for simple graphs.*

Proof. The proof follows from Theorem 3.29. We demonstrate the application of Theorem 3.29 by giving a complete proof. We show that computing the Shapley-Shubik indices is at least as hard as computing the total number of winning coalitions. Let N_i be the number of connected spanning subgraphs of G with i edges. We know that computing N_i is NP-hard [29]. We show that if there is an algorithm polynomial in the number of edges to compute the Shapley-Shubik index of all edges in the graph, then each N_i can be computed in polynomial time.

We obtain graph G_0 by the following transformation: for some node $v \in V(G)$, we link it by a new edge x to a new node v_x. Then, by definition, the Shapley-Shubik value $\kappa_x(G_0)$ of player x is $\sum_{r=0}^{m} r! N_r(|E(G)| - r)! = \sum_{r=0}^{m} r! N'_r$, where we write N'_r for $N_r(m-r)!$, for all r.

Similarly we can construct G_i by adding a path P_i of length i to v_x where P_i has no edge or vertex intersection with G. Therefore

$$\sum_{r=0}^{m}(r+i)! N'_r = \kappa_x(G_i). \tag{12.1}$$

For $i = 0, \ldots, m$, we get an equation of the form of (12.1) for each G_i. The left-hand side of the set of equations can be represented by an $(m+1) \times (m+1)$ matrix A where $A_{ij} = (i + j - 2)!$. The set of equations is independent because A has a non-zero determinant of $(1!2! \cdots m!)^2$ (see e.g. Theorem 1.1 [20]). If there is a polynomial time algorithm to compute the Shapley-Shubik index of each edge in a simple graph, then we can compute the right-hand side of each equation corresponding to G_i.

The biggest possible number in the equation is less than $(2m)!$ and can be represented efficiently. The biggest possible number in the equation is less than $(2m)!$ and can be represented efficiently. This follows from the fact that $m! \leq m^m$ and hence to represent $(2m)!$, one will use at most $\log_2((2m)^{2m}) = 2m(1 + \log_2 m) \leq 3m \log_2 m$ bits.

We can use Gaussian elimination to solve the set of linear equations in $O(m^3)$ time. Moreover, each number that occurs in the algorithm can also be stored in a number of bits quadratic of the input size (Theorem 4.10 [123]). Therefore SCG-SS is NP-hard. $\square$

12.5 Polynomial time cases

In this section, we present polynomial time algorithms to compute voting power indices for restricted graph classes including graphs with bounded treewidth. We first consider the trivial case of a tree. If the graph $G = (N, E)$ is a tree then there is a total of $n - 1$ edges and only the grand coalition of edges is a winning coalition. Therefore a tree is equivalent to a unanimity game. This means that each edge has a Banzhaf index and Shapley-Shubik index of $\frac{1}{n-1}$. In the case of the same tree structure but with multiple parallel edges, we refer to this multigraph as a *pseudo-tree*.

Proposition 12.6. *Let $G = (N, E, s)$ be a pseudo-tree such that the underlying edges are $1, \ldots, m$ with multiplicities $s_1, \ldots, s_m$. Then,*

$$\eta_{i_1} = \prod_{\substack{j=1 \\ j \neq i}}^{m} (2^{s_j} - 1). \tag{12.2}$$

Proof. Note that $m = n - 1$ in this case. Suppose edge i_1 is a parallel edge corresponding to edge i in the underlying graph. Edge i_1 is critical for a coalition C if the coalition C contains no edges parallel to i_1 but contains at least one subedge corresponding to each edge other than i. The number of such coalitions is $\prod_{\substack{j=1 \\ j \neq i}}^{m} (2^{s_j} - 1)$, which gives (12.2).
□

Proposition 12.7. *Let $G = (N, E, s)$ be a pseudo-tree such that the underlying edges are $1, \ldots, m$ with multiplicities $s_1, \ldots, s_m$ where $s = \sum_{i=1}^{m} s_i$. Then the Shapley-Shubik indices can be computed in time polynomial in the total number of edges and number of players.*

Proof. Denote by e_r the coefficient of x^r in

$$\prod_{\substack{1 \leq j \leq n-1 \\ j \neq i}} ((1 + x)^{s_j} - 1).$$

Then e_r is the number of coalitions with r edges which include at least one parallel edge for each underlying edge j except i. Denote by i_k the kth parallel edge of underlying edge $i \in E$. Then, by definition of the Shapley-Shubik value, for $1 \leq k \leq s_i$,

$$\kappa_{i_k}(G) = \sum_{r=n-2}^{s-s_i} e_r r!(s - r - s_i)!.$$

Assume that the total number of edges s is polynomial in input size. Then each of the values s_i are also polynomial in the input size. In that case, the generating function can be expanded in polynomial time and the Shapley-Shubik indices can be computed. □

We now consider graphs with bounded treewidth. Note that trees and pseudo-trees have treewidth 1.

Definition 12.8. *For a graph $G = (V, E)$, a* tree decomposition *is a pair (X, T), where $X = \{X_1, ..., X_n\}$ with $X_j \subset 2^V$ for $1 \leq j \leq n$, and T is a tree whose nodes are the subsets X_i with the following properties:*

1. $\bigcup_{1 \leq i \leq n} X_i = V$
2. *For every edge $(v, w) \in E$, there is a subset X_i that contains both v and w.*
3. *If X_i and X_j both contain a vertex v, then all nodes X_z of the tree in the path between X_i and X_j also contain v.*

The width *of a tree decomposition is the size of its largest set X_i minus one. The* treewidth *$tw(G)$ of a graph G is the minimum width among all possible tree decompositions of G.*

Proposition 12.9. *If the reliability polynomial defined in Definition 12.3 can be computed in polynomial time, then the following problems can be computed in time polynomial in the number of edges:*

1. *the number of connected spanning subgraphs;*
2. *the Banzhaf indices of edges.*

Proof. We deal with each case separately.

1. By definition, N_i is the number of connected spanning subgraphs with i edges. The value N_i is also present as a coefficient in the reliability polynomial. If all coefficients N_i are computable in polynomial time, then the total number of connected spanning subgraphs $\sum_{i=0}^{m} N_i$ is computable in polynomial time.
2. We know that $\eta_i(G) = 2\omega_i(G) - \omega(G)$ (see Lemma 3.27) where $\omega(G)$ is equal to the total number of winning coalitions and $\omega_i(G)$ is the number of winning coalitions including player i. Consider the graph G where the probability of edge i being operational is set to 1 whereas the probability of other edges being operational is set to 0.5. Then the reliability of the graph being connected is equal to the ratio of the number of connected spanning subgraphs that include edge i to 2^{M-1}, the total number of subgraphs that include i. Therefore, $\omega_i(v)$ the number of connected spanning subgraphs including edge i can be computed in polynomial time too. $\square$

Corollary 12.10. *Banzhaf indices of edges can be computed in polynomial time for graphs with bounded treewidth.*

Proof. This follows from the polynomial time algorithm to compute the reliability of a graph with treewidth k for some fixed k [7]. □

Definition 12.11. *Let $G = (V, E)$ be a graph with source s and sink t. Then G is a* series-parallel graph *if it may be reduced to K_2 by a sequence of the following operations:*

1. *replacement of a pair of parallel edges by a single edge that connects their common endpoints;*
2. *replacement of a pair of edges incident to a vertex of degree 2 other than s or t by a single edge so that 2 degree vertices get removed.*

Graphs with bounded treewidth can be recognized in polynomial time [6]. Series-parallel graphs are well-known classes of graphs with constant treewidth. Other graph classes with bounded treewidth are cactus graphs and outer-planar graphs. We see that whereas computing Banzhaf values of edges in general SCGs is NP-hard, important graph classes can be recognized and their Banzhaf values computed in polynomial time.

When edges have special properties, their power indices may be easier to compute. We define a *bridge* in a connected graph to be an edge whose removal results in the graph being disconnected. A graph class is *hereditary* if for every graph in the class, every subgraph is also in the class.

Proposition 12.12. *If graph G belongs to a hereditary graph class, for which the reliability polynomial of a graph can be computed in polynomial time, then the Shapley-Shubik index of a bridge can be computed in time polynomial in the total number of edges.*

Proof. Let $G = (V, E)$ be a graph where edge k is a bridge which connects two components $A = (V_A, E_A)$ and $B = (V_B, E_B)$. Then $|E| = |E_A| + |E_B| + 1$. If the reliability polynomial of G can be computed in polynomial time, then the reliability polynomial for each of the components A and B can be computed. Denote by $N_i(A)$ and $N_i(B)$ the number of connected spanning subgraphs in A and B respectively. These values are also the coefficients in the reliability polynomial of A and B respectively. Then the Shapley-Shubik index of player k is:

$$\phi_k(G) = \frac{\sum_{i=|V_A|-1}^{|E_A|} \sum_{j=|V_B|-1}^{|E_B|} N_i(A)N_j(B)(i+j)!(|E_A|+|E_B|-i-j)!}{|E|!}.$$

□

Our next result is that if the reliability of a simple graph can be computed then the Banzhaf indices of the corresponding multigraph can be computed. A naive approach would be to compute the Banzhaf values of each edge in a simple graph and then, for the corresponding parallel edges in the multigraph, divide the Banzhaf value of the overall edge by the number of parallel edges. However, as the following example shows, this approach is incorrect:

Example 12.13. Let $G = (V, E, s)$ be the multigraph in Figure 12.1. Then, $\eta_{4_1}(v_G) = 10$, $\eta_{1_1}(v_G) = 14$, and $\eta_2(v_G) = \eta_3(v_G) = 28$. Therefore $\beta_{4_1}(v_G) = \frac{10}{3\times10+2\times14+28+28} = \frac{5}{57}$. Moreover, $\beta_{1_1}(v_G) = \frac{7}{57}$ and $\beta_2(v_G) = \beta_3(v_G) = \frac{14}{57}$. If we examine the underlying graph of G' in Figure 12.1, then $\eta_4(v'_G) = 4$ and $\eta_1(v'_G) = \eta_2(v'_G) = \eta_3(v'_G) = 2$ giving $\beta_4(G') = 2/5$ and $\beta_i(G') = 1/5$ for $i = 1, 2, 3$. Therefore, the Banzhaf values of edges in the underlying graph do not seem to give any direct way of computing the Banzhaf values in the multigraph.

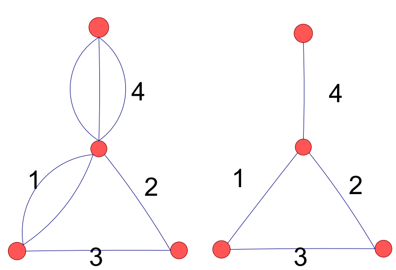

Fig. 12.1. Multigraph and its underlying graph

Lemma 12.14. *If there is an algorithm to compute the reliability of the underlying simple graph, then the algorithm can be used to compute the reliability of the corresponding multigraph.*

Proof. Let $G = (V, E, s)$ be a multigraph in which there are s_i parallel edges $i_1, \ldots, i_{s_i}$ corresponding to edge i. Let p_{i_j} be the probability that the jth parallel edge of edge i is operational. In that case $Rel(G, p)$ is equal to $Rel(G', p')$, where G' is the corresponding simple graph of G and the probability p_i that edge i is operational is $1 - \prod_{j=1}^{s_i}(1 - p_{i_j})$. □

We now prove in Proposition 12.15 that if there is an algorithm to compute the reliability of the underlying simple graph G, then it can be used to compute the Banzhaf indices of the edges in the corresponding multigraph of G. It would appear that the proposition follows directly from Lemma 12.14 and Proposition 12.9. However, one needs to be careful that the reliability computed is the reliability of the overall graph. Example 12.13 shows that computing the Banzhaf values of the edges in the underlying simple graph does not directly provide the Banzhaf values of the parallel edges in the corresponding graph.

Proposition 12.15. *For a multigraph G and edge i, let G' be the multigraph where all the other edges parallel to edge i are deleted. Then if the reliability of G' can be computed in polynomial time, then the Banzhaf value of edge i in G can be computed directly by analysing G'.*

Proof. Recall that G is a multigraph with a total of M edges. Given an algorithm to compute the reliability of G', we provide an algorithm to compute the Banzhaf values of the parallel edges of edge i in G. For graph G', set the operational probabilities of all edges to 0.5 except i which has an operation probability of $1 - 0.5^{s_i}$. and compute the overall reliability $r(G')$ of the graph. Then, by Lemma 12.14, $\omega(G)$ is $2^M r(G')$.

Now for G', set the operational probabilities of all edges to 0.5 except i which has an operation probability of 1. Let the reliability of G' with the new probabilities be $r'(G')$. We see that $r'(G')$ is equal to $\omega_i(G')/2^{M-s_i}$. Then $\omega_i(G)$ =

$2^{s_i-1}\omega_i(G') = 2^{M-1}r'(G')$. The Banzhaf value of i is then $2\omega_i(G) - \omega(G)$. A similar approach gives Banzhaf values of the other edges, from which all the Banzhaf indices can be computed. □

12.6 Other power indices

We also consider the complexity of computing the Holler indices and Deegan-Packel indices and find that they can be computed in polynomial time.

Proposition 12.16. *For SCGs corresponding to multigraphs, Holler indices and Deegan-Packel indices can be computed in polynomial time.*

Proof. We use the fact that the number of trees in a multigraph can be computed in polynomial time, which follows from *Kirchhoff's matrix tree theorem* [99]. Given a connected graph G with n vertices, let $\lambda_1, \lambda_2, ..., \lambda_{n-1}$ be the non-zero eigenvalues of the Laplacian matrix of G (the Laplacian matrix is the difference of the degree matrix and the adjacency matrix of the graph). Kirchhoff proved that the number of spanning trees of G is equal to any cofactor of the Laplacian matrix of G [99]: $t(G) = \frac{1}{n}\lambda_1\lambda_2\cdots\lambda_{n-1}$. So now that we have a polynomial-time method to compute the number of spanning trees $t(G)$ of graph G, we claim this is sufficient to compute the Holler values of the edges. If an edge i is a bridge, then it is present in every spanning tree and its Holler value is simply the total number of spanning trees. If i is not a bridge then $|M_i| = t(G) - t(G \setminus e)$. Moreover, since the size of every minimal winning coalition is the same, namely $(n-1)$, the Holler indices and Deegan-Packel indices coincide for an SCG. □

12.7 Conclusion

This chapter examined fairness-based cooperative game solutions of SCGs, for allocating resources to edges. We looked at the exact computation of power indices. In [22], an optimal randomized algorithm to compute Banzhaf indices and Shapley-Shubik indices with the required confidence interval and accuracy is presented. Since the analysis in [22] is not limited to a specific representation of a coalitional game, it can be used to approximate Banzhaf indices and Shapley-Shubik indices in SCGs.

The results of the chapter are summarized in Table 12.1. This framework can be extended to give an ordering on the importance of nodes in the graph [40]. To convert a resource allocation to edges to one on nodes, the payoff for an edge is divided equally between its two adjacent nodes. The total payoff of a node is the sum of the payoffs it gets from all its adjacent edges. This gives a way to quantify and compare the centrality or connecting role of each node. It will be interesting to understand the properties of such orderings, especially for unique cooperative solution concepts such as the nucleolus, Shapley-Shubik and Banzhaf indices.

The complexity of computing the Shapley-Shubik index for an SCG with a graph of bounded treewidth is open. If this problem is NP-hard, it will answer the question posed in the conclusion of [22] on whether there are any domains where computing one of the Banzhaf index and Shapley-Shubik index is easy, whereas computing the other is hard.

Table 12.1. Complexity of SCGs

Problem	Input	Complexity
SCG-BANZHAF-VALUE	Simple, bipartite, planar graph	#P-complete
SCG-BANZHAF-INDEX	Simple graph	?
SCG-BANZHAF-(VALUE/INDEX)	Multigraph with bounded treewidth	P
SCG-SS	Multigraph	#P-complete
SCG-SS	Multigraph with bounded treewidth	?
SCG-H-(VALUE/INDEX)	Multigraph	P
SCG-DP-(VALUE/INDEX)	Multigraph	P

13

Nucleolus of spanning connectivity games

The earth to be spann'd, connected by network,
The races, neighbors, to marry and be given in marriage,
The oceans to be cross'd, the distant brought near,
The lands to be welded together.
- Walt Whitman (Passage to India)

Two hold a garment; both claim it all. Then the one is awarded half, the other half. Two hold a garment; one claims it all, the other claims half. Then the one is awarded 3/4, the other 1/4.
- Talmud

Abstract We consider the least core imputations and the nucleolus of the spanning connectivity game. For any least core imputation, we refer to the *value* of the spanning connectivity game as the payoff of any coalition with the worst(minimum) excess. We show that the value is equal to the reciprocal of the *strength* of the underlying graph.

We efficiently compute a unique partition of the edges of the graph, called the prime-partition, and find the set of coalitions which always get the worst excess for every least core imputation. We define a partial order on the elements of the prime-partition which allows us to compute the nucleolus in polynomial time.

We also consider the problem of maximizing the probability of hitting a strategically chosen *hidden network* by placing a wiretap on a single link of a communication network. This can be seen as a two-player win-lose (zero-sum) game that

we call the *wiretap game*. The nucleolus turns out be the unique maxmin strategy which satisfies certain desirable properties.

13.1 Introduction

In the previous chapter, it was seen that computing Banzhaf values and Shapley-Shubik indices of an SCG are #P-complete. In this chapter, we will outline a polynomial time algorithm to compute the nucleolus of the SCG. The analysis is restricted to unweighted simple graphs. However, all the results extend to multi-graphs. It is easy to see that any preimputation of a SCG is also an imputation since there can be no edge e such that $v(e) = 1$ unless it is the only edge in the graph.

For any least core imputation, we refer to the *value* of the SCG as the payoff of any coalition with the worst excess. Therefore the value of the SCG G is equal to $1 - \epsilon_1(G)$ where $-\epsilon_1$ is the worst excess of a least core imputation of the SCG. We show that the value of the SCG is equal to the reciprocal of the *strength* of the underlying graph, a concept introduced by Gusfield [103].

We efficiently compute a unique partition of the edges of the graph, called the *prime-partition*. We find the set of coalitions which get the worst excess for any least core imputation. Using these special coalitions, which we call *omni-connected-spanning-subgraphs*, we define a partial order on the elements of the prime-partition. Our definition in terms of omni-connected-spanning-subgraphs is central to proving our results.

From the partial order, we obtain a linear number of simple two-variable inequalities that define the least-core-polytope. In contrast, the natural description of the least-core-polytope is as the solutions to a linear program with exponentially many constraints. Our definition of the partial order allows us to find all least core imputations that minimize the number of coalitions with the worst excess.

Among these imputations, we efficiently compute the *unique* least core imputation that maximizes the second worst excess. Hence, this imputation is the nucleolus of the game. The nucleolus turns out be a highly desirable maxmin strategy of the wiretap game (which is defined in Section 13.8).

13.2 Related work

The strength of an unweighted graph, which has a central role in our work, is also called the edge-toughness, and relates to the classical work of Nash-Williams [157] and Tutte [207]. Cunningham [53] generalized the concept of strength to edge-weighted graphs and proposed a strongly polynomial-time algorithm to compute it. Computing the strength of a graph is a special type of ratio optimization in the field of submodular function minimization [93]. Cunningham used the strength of a graph to address two different one-player optimization problems: the optimal attack and reinforcement of a network. The prime-partition we use is a truncated version of the principal-partition, first introduced by Narayanan [156] and Tomizawa [205]. The principal-partition was used in an extension of Cunningham's work to an online setting [172].

In many cases the nucleolus is hard to compute. The computational complexity of computing the nucleolus has attracted much attention [126], with both negative results [68, 73, 59], and positive results [102, 70, 125, 198].

13.3 Least core of SCGs

Denote the set of spanning subgraphs of G by $\mathcal{S}$. For a graph G, the value of the SCG $val(G)$ is defined by the least-core payoff of any coalition with the worst excess. Thus $val(G) = 1 - \epsilon_1(G)$. It is easy to see that least core payoffs are the solutions $\{x \in I(E) \mid \sum_{e \in S} x_e \geq val(G) \text{ for all } S \in \mathcal{S}\}$ to the following linear program, which has the optimal value $val(G)$.

$$\begin{aligned} &\max z \\ &\text{s.t.} \quad \textstyle\sum_{e \in S} x_e \geq z \text{ for all } S \in \mathcal{S}\,, \\ &\qquad\; x \in I(G)\,. \end{aligned} \tag{13.1}$$

The following simple observation shows the importance of minimum connected spanning graphs in the analysis of the SCG. We denote by G^x the edge-weighted graph comprising the graph G with edge weights $x(e)$ for all $e \in E$. Let $w^*(x)$ be the weight of a minimum connected spanning graph of G^x.

Fact 1 *The set of coalitions with the worst excess for imputation x is*

$$\{S \in \mathcal{S} \mid \sum_{e \in S} x_e = w^*(x)\} .$$

Proposition 13.1. *An element of the least core of an SCG can be found in polynomial time.*

Proof. The size of the linear program (13.1) is exponential in the size of the graph G, with an inequality for every subset of edges. However, this linear program can be solved using the ellipsoid method and a separation oracle, which verifies in polynomial time whether a solution is feasible or returns a violated constraint [190]. For a candidate solution $x = (x_1, \ldots, x_{|E|})$, we find in polynomial time the minimum spanning tree T of the graph G^x. Use *Kruskal's algorithm* to compute the minimum spanning tree T of graph G^x. If we have $x(E(T)) - 1 \geq -\epsilon$, then x is feasible. Otherwise, the constraint $e(x, T) \geq -\epsilon$ is violated. □

The same separation oracle idea can be used to prove the following proposition:

Proposition 13.2. *For a SCG, solving SUPERIMP and CoS is in P.*

Proof. Assume that the graph G has no bridges, because if there are bridges, then the bridge acts as a vetoer and the core is non-empty. We consider a superimputation $(x_1, \ldots, x_{|E|})$ such that $x(E) = 1 + \triangle$. For a candidate solution $x = (x_1, \ldots, x_{|E|})$, we find in polynomial time the minimum spanning tree T of the graph G^x. If $x(T) \geq 1$, then $x(S) \geq 1$ for all $S \subset E$ and x is a superimputation. If $x(T) < 1$, then x is not a superimputation.

The size of the linear program (11.4) is exponential in the size of the graph G, with an inequality for every subset of edges. Again, this linear program can be solved using the ellipsoid method and a separation oracle, which verifies in polynomial time whether a solution is feasible or returns a violated constraint [190]. We see that our solution to SUPERIMP for an SCG provides a separation oracle to solve CoS for the same SCG. □

We will see in this chapter that not only is there a combinatorial method to compute a least core imputation of a SCG in polynomial time but we also present a way to compute the nucleolus in polynomial time.

13.4 Cut-rate

In this section, we show that for any least core imputation of the SCG, the payoff of any coalition with the worst excess is equal to the *cut-rate* of the graph (Theorem 13.13). The cut-rate which is a property of a graph will be defined later in Definition 13.5. The section does not present algorithmic results but the insights and tools from this section will be used in the latter sections to devise a combinatorial algorithm to compute the nucleolus of the SCG.

We start with the basic notations and definitions. From here on we fix a connected graph $G = (V, E)$. Unless mentioned explicitly otherwise, any implicit reference to a graph is to G and α is an edge-imputation, which is a probability distribution on the edges E. For ease, we often refer to the weighted graph G^α simply by α, where this usage is unambiguous. For a subgraph H of G, we denote by $\alpha(H)$ the sum $\sum_{e \in E(H)} \alpha(e)$, where $E(H)$ is the edge set of H.

Definition 13.3. *For every edge-imputation α, we denote its distinct weights by $x_1^\alpha > \ldots > x_m^\alpha \geq 0$ and define $\mathcal{E}(\alpha) = \{E_1^\alpha, \ldots, E_m^\alpha\}$ such that $E_i^\alpha = \{e \in E \mid \alpha(e) = x_i^\alpha\}$ for $i = 1, \ldots, m$.*

Our initial goal is to characterize those partitions $\mathcal{E}(\alpha)$ that can arise from least-core-imputations α. We start with the following simple setting. Assume that the prospective imputation is restricted to α such that $|\mathcal{E}(\alpha)| = 2$, and $x_2^\alpha = 0$. Thus, the imputation's only freedom is the choice of the set E_1^α. This is done as a warm up exercise to highlight the importance of the *cut-rate* and minimum spanning subgraphs in our analysis later on. In general an imputation may contain distinct payoff for all players.

By Fact 1, a coalition with the worst excess for α is a minimum connected spanning subgraph H of α. So in order to maximize the worst excess, E_1^α should be chosen so as to maximize $\alpha(H)$. How can such an E_1^α be found? To answer, we relate the weight of a minimum connected spanning subgraph H of α to E_1^α.

To determine $\alpha(H)$, we may assume about H that for every connected component C of $(V, E \setminus E_1^\alpha)$ we have $E(H)$ contains $E(C)$, since $\alpha(e) = 0$ for every $e \in E(C)$. We can also assume that $|E_1^\alpha \cap E(H)|$ is the number of connected components in $(V, E \setminus E_1^\alpha)$ minus 1, since this is the minimum number of edges in $E(H)$ that a connected spanning subgraph may have. To formalize this we use the following notation.

Definition 13.4. *Let $E' \subseteq E$. We set $C_G(E')$, to be the number of connected components in the graph $G \setminus E'$, where $G \setminus E'$ is a shorthand for $(V, E \setminus E')$. If $E' = \emptyset$ we just write C_G.*

Using the above notation, a connected spanning subgraph H is a minimum connected spanning subgraph of α if $|H \cap E_1^\alpha| = C_G(E_1^\alpha) - C_G = C_G(E_1^\alpha) - 1$. Now we can compute $\alpha(H)$. By definition, $x_1^\alpha = \frac{1}{|E_1^\alpha|}$ and $x_2^\alpha = 0$ and therefore

$$\alpha(H) = \frac{C_G(E_1^\alpha) - C_G}{|E_1^\alpha|}.$$

We call this ratio that determines $\alpha(H)$ the cut-rate of E_1^α. Note that it uniquely determines the weight of a minimum connected spanning subgraph of α.

Definition 13.5. *Let $E' \subseteq E$. The* cut-rate *of E' in G is denoted by $cr_G(E')$ and defined as follows.*

$$cr_G(E') := \begin{cases} \frac{C_G(E') - C_G}{|E'|} & \text{if } |V| > 1 \text{ and } |E'| > 0\,, \\ 0 & \text{otherwise}\,. \end{cases} \tag{13.2}$$

We write $cr(E')$, except to make a point of referring to a different graph.

Thus, when $|\mathcal{E}(\alpha)| = 2$ and $x_2^\alpha = 0$, a best choice of E_1^α is one for which $cr(E_1^\alpha)$ is maximum. Since E is finite, an E_1^α that maximizes $cr(E_1^\alpha)$ exists. If we can find such a set E_1^α, we distribute the payoff 1 uniformly over E_1^α to get the best possible worst excess among homogeneous imputations.

Definition 13.6. *The* cut-rate *of G is defined as* $opt := \max_{E' \subseteq E} cr(E')$.

By *opt*, we always refer to the cut-rate of the graph G. In case we refer to the cut-rate of some other graph, we add the name of the graph as a subscript. The value *opt* is a well known and studied attribute of a graph. It is equal to the reciprocal of the strength of a graph, as defined by Gusfield [103] and named by Cunningham [53]. There exists a combinatorial algorithm for computing the strength, and hence *opt*, that runs in time polynomial in the *size* of the graph, by which we always mean $|V| + |E|$.

We generalize the above technique to the case that α is not restricted. Assume again that H is a minimum connected spanning subgraph of α. Intuitively, even if α has more than 2 distinct weights we would expect $|E_1^\alpha \cap E(H)|$ to be as small as possible, i.e., $C_G(E_1^\alpha) - C_G$. We would also expect $|(E_1^\alpha \cup E_2^\alpha) \cap E(H)|$ to be as small as possible, i.e., $C_G(E_1^\alpha \cup E_2^\alpha) - C_G$. If these both hold then $|E_2^\alpha \cap E(H)| = C_G(E_1^\alpha \cup E_2^\alpha) - C_G(E_1^\alpha)$, which is the increase in the number of components we get by removing the edges of E_2^α from $G \setminus E_1^\alpha$. Thus, the total weight contributed to H by edges in $E(H) \cap E(E_2^\alpha)$ is $x_2^\alpha(C_G(E_1^\alpha \cup E_2^\alpha) - C_G(E_1^\alpha))$. Now, unlike the previous case, we do not know x_2^α. However, this is not a problem since, as we shall see, we are interested in the ratio

$$\frac{\alpha(E(H) \cap E_2^\alpha)}{\alpha(E_2^\alpha)} = \frac{C_G(E_1^\alpha \cup E_2^\alpha) - C_G(E_1^\alpha)}{|E_2^\alpha|} .$$

We use the following notation to express this and its extension to more weights.

Definition 13.7. *For* $\ell = 1, \ldots, |\mathcal{E}(\alpha)|$ *we set*

$$cr_\ell^\alpha = \frac{C_G(\cup_{i=1}^{\ell} E_i^\alpha) - C_G(\cup_{i=1}^{\ell-1} E_i^\alpha)}{|E_\ell^\alpha|}.$$

The intuition above indeed holds, as stated in the following proposition.

Proposition 13.8. *Let H be a minimum connected spanning subgraph of α. Then $|E(H) \cap E_\ell^\alpha| = |E_\ell^\alpha| cr_\ell^\alpha$ for every ℓ such that $x_\ell^\alpha > 0$.*

Proof. Let H be a minimum connected spanning subgraph of α. And let t be the maximum such that $x_t^\alpha > 0$. We next show that $|E(H) \cap E_i^\alpha| = |E_i^\alpha| cr_i^\alpha$ for $i = 1, \ldots, t$.

Assume for the sake of contradiction that this is not so. Let k be minimal such that $|E(H) \cap E_k^\alpha| \neq |E_k^\alpha| cr_k^\alpha$. By the minimality of k we have

$$|E(H) \cap (\cup_{i=1}^{k-1} E_i^\alpha)| = \sum_{i=1}^{k-1} |E_i^\alpha| cr_i^\alpha. \tag{13.3}$$

Set $E' = \cup_{i=1}^{k} E_i^k$. By the definition of cut-rate the number of connected components in $G \setminus E'$ is

$$C_G(E') = 1 + \sum_{i=1}^{k} |E_i^\alpha| cr_i^\alpha. \tag{13.4}$$

Thus $|E(H) \cap E'|$ is at least $\sum_{i=1}^{k} |E_i^\alpha| cr_i^\alpha$ and therefore by (13.3) we have $|E(H) \cap E_k^\alpha| \geq |E_k^\alpha| cr_k^\alpha$.

Assume $|E(H) \cap E_k^\alpha| > |E_k^\alpha| cr_k^\alpha$. Then, by (13.3), we have

$$|E(H) \cap E'| > \sum_{i=1}^{k} |E_i^\alpha| cr_i^\alpha. \tag{13.5}$$

We show next that this implies that there exists a connected spanning subgraph whose weight by α is strictly less than $\alpha(H)$ in contradiction to H being a minimum connected spanning subgraph. Set $s = C_G(E')$ and let $C_1, \ldots, C_s$ be the connected components of $G \setminus E'$. Now as H is a minimum connected spanning subgraph the set of edges in $E(H) \cap E'$ does not have a cycle, otherwise we could have removed one of them to get a connected spanning subgraph with strictly less weight. Thus the number of connected components of $E(H) \setminus E'$ is $1 + |E \cap E(H)|$. Set $r = |E' \cap E(H)|$ and let $H_1, \ldots, H_r$ be the connected components of $H \setminus E'$.

Note that for each $i \in \{1, \dots, r\}$ there exists a unique $j \in \{1, \dots, s\}$ such that $E(H_i) \subseteq E(C_j)$. For each $j \in \{1, \dots, s\}$ set I_j to be the set of all $i \in \{1, \dots, r\}$ such that $E(H_i) \subseteq E(C_j)$. By (13.4) and (13.5) we have $s < r$ and therefore by the pigeon-hole principle there exists $j \in \{1, \dots, r\}$ such that $|I_j| > 1$. Since C_j is a connected component and H a connected spanning subgraph there exist $x, y \in I_j$ and $e = \{u, v\} \in E(C_j) \setminus \cup_{i=1}^{|I_j|} E(H_i)$ such that $u \in V(H_x)$ and $v \in V(H_y)$. Again because H is a connected spanning subgraph there is a path in H between u and v this path contains edges not in $E(C_j)$ because u, v are in different connected components of $H \setminus E'$. Thus this path contains an edge $e' \in E'$ because only edges from E' connect the vertices of C_j to the rest of the graph. Consequently $(H \setminus \{e'\}) \cup \{e\}$ is a connected spanning subgraph. Since $e \notin E'$ we have $\alpha(e) < \alpha(e')$ and consequently $\alpha(H) > \alpha((H \setminus \{e'\}) \cup \{e\})$. □

Using Proposition 13.8 we can relate the weight of a minimum connected spanning subgraph of α to the sets of $\mathcal{E}(\alpha)$. This relationship also characterizes the least-core-imputations, which are the edge-imputations whose minimum connected spanning subgraph weight is the maximum possible. The characterization is stated in Theorem 13.13. However, before we prove Theorem 13.13, we need the help of useful facts and propositions.

Fact 2 *Let H be a minimum connected spanning subgraph of α and $m = \mathcal{E}(\alpha)$ then $\alpha(H) = \sum_{i=1}^{m} x_i^\alpha |E_i^\alpha| cr_i^\alpha$ and for each $i = 1, \dots, m$ if $cr_i^\alpha < 1$ then there exists $e \in E_i^\alpha \setminus E(H)$.*

Proof. By definition $|E(H)| = \sum_{i=1}^{m} |E(H) \cap E_i^\alpha|$. Therefore $\alpha(H) = \sum_{i=1}^{m} x_i^\alpha |E(H) \cap E_i^\alpha|$. By applying Proposition 13.8 we get that $\alpha(H) = \sum_{i=1}^{m} x_i^\alpha |E_i^\alpha| cr_i^\alpha$.

Fix $i \in \{1, \dots, m\}$. By Proposition 13.8 we have $|E(H) \cap E_i^\alpha| = |E_i^\alpha| cr_i^\alpha$ and hence if $cr_i^\alpha < 1$ then $|E(H) \cap E_i^\alpha| < |E_i^\alpha|$ and therefore $E_i^\alpha \setminus E(H)$ is not empty. □

Fact 3 *Let $E_1, \dots, E_s \subseteq E$ be such that $E_i \cap E_j = \emptyset$ for every distinct $i, j \in \{1, \dots, s\}$. For $\ell = 1, \dots, s$ let r_ℓ be the cut-rate of E_ℓ in $G \setminus \cup_{i=1}^{\ell-1} E_\ell$. Assume that $r_\ell \geq y$ ($r_\ell \leq y$) for each $\ell = 1, \dots, s$. Then if there exists $i \in \{1, \dots, s\}$ such that $r_i > y$ ($r_i < y$) we have $cr(\cup_{i=1}^{s} E_i) > y$ ($r < y$) and otherwise $cr(\cup_{i=1}^{s} E_i) = y$.*

Proof. By the definition of cut-rate $C_G(\cup_{i=1}^s E_i) = C_G + \sum_{i=1}^s r_i|E_i|$ and hence

$$cr(\cup_{i=1}^s E_i) = \frac{(C_G + \sum_{i=1}^s x_i|E_i|) - C_G}{\sum_{j=1}^s |E_j|} \geq \frac{\sum_{i=1}^s y|E_i|}{\sum_{j=1}^s |E_j|} = y.$$

Note that the above inequality is strict unless $r_i = y$ for $i = 1, \dots, s$. The proof for the case that $r_i \leq y$ for $i = 1, \dots, s$, is the same. $\square$

Lemma 13.9. *If we have $E_1, E_2 \subseteq E$ and $E_1 \cap E_2 = \emptyset$ then*

$$C_G(E_1 \cup E_2) \geq C_G(E_1) + C_G(E_2) - C_G \tag{13.6}$$

Proof. We prove the statement by induction. Where there is no ambiguity, we will sometimes refer to $C_G(E_1)$ also as the components of the graph G when E_1 is deleted from G.. If $C_G(E_1) + C_G(E_2) = 2C_G$ then (13.6) holds. Assume as an inductive hypothesis, that (13.6) holds when $C_G(E_1) + C_G(E_2) = k > 2C_G$. We show that (13.6) holds for $C_G(E_1) + C_G(E_2) = k + 1$. Suppose E_1, E_2 are such that $C_G(E_1) + C_G(E_2) = k + 1$. Without loss of generality, assume $C_G(E_1) > C_G$. Therefore there is an edge $e \in E_1$ between two different elements of $C_G(E_1)$. This means that $C_G(E_1 \setminus e) = C_G(E_1) - 1$. This implies that $C_G(E_1 \setminus e) + C_G(E_2) = k$, so by the inductive hypothesis, we have $C_G((E_1 \setminus e) \cup E_2) \geq k - C_G$. Now notice that e is in a single element of $C_G((E_1 \setminus e) \cup E_2)$ and because we chose e so that it goes between two different elements of $C_G(E_1)$, removing it from the graph will add a component, giving (13.6).

Definition 13.10. *A minimal set $E' \subseteq E$ such that $cr(E') = opt$ is a* prime-set *of G.*

Proposition 13.11. *For $E', E'' \subset E$ such that $cr(E') = cr(E'') = opt$ the following holds:*

1. *$opt_{G \setminus E'} \leq opt$.*
2. *If $E'' \neq E'$ then $cr_{G \setminus E'}(E'' \setminus E') = opt$.*
3. *If $E'' \cap E' \neq \emptyset$ then $cr(E'' \cap E') = opt$.*
4. *If $E'' \setminus E' \neq \emptyset$ then $opt_{G \setminus E'} = opt$.*

5. If E' is a prime-set then either $E' \subseteq E''$ or $E' \cap E'' = \emptyset$.

Proof. Note that $opt = 0$ only if $E = \emptyset$ and therefore in this case the proposition trivially holds. Assume that $opt > 0$. Hence by the definition of cut-rate we have $E', E'' \neq \emptyset$. We shall also assume that $E' \neq E''$ since otherwise the last four items hold trivially. We next prove the first item.

Let $E^* \subseteq E \setminus E'$ be such that $cr_{G \setminus E'}(E^*) = opt_{G \setminus E'}$. By definition such a set exists. Observe that $cr_{G \setminus E'}(E^*) \leq opt$ because otherwise since $cr(E') = opt$ by Fact 3, we have $cr(E' \cup E^*) > opt$, which is a contradiction to the maximality of opt. Note that if $cr(E') < opt$, then it is not necessary at that $opt_{G \setminus E'} \leq opt$.

We now prove the second and third items. According to the first item $cr_{G \setminus E'}(E'' \setminus E') \leq opt$ and by definition $cr(E'' \cap E') \leq opt$. In the following, (13.7) holds because both sides of the equation count the number of connected components in the graph $G \setminus (E' \cup E'')$.

$$C_G(E' \cup E'') = C_{G \setminus (E' \cap E'')}(\,(E' \setminus (E' \cap E'')) \,\cup\, (E'' \setminus (E' \cap E''))\,) \tag{13.7}$$

$$\geq C_{G \setminus E' \cap E''}(E' \setminus (E' \cap E'')) + C_{G \setminus E' \cap E''}(E'' \setminus (E' \cap E'')) - C_G(E' \cap E'')) \tag{13.8}$$

$$= C_G(E') + C_G(E'') - C_G(E' \cap E'') \tag{13.9}$$

The inequality (13.8) follows by applying Lemma 13.9 to the right-hand side of (13.7). The equality (13.9) is true by the same logic as (13.7), and so we have

$$C_G(E' \cup E'') \geq C_G(E') + C_G(E'') - C_G(E' \cap E'') \tag{13.10}$$

By definition of opt_G we have (13.11). Applying (13.10) to the right-hand side of (13.11) gives (13.12). Equation (13.13) is obtained by a simple re-writing of the right-hand side of (13.12).

$$
\begin{aligned}
opt_G \geq & \frac{C_G(E' \cup E'') - C_G}{|E' \cup E''|} && (13.11)\\
\geq & \frac{C_G(E') + C_G(E'') - C_G(E' \cap E'') - C_G}{|E'| + |E''| - |E' \cap E''|} && (13.12)\\
= & \frac{(C_G(E') - C_G) + (C_G(E'') - C_G) - (C_G(E' \cap E'') - C_G)}{|E'| \qquad\qquad + |E''| \qquad\qquad - |E' \cap E''|} && (13.13)\\
\geq & opt_G && (13.14)
\end{aligned}
$$

The inequality (13.14) follows from (13.13), since, by definition we have the following

$$
\frac{C_G(E') - C_G}{|E'|} = \frac{C_G(E'') - C_G}{|E''|} = opt_G \ , \quad \frac{C_G(E' \cap E'') - C_G}{|E' \cap E''|} \leq opt_G \ .
$$

Hence, the inequalities (13.11), (13.12), (13.14), and (13.10) hold as equalities, which proves item 2 and 3.

Finally we prove the last two items. Assume $E'' \setminus E' \neq \emptyset$. By the first item $opt_{G \setminus E'} \leq opt$. By the second item $cr_{G \setminus E'}(E'' \setminus E') = opt$ and hence also $opt_{G \setminus E'} \geq opt$ and consequently $opt_{G \setminus E'} = opt$.

Assume that E' is a prime-set. If $E' \cap E'' \neq \emptyset$ then by the second item $cr(E' \cap E'') = opt$ and hence by the definition of prime-set $E' \cap E'' = E'$ which implies $E' \subseteq E''$. $\square$

If set $E' \subseteq E$ has cut-rate opt_G in G, we will refer to E' has an *optimal set*.

Observation 13.12 *Item 1 of Proposition 13.11 means that if an optimal set is deleted from the graph, then the cut-rate of the graph cannot increase. However, this is not necessarily true in case a non-optimal set is deleted from the graph. Our observation will be useful later on in Algorithm 12 where optimal sets are deleted from the graph.*

We are now in a position to prove the main theorem of the section.

Theorem 13.13. *Let H be a minimum connected spanning subgraph of α and $m = |\mathcal{E}(\alpha)|$. Then $\alpha(H) \leq opt$ and we have $\alpha(H) = opt$ if and only if*

1. $cr^{\alpha}_{\ell} = opt$ for $\ell = 1, \ldots, m-1$, and

2. *if* $cr^{\alpha}_{m} \neq opt$ *then* $x^{\alpha}_{m} = 0$.

Proof. Let β be a edge-imputation and $s = |\mathcal{E}(\beta)|$. We say β is *strong* if $cr^{\beta}_{\ell} = opt$ for $\ell = 1, \ldots, s-1$ and if $cr^{\beta}_{s} \neq opt$ then $x^{\beta}_{s} = 0$. From here on in this section H is a minimum connected spanning subgraph of α. Assume α is strong. By Fact 2

$$\alpha(H) = \sum_{\ell=1}^{|\mathcal{E}(\alpha)|} x^{\alpha}_{\ell} |E^{\alpha}_{\ell}| cr^{\alpha}_{\ell}.$$

Therefore as we have $cr^{\alpha}_{i} = opt$ for every i such that $x^{\alpha}_{i} > 0$ we conclude

$$\alpha(H) = opt \sum_{\ell=1}^{|\mathcal{E}(\alpha)|} x^{\alpha}_{\ell} |E^{\alpha}_{\ell}|.$$

Finally, since α is an edge-imputation $\sum_{\ell=1}^{|\mathcal{E}(\alpha)|} x^{\alpha}_{\ell} |E^{\alpha}_{\ell}| = 1$, we get that $\alpha(H) = opt$. Now the theorem directly follows from the subsequent lemma. □

Lemma 13.14. *Let H be a minimum connected spanning subgraph of α, then $\alpha(H) \leq opt$ and if $\alpha(H) = opt$ then α is strong.*

Intuition for the proof Lemma 13.14.

The proof of the Lemma 13.14 is by induction on s, the maximum index such that $x^{\alpha}_{s} > 0$. The basis of the induction is straightforward. The induction assumption states that for an edge-imputation β with $s-1$ distinct strictly positive weights, and minimum connected spanning subgraph H' of β we have $\beta(H') \leq opt$ and if $\beta(H') = opt$ then β is strong.

The main idea in the induction step is to show that one can shift around some of the weight of α in order to get a new edge-imputation β, such that β has exactly $s-1$ strictly positive distinct weights and $\beta(H') \geq \alpha(H)$ (or $\beta(H') > \alpha(H)$), where H' is a minimum connected spanning subgraph of β. Now since β has $s-1$ strictly positive distinct weights the induction assumption applies to it and hence $\beta(H) \leq opt$. This in turn implies that $\alpha(H) \leq opt$. Now by the above if $\alpha(H) = opt$ then also $\beta(H) = opt$ and hence by the induction assumption β is strong. With a bit of extra work this leads to α being strong.

The induction step consists of two separate cases. In the first case it is assumed that $cr_s^\alpha \leq cr(\cup_{i=1}^{s-1} E_i^\alpha)$, in the second $cr_s^\alpha > cr(\cup_{i=1}^{s-1} E_i^\alpha)$.

In the first case, by taking all the weight assigned by α to the edges of E_s^α and distributing it equally among the edges in $\cup_{i=1}^{s-1} E_i^\alpha$, one gets a new edge-imputation γ that has $s-1$ distinct strictly positive weights and $\alpha(H) \leq \gamma(H')$, where H' is a minimum connected spanning subgraph of γ. In the second case, one obtains the new edge-imputation from α in the following way. A constant amount of weight χ is reduced from each edge in $\cup_{i=1}^{s-1} E_i^\alpha$ and divide the total removed weight $\chi|\cup_{i=1}^{s-1} E_i^\alpha|$ equally among the edges of E_s^α thus getting a new edge-imputation δ where $\alpha(H) < \delta(H'')$, where H'' is a minimum connected spanning subgraph of δ. The value of χ is chosen so that δ gives the same weight to all the edges in $E_s^\alpha \cup E_{s-1}^\alpha$. Therefore the number of strictly positive weights of δ is $s-1$.

Proof of Lemma 13.14.

If $s = 1$ then by Proposition 13.8 we have $\alpha(H) = x_1^\alpha |E_1^\alpha| cr_1^\alpha = cr_1^\alpha = cr(E_1^\alpha) \leq opt$. Note that if equality holds then $cr_1^\alpha = opt$ and hence α is strong.

Let $s > 1$. The induction assumption is that for every edge-imputation β that has $s-1$ strictly positive weights, we have $\beta(H') \leq opt$ for H' that is a minimum connected spanning subgraph of β and if $\beta(H') = opt$ then β is strong.

For the inductive step we deal with two cases separately. In case **(a)** we assume that

$$cr_s^\alpha \leq \frac{\sum_{i=1}^{s-1} cr_i^\alpha |E_i^\alpha|}{\sum_{i=1}^{s-1} |E_i^\alpha|} \,. \tag{13.15}$$

In case **(b)** we assume that

$$cr_s^\alpha > \frac{\sum_{i=1}^{s-1} cr_i^\alpha |E_i^\alpha|}{\sum_{i=1}^{s-1} |E_i^\alpha|} \,. \tag{13.16}$$

Note that we chose to write the more cumbersome $\frac{\sum_{i=1}^{s-1} cr_i^\alpha |E_i^\alpha|}{\sum_{i=1}^{s-1} |E_i^\alpha|}$ instead of $cr(\cup_{i=1}^{s-1} |E_i^\alpha|)$ as this form serves our purpose better.

(a)

Set

$$\rho = \frac{x_s^\alpha |E_s^\alpha|}{\sum_{i=1}^{s-1} |E_i^\alpha|} ,$$

which is the total weight of E_s^α divided equally among all edges in $\cup_{i=1}^{s-1} E_i^\alpha$. Define $\gamma : E(G) \to \mathbb{R}$ so that $\gamma(e) = \alpha(e) + \rho$ for every $e \in \cup_{j=1}^{s-1} E_j^\alpha$ and $\gamma(e) = 0$ for every $e \in E(G) \setminus \cup_{j=1}^{s-1} E_j^\alpha$. According to this definition

$$\sum_{e \in E} \gamma(e) = \sum_{e \in E} \alpha(e) - \sum_{e \in E_s^\alpha} x_s^\alpha + \sum_{e \in \cup_{i=1}^{s-1} E_i^\alpha} \rho .$$

Since α is an edge-imputation we can replace $\sum_{e \in E(G)} \alpha(e)$ with 1. Doing so in the above equation in addition to replacing ρ by its value gives us

$$\sum_{e \in E(G)} \gamma(e) = 1 - \left(x_s^\alpha |E_s^\alpha| - \frac{x_s^\alpha |E_s^\alpha|}{\sum_{i=1}^{s-1} |E_i^\alpha|} \sum_{i=1}^{s-1} |E_i^\alpha| \right) = 1.$$

By definition γ has exactly $s - 1$ strictly positive weights and hence, γ is an edge-imputation and the induction assumption applies to γ. Let H' be a minimum connected spanning subgraph of γ. By the induction assumption $\gamma(H') \le opt$. We next show that $\alpha(H) \le \gamma(H')$ and hence $\alpha(H) \le opt$. According to the construction of γ we have $x_i^\gamma > x_j^\gamma$ if and only if $x_i^\alpha > x_j^\alpha$ for any $i, j \in \{1, \ldots, s - 1\}$ and therefore $E_i^\alpha = E_i^\gamma$ for $i = 1, \ldots, s - 1$. This in turn implies that $cr_i^\alpha = cr_i^\gamma$ for $i = 1, \ldots, s - 1$. According to Fact 2 we have

$$\gamma(H') = \sum_{i=1}^{s-1} x_i^\gamma cr_i^\gamma |E_i^\gamma| .$$

By replacing E_i^α with E_i^γ and cr_i^α with cr_i^γ and x_i^γ with $x_i^\alpha + \rho$ for $i = 1, \ldots, s - 1$ we get

$$\gamma(H') = \sum_{i=1}^{s-1} (x_i^\alpha + \rho) cr_i^\alpha |E_i^\alpha|.$$

This implies

$$\gamma(H') = \sum_{i=1}^{s} x_i^\alpha cr_i^\alpha |E_i^\alpha| - x_s^\alpha cr_s^\alpha |E_s^\alpha| + \rho \sum_{i=1}^{s-1} cr_i^\alpha |E_i^\alpha|.$$

By Fact 2, we can replace $\sum_{i=1}^{s} x_i^\alpha cr_i^\alpha |E_i^\alpha|$ by $\alpha(H)$. By also replacing ρ with its value we have

$$\gamma(H') = \alpha(H) - x_s^\alpha |E_s^\alpha| \left(cr_s^\alpha - \frac{\sum_{i=1}^{s-1} cr_i^\alpha |E_i^\alpha|}{\sum_{i=1}^{s-1} |E_i^\alpha|} \right). \tag{13.17}$$

Now by (13.15) and (13.17) we have $\alpha(H) \le \gamma(H')$.

Assume that $\alpha(H) = opt$. Since $\alpha(H) \le \gamma(H') \le opt$ we get $\gamma(H') = opt$. Thus by the induction assumption γ is strong and hence $cr_i^\gamma = opt$ for $i = 1, \dots, s-1$. Since $cr_i^\alpha = cr_i^\gamma = opt$ for $i = 1, \dots, s-1$, to conclude that α is strong. Thus, we only need to show that $cr_s^\alpha = opt$. By replacing $\alpha(H), \gamma(H'), cr_1^\alpha, \dots cr_{s-1}^\alpha$ with opt in (13.17) we get

$$cr_s^\alpha = \frac{\sum_{i=1}^{s-1} opt |E_i^\alpha|}{\sum_{i=1}^{s-1} |E_i^\alpha|} = opt.$$

(b)

Let

$$\chi = (x_{s-1}^\alpha - x_s^\alpha)(1 + \frac{|E_s^\alpha|}{\sum_{i=1}^{s-1} |E_i^\alpha|})^{-1}.$$

Let δ be such that

$$\delta(e) = \begin{cases} \alpha(e) + \chi & \text{if } e \in E_s^\alpha, \\ \alpha(e) - \chi \frac{|E_s^\alpha|}{\sum_{i=1}^{s-1} |E_i^\alpha|} & \text{if } e \in \cup_{i=1}^{s-1} E_i^\alpha, \\ 0 & \text{otherwise.} \end{cases} \tag{13.18}$$

Note that χ is such that $x_s^\alpha + \chi = x_{s-1}^\alpha - \chi \frac{|E_s^\alpha|}{\sum_{i=1}^{s-1} |E_i^\alpha|}$. Consequently, δ assigns the same weight to each edge in $E_{s-1}^\alpha \cup E_s^\alpha$ and hence δ has exactly $s-1$ strictly positive weights.

We next show that δ is an edge-imputation. By definition

$$\sum_{e \in E(G)} \delta(e) = \sum_{e \in E(G)} \alpha(e) + \sum_{e \in E_s^\alpha} \chi - \sum_{e \in \cup_{j=1}^{s-1}} \chi \frac{|E_s^\alpha|}{\sum_{i=1}^{s-1} |E_i^\alpha|}.$$

Since α is an edge-imputation we can replace $\sum_{e \in E(G)} \alpha(e)$ by 1 in the above to conclude

$$\sum_{e \in E(G)} \delta(e) = 1 + \chi \left(|E_s^\alpha| - \frac{|E_s^\alpha|}{\sum_{i=1}^{s-1} |E_i^\alpha|} \sum_{i=1}^{s-1} |E_i^\alpha| \right) = 1.$$

Note that by the choice of χ we have $\delta(e) > x_s^\alpha$ for every $e \in \cup_{i=1}^{s} E_i^\alpha$ and since $\delta(e) = 0$ for any other edge all the weights δ assigns are non-negative. Thus δ is an edge-imputation. Let H' be a minimum connected spanning subgraph of δ. Now as δ is an edge-imputation with $s-1$ strictly positive weights, by the induction assumption we have $\delta(H') \leq opt$. We conclude the claim by showing that $\alpha(H) < \delta(H')$.

By Fact 2 we have

$$\delta(H') = \sum_{i=1}^{s-1} x_i^\delta cr_i^\delta |E_i^\delta|. \tag{13.19}$$

According to the construction of δ we have $x_i^\delta > x_j^\delta$ if and only if $x_i^\alpha > x_j^\alpha$ for any $i, j \in \{1, \ldots, s-2\}$ and therefore $E_i^\alpha = E_i^\delta$ for $i = 1, \ldots, s-2$ which in turn implies that $cr_i^\alpha = cr_i^\delta$ for $i = 1, \ldots, s-2$. Thus $E_i^\alpha = E_i^\delta$ and $cr_i^\alpha = cr_i^\delta$ for $i = 1, \ldots, s-2$. Consequently by replacing $|E_{s-1}^\delta|$ with $|E_{s-1}^\alpha| + |E_s^\alpha|$ and cr_i^γ with cr_i^α for $i = 1, \ldots, s-2$ in (13.19) we get

$$\delta(H') = \sum_{i=1}^{s-2} x_i^\delta cr_i^\alpha |E_i^\alpha| + x_{s-1}^\delta cr_{s-1}^\delta (|E_{s-1}^\alpha| + |E_s^\alpha|). \tag{13.20}$$

By definition of cut-rate we have

$$cr_{s-1}^\delta = \frac{|E_{s-1}^\alpha| cr_{s-1}^\alpha + |E_{s-1}^\alpha| cr_s^\alpha}{|E_{s-1}^\alpha| + |E_s^\alpha|} \ ,$$

Plugging this in (13.20) gives us

$$\delta(H') = \sum_{i=1}^{s-2} x_i^{\delta} cr_i^{\alpha} |E_i^{\alpha}| + x_{s-1}^{\delta} cr_{s-1}^{\alpha} |E_{s-1}^{\alpha}| + x_{s-1}^{\delta} cr_s^{\alpha} |E_s^{\alpha}|.$$

Since $x_s^{\delta} = x_s^{\alpha} + \chi$ and $x_i^{\delta} = x_i^{\alpha} - \chi \frac{|E_s^{\alpha}|}{\sum_{i=1}^{s-1} |E_i^{\alpha}|}$ for $i = 1, \dots s-2$ we get

$$\delta(H') = \sum_{i=1}^{s} x_i^{\alpha} cr_i^{\alpha} |E_i^{\alpha}| + \chi |E_s^{\alpha}| \left(cr_s^{\alpha} - \frac{\sum_{i=1}^{s-1} cr_i^{\alpha} |E_i^{\alpha}|}{\sum_{i=1}^{s-1} |E_i^{\alpha}|} \right). \tag{13.21}$$

By Fact 2 we can also replace $\sum_{i=1}^{s} x_i^{\alpha} cr_i^{\alpha} |E_i^{\alpha}|$ with $\alpha(H)$ in (13.21). This together with (13.16) implies that $\delta(H') > \alpha(H)$. Note that as $opt > \delta(H')$ it can not be the case that $\alpha(H) = opt$. □

An immediate implication of Theorem 13.13 is that opt is an upper bound on the payoff of any coalition with the worst excess for any payoff. Since this payoff opt can be achieved by distributing all payoff over an edge set that has cut-rate opt, we get the following.

Corollary 13.15. *The value of the SCG is opt.*

Also, we mentioned earlier that there are polynomial time combinatorial algorithms to compute the value opt and an optimal set of a graph. Therefore, we can use such an algorithm as a subroutine to provide a polynomial time combinatorial algorithm to compute an imputation in the least core.

13.5 Prime Partition

From Theorem 13.13, we know what the value of the game is and we know a characterization of the $\mathcal{E}(\alpha)$'s for least-core-imputations α. Yet this characterization does not give us a simple way to find least-core-imputations. Resolving this is our next goal. Since the set of least-core-imputations is convex, it is easy to show that there exists a least-core-imputation β such that for every $e_1, e_2 \in E$ we have $\beta(e_1) = \beta(e_2)$ if and only if $\gamma(e_1) = \gamma(e_2)$ for every least-core-imputation γ. This implies that $\mathcal{E}(\beta)$ refines $\mathcal{E}(\gamma)$ for every least-core-imputation γ, where by "refines" we mean the following.

Definition 13.16. *Let $\mathcal{E}_1, \mathcal{E}_2$ be partitions of E. Then $\mathcal{E}_1$* refines *$\mathcal{E}_2$ if for every set $E' \in \mathcal{E}_1$ there exists a set $E'' \in \mathcal{E}_2$ such that $E' \subseteq E''$.*

Thus, there exists a partition of E that is equal to $\mathcal{E}(\beta)$ for some least-core-imputation β and refines $\mathcal{E}(\gamma)$ for every least-core-imputation γ. We call such a partition the prime-partition. It is unique since there can not be different partitions that refine each other.

Definition 13.17. *The* prime-partition *$\mathcal{P}$ is the unique partition that is equal to $\mathcal{E}(\beta)$ for some least-core-imputation β and refines $\mathcal{E}(\gamma)$ for every least-core-imputation γ.*

We now show that the prime-partition exists and can be computed in time polynomial in the size of G. This is the main result of this section.

We introduce a polynomial time algorithm that on input graph $G = (V, E)$ returns a partition of E, which afterwards we show is the prime-partition of G. The algorithm uses oracle access to a routine `PrimeSet` that given a graph returns its cut-rate and one of its prime sets. This routine runs in time polynomial in the size of G and is introduced after Algorithm 12. We first describe Algorithm 12 and the routine `PrimeSet`. Later, in Subsection 13.5.2, we will show that the output of Algorithm 12 is indeed the prime-partition.

13.5.1 Construction of the prime-partition

Algorithm 12 Prime-partition construction

Input: Graph G .

Output: Prime partition $\mathcal{P}$.

$\mathcal{P} \leftarrow \emptyset$
if $E(G) = \emptyset$ **then**
  return $\mathcal{P}$
end if
$i \leftarrow 1$
$(opt, P_i) \leftarrow$ `PrimeSet`(G)
$G_i \leftarrow G$
repeat
  $\mathcal{P} \leftarrow \mathcal{P} \cup \{P_i\}$
  $G_{i+1} \leftarrow G_i \setminus P_i$
  $i \leftarrow i + 1$
  $(c, P_i) \leftarrow$ `PrimeSet`(G_i)
until $c < opt$
if $E(G_i) \neq \emptyset$ **then**
  $\mathcal{P} \leftarrow \mathcal{P} \cup \{P_i\}$
end if
return $\mathcal{P}$

Each computation done by Algorithm 12 requires running time polynomial in $|V|$ including the calls to `PrimeSet`. Therefore, the only reason the running time of Algorithm 12 may be too long is the repeat loop. Note that, if `PrimeSet` returns an empty set, then it also sets $c = 0$. Thus, after any iteration of the repeat loop that is not the last the number of edges of G' is decreased by at least 1. Observe that `PrimeSet` returns $(0, \emptyset)$ and does not reach the repeat loop if $opt = 0$. Thus only if $opt > 0$ then the repeat loop is reached and then the above ensures that it goes through at most $|E|$ iterations. Hence the running time of Algorithm 12 is polynomial in the size of G as long as there is a polynomial time algorithm for `PrimeSet`.

We now show how to use existing algorithms to design the routine `PrimeSet` for finding a minimal optimal set. We assume that the graph is connected, in case it is not connected we run the routine separately on each connected component and return the `PrimeSet` (and value *opt* that achieves the largest *opt* among these connected components. If there is more than one, pick one arbitrarily. By Fact 3, the cut-rate of $\cup_{i=1}^{m-1} E_i$ in G is *opt* and therefore by Proposition 13.20, we have that $\mathcal{P}$ refines $\mathcal{S}_1$. For our goal, we extend the notion of cut-rate of a graph to edge weighted graphs.

Definition 13.18. *Let* $E' \subseteq E$ *and* $\omega : E \to \mathbb{R}^+$. *The* cut-rate *of* E' *in* G, ω *is denoted by* $cr_\omega(E')$ *and defined as follows.*

$$cr_\omega(E') \coloneqq \begin{cases} \frac{C_G(E') - C_G}{\omega(E')} & \text{if } |V| > 1 \text{ and } |E'| > 0\ , \\ 0 & \text{otherwise}\ . \end{cases} \tag{13.22}$$

The cut-rate of G, ω *where* $\omega : E \to \Re^+$ *is defined as*

$$opt_\omega \coloneqq \max_{E' \subseteq E} cr_\omega(E') \tag{13.23}$$

There exists strongly polynomial algorithms in [53, 206, 45] that on G, ω returns opt_ω. We shall assume from here on that opt_ω is given and omit the fact that this is done by the mentioned algorithm.

A prime-set of G is found as follows. If $E = \emptyset$ then stop and return $(0, \emptyset)$. Otherwise, set $\omega : E \to \Re^+$ so that $\omega(e) = 1$ for every $e \in E$. Note that in this case $opt = opt_\omega$ and hence we assume *opt* is known. Set $\omega' = \omega$. Next iterate e over the elements of E according to some arbitrary order and in each iteration do the following. Set $\omega'(e)$ to be 2 and if $opt_{\omega'} = opt$ then set ω to be ω' and otherwise set ω' to be ω. That is, $\omega(e)$ is changed only if $opt_{\omega'} = opt$ and otherwise remains the same. After the iterative process is over set $E' = \{e \in E \mid \omega(e) = 1\}$ and return (opt, E'). Note that the total number of operations done is polynomial in the size of G and so is the running time. To show that indeed this achieves our goal, we only need to prove that E' is a prime set of G.

Let us look at any fixed iteration over e. By definition ω is changed only if $opt_{\omega'} = opt$ and then it is set to ω'. This implies that there exists $E^* \subseteq E$ such that $cr_{\omega'}(E^*) = opt$. Now it can not be the case that $\omega'(e) = 2$ for some $e \in E^*$, since this would imply that $cr(E^*) > opt$. Consequently, $cr_\omega(E^*) = opt$. This is true for any fixed iteration and hence also for the last. Therefore, there exists $E'' \subseteq E'$ such that $cr(E'') = opt$. Finally assume for the sake of contradiction that $E'' \subset E'$. Let $e' \in E' \setminus E''$. This implies that in the iteration dedicated to e' we had $opt_{\omega'} \neq opt$. Since at this stage $\omega'(e) = 1$ for every E'' it must be that $opt_{\omega'} > opt$. Yet this can not be since the weights assigned to each edge by ω' is at least as that assigned by ω and hence at every iteration $opt_{\omega'} \leq opt$.

From here on in this section $t = |\mathcal{P}|$, where $\mathcal{P}$ is the output of Algorithm 12 on input graph G, and the elements of $\mathcal{P}$ are named as they were named by Algorithm 12, thus $\mathcal{P} = \{P_1, \ldots, P_t\}$. In addition let $E^0 = \emptyset$ and for each $k = 1, \ldots, t$ let $E^k = \cup_{i=1}^{k} P_i$ and r_k be the cut-rate of P_k in $G \setminus E^{k-1}$.

13.5.2 The output of Algorithm 12 is the prime partition

Proposition 13.19. *There exists a least-core-imputation β such that $\mathcal{E}(\beta) = \mathcal{P}$.*

Proof. Assume $r_t < opt$, i.e., the last set returned in Algorithm 12 does not have cut-rate of opt. Set $\rho = \frac{1}{\sum_{i=1}^{t}(t-i)|E_i|}$ and let $\beta : E \to \mathbb{R}$, where for each $i = 1, \ldots, t$ and $e \in P_i$ we have $\beta(e) = (t-i)\rho$. Observe that

$$\sum_{e \in E} \beta(e) = \sum_{i=1}^{t} x_i^\beta |E_i^\beta| = \sum_{i=1}^{t} \rho(t-i)|E_i^\beta| = \rho \sum_{i=1}^{t}(t-i)|E_i^\beta| = \rho\rho^{-1} = 1$$

and hence β is an edge-imputation. By definition $E_i^\beta = P_i$ for $i = 1, \ldots, t$. We next show that $r_i = opt$ for $i = 1, \ldots, t-1$. By Theorem 13.13 this implies that β is a least-core-imputation.

Algorithm 12 selects P_1 so that $r_1 = opt$. Let $k < t$ and assume that $r_j = opt$ for every $j < k$. Hence by Fact 3 we have $cr(E^{k-1}) = opt$. Consequently $r_k \leq opt$ since otherwise by Fact 3 we get that $cr(E^{k-1}) > opt$. As P_k is not the last set added to $\mathcal{P}$ by Algorithm 12, we have $r_k \geq opt$ and hence it is the case that $r_k = opt$.

Now assume that $r_t = opt$. The same argument as above works when $r_t = opt$. In this case, set $\rho = \frac{1}{\sum_{i=1}^{t}(t-i+1)|E_i|}$ and let $\beta : E \to \mathbb{R}$, where for each $i = 1, \ldots, t$ and $e \in P_i$ we have $\beta(e) = (t-i+1)\rho$. $\square$

We next show that $\mathcal{P}$ refines $\mathcal{E}(\alpha)$ for every least-core-imputation α. We start with a simple case that we use later on to prove the general result.

Proposition 13.20. *If $cr(E') = opt$ for $E' \subseteq E$ then $\mathcal{P}$ refines $\{E', E \setminus E'\}$.*

Proof. If $E' = E$ then the proposition holds trivially. Hence we only need to prove the proposition holds when $E' \subset E$. We show that $P_i \cap E' = \emptyset$ or $P_i \subseteq E'$ for every $i = 1, \ldots, t$. Let $k \in \{1, \ldots, t\}$. If $E' \setminus E^{k-1} = \emptyset$ then $P_k \cap E' = \emptyset$. Therefore we only need to deal with the case that $E' \setminus E^{k-1} \neq \emptyset$. Assume this is indeed so. By Proposition 13.19, we have $r_i = opt$ for $i = 1, \ldots, t-1$ hence by Fact 3 we get $cr(E^{k-1}) = opt$. Since also $cr(E') = opt$, by Proposition 13.11, we have $opt_{G \setminus E^{k-1}} = opt$ and $cr_{G\setminus E^{k-1}}(E' \setminus E^{k-1}) = opt$. We separate the proof into two cases the first $k = 1, \ldots, t-1$ and in the second $k = t$.

Recall that Algorithm 12 selects P_k so that it is a prime-set in $G \setminus E^{k-1}$. So now $opt_{G\setminus E^{k-1}} = opt$ and $cr_{G\setminus E^{k-1}}(P_k) = cr_{G\setminus E^{k-1}}(E' \setminus E^{k-1}) = opt$. Thus by Proposition 13.11 either $P_k \cap (E' \setminus E^{k-1}) = \emptyset$ or $P_k \subseteq (E' \setminus E^{k-1})$. If $P_k \subseteq (E' \setminus E^{k-1})$ then $P_k \subseteq E'$, and if $P_k \cap (E' \setminus E^{k-1}) = \emptyset$ then $P_k \cap E^{k-1} = \emptyset$ because $P_k \cap E^{k-1} = \emptyset$.

Assume $k = t$ and for the sake of contradiction that $E' \setminus E^{t-1} \neq \emptyset$. Since we have shown that $opt_{G\setminus E^{t-1}} = opt$ and $cr_{G\setminus E^{t-1}}(E' \setminus E^{t-1}) = opt$ it is the case that $G \setminus E^{t-1}$ has a prime-set that has cut-rate opt in $G \setminus E^{t-1}$. This subset is strictly contained in $E \setminus E^{t-1}$ Proposition 13.11 implies that every prime set in $E \setminus E^{t-1}$ is strictly contained in $E \setminus E^{t-1}$. Hence, Algorithm 12 would have found a prime-set $E^* \subset E \setminus E^{t-1}$ and added it to $\mathcal{P}$. That is $E^* \in \mathcal{P}$. Yet this can not be since by construction $\mathcal{P}$ is a partition of E. □

Proposition 13.21. *If γ is a least-core-imputation then $\mathcal{P}$ refines $\mathcal{E}(\gamma)$.*

Proof. Let $t = |\mathcal{E}(\gamma)|$. Recall that since γ is a least-core-imputation by definition for $i = 1, \ldots, t-1$ we have $cr_i^\gamma = opt$. If $t = 1$ then the only set in $\mathcal{E}(\gamma)$ is E and hence the lemma trivially holds. By Proposition 13.20 the lemma also holds when $|\mathcal{E}(\gamma)| = 2$. Assume by way of induction that proposition holds for any partition $\mathcal{S} = \{E_1, \ldots, E_{t-1}\}$ of E such that $cr_{G\setminus \cup_{i=1}^{\ell} E_i}(E_\ell) = opt$. Let $\mathcal{S}_1 = \{\cup_{i=1}^{t-1} E_i^\alpha, E_t^\alpha\}$ and $\mathcal{S}_2 = \{E_1, \ldots, E_{t-2}^\alpha, E_{t-1}^\alpha \cup E_t^\alpha\}$. Note that if $\mathcal{P}$ refines both $\mathcal{S}_1$ and $\mathcal{S}_2$ then it refines $\mathcal{S}$. By the induction assumption, $\mathcal{P}$ refines $\mathcal{S}_2$. By Fact 3 we have that $cr(\cup_{i=1}^{m-1} E_i) = opt$ and therefore $\mathcal{P}$ refines $\mathcal{S}_1$ by Proposition 13.20. □

Theorem 13.22. *The prime-partition exists and can be computed in time polynomial in the size of G.*

Proof. The theorem follows from Algorithm 12 and Proposition 13.21.

The prime-partition $\mathcal{P}$ reveals a lot about the structure of the least-core-imputations. Yet by itself $\mathcal{P}$ does not give us a simple means for generating least-core-imputations. Using the algorithm for finding $\mathcal{P}$ one can show that, depending on G, there may be a unique element in $\mathcal{P}$ whose edges are assigned 0 by every least-core-imputation.

Lemma 13.23. *$cr_G(E) \neq opt$ if and only if there exists a unique set $D \in \mathcal{P}$ such that for every least-core-imputation α and $e \in D$ we have $\alpha(e) = 0$. If D exists then it can be found in running time polynomial in the size of G.*

Proof. From here on we shall always refer to the set D in Lemma 13.23 as the degenerate set. In case it does not exist, D or $\{D\}$ should be treated as if it was the empty set.

If $cr(E) = opt$, then an edge-imputation that assigns equal weight to all edges is a least-core-imputation and so there is no degenerate set.

We now prove that if $cr(E) \neq opt$, then the degenerate set exists, it is unique, and can be found in running time polynomial in the size of G.

Assume that $cr(E) \neq opt$. By the definition of opt, this can only happen if $cr(E) < opt$. Let β be a least-core-imputation such that $\mathcal{E}(\beta) = \mathcal{P}$ and set $t = |\mathcal{P}|$. By definition, E_t^β assigns strictly positive weights to the edges in each E_i^β for every $i = 1, \ldots, t-1$. Hence, the only candidate for being the degenerate set is E_t^β. We next show that this is indeed the case.

Assume for the sake of contradiction that there exists a least-core-imputation γ that assigns strictly positive weights to the edges in E_t^β. Let $d = \min_{i \in \{1,\ldots,t-1\}}\{x_i^\beta - x_{i-1}^\beta\}/10$ and set $\delta = (1-d)\beta + d\gamma$ (the choice of 10 is arbitrary). Observe that δ has the same number of distinct weights as γ, and $\mathcal{E}(\delta) = \mathcal{E}(\beta)$ and we have $E_i^\delta = E_i^\beta$

for $i = 1, \ldots, t$. Let H be a minimum connected spanning subgraph of δ. Since δ is a convex combination of least-core-imputations it is a least-core-imputation and therefore, by Corollary 13.15, we have $\delta(H) = opt$. We next get the required contradiction by showing that $\delta(H) < opt$.

Since $\mathcal{E}(\delta)$ is a partition of E we have

$$\delta(H) = \sum_{i=1}^{t} x_i^{\delta} |E(H) \cap E_i^{\delta}|. \tag{13.24}$$

As δ is a least-core-imputation, by Theorem 13.13, we have $cr_i^{\beta} = opt$ for $i = 1, \ldots, t-1$ and hence $cr_t^{\beta} < opt$, since otherwise, by Fact 3, we have $cr(E) \geq opt$. By Proposition 13.8, we have $|E(H) \cap E_i^{\delta}| = opt|E_i^{\delta}|$ for $i = 1, \ldots, t-1$. Applying this to (13.24) we get

$$\delta(H) = x_t^{\delta} |E(H) \cap E_t^{\delta}| + opt \sum_{i=1}^{t-1} x_i^{\delta} |E_i^{\delta}| \ . \tag{13.25}$$

Now, since H is a minimum connected spanning subgraph, $|E(H) \cap E_t^{\delta}|$ is the minimum possible, which in this case is $cr_t^{\delta}|E_t^{\delta}|$. Since $cr_t^{\delta} < opt$, we get that $cr_t^{\delta}|E_t^{\delta}| < opt|E_t^{\delta}|$. Thus, by replacing $|E(H) \cap E_t^{\delta}|$ by $opt|E_t^{\delta}|$ in (13.25), we get

$$\delta(H) < opt \sum_{i=1}^{t} x_i^{\delta} |E_i^{\delta}| = opt \ ,$$

where the equality is because δ is an edge-imputation.

We now explain how to compute D. Once opt is known, one only needs to check if $\frac{|V|-1}{|E|} = opt$. If the answer is yes, then there is no degenerate set; if the answer is no, then D is the last set inserted to $\mathcal{P}$ by Algorithm 12. □

See Figure 13.1 for an example of the prime-partition and the degenerate set.

13.6 Parent-child relation

We use the prime-partition to define a special subset of the minimum connected spanning subgraphs that we call the omni-connected-spanning-subgraphs, which are useful for proving the characterization of least-core-imputations and their refinements.

Definition 13.24. *A connected spanning subgraph H is an* omni-connected-spanning-subgraph *if for every $P \in \mathcal{P} \setminus \{D\}$ we have*

$$|E(H) \cap P| = |P| \cdot opt \,.$$

Proposition 13.25. *There exists an omni-connected-spanning-subgraph.*

Proof. Let β be a least-core-imputation such that $\mathcal{E}(\beta) = \mathcal{P}$. Let H be a minimum connected spanning subgraph of β. Then by Proposition 13.8, we have that $|E(H) \cap E^{\beta}_{\ell}| = |E^{\beta}_{\ell}| cr^{\beta}_{\ell}$ for every ℓ such that $x^{\beta}_{\ell} > 0$. Since $\mathcal{E}(\beta) = \mathcal{P}$, by Algorithm 12 we know that $cr^{\beta}_{\ell} = opt$ for every ℓ such that $x^{\beta}_{\ell} > 0$. Therefore, for every $P \in \mathcal{P} \setminus \{D\}$ we have $|E(H) \cap P| = |P| \cdot opt$. By definition, H is an omni-connected-spanning-subgraph. □

The omni-connected-spanning-subgraphs are the set of the coalitions which get the worst excess for any least-core-imputation.

Proposition 13.26. *For every edge-imputation α such that $\mathcal{P}$ refines $\mathcal{E}(\alpha)$ and $\alpha(e) = 0$ for every $e \in D$ and omni-connected-spanning-subgraph H, we have $\alpha(H) = opt$.*

Proof. Let H be an omni-connected-spanning-subgraph and α such that $\mathcal{P}$ refines $\mathcal{E}(\alpha)$ and $\alpha(e) = 0$ for every $e \in D$. Thus for each $P \in \mathcal{P}$ there exists y^P such that $\alpha(e) = y^P$ for every $e \in P$. Since $\mathcal{P}$ is a partition of E, we have $\alpha(H) = \sum_{P \in \mathcal{P}} y^P |H \cap P|$ and as H is an omni-connected-spanning-subgraph also $|H \cap P| = |P| opt$ for every $P \in \mathcal{P}$. Consequently,

$$\alpha(H) = \sum_{P \in \mathcal{P}} y^P |P| opt = opt \sum_{P \in \mathcal{P}} y^P |P|.$$

Now, as α is an edge-imputation and $\sum_{P \in \mathcal{P}} y^P |P| = 1$, with the above, we get $\alpha(H) = opt$. Now, if α is least-core-imputation, by Corollary 13.15, the value of the game is opt and H is a minimum connected spanning subgraph of α. $\square$

The importance of omni-connected-spanning-subgraphs stems from the following scenario. Assume that $\mathcal{P}$ refines $\mathcal{E}(\alpha)$ and $\alpha(e) = 0$ for every $e \in D$, and let H be an omni-connected-spanning-subgraph. By Proposition 13.26, we know that $\alpha(H) = opt$. Suppose we can remove from H an edge from $E(H) \cap P$, where P is a nondegenerate element of $\mathcal{P}$, and add a new edge from another set $P' \setminus E(H)$ in order to get a new connected spanning subgraph. Assume α assigns to the edge removed strictly more weight than it assigns to the edge added. Then the new connected spanning subgraph has weight strictly less than $\alpha(H)$ and hence strictly less than opt, since $\alpha(H) = opt$ by Proposition 13.26. Consequently, α is not a least-core-imputation and we can conclude that any edge-imputation β that assigns to each edge in P strictly more weight than to the edges in P' is not a least-core-imputation. This intuition is captured by the following definition, which leads to the characterization of least-core-imputations in Theorem 13.29.

Definition 13.27. *Let $P, P' \in \mathcal{P} \setminus D$ be distinct. Then P* leads to *P' if and only if there exists an omni-connected-spanning-subgraph H with $e \in P \setminus E(H)$ and $e' \in P' \cap E(H)$ such that $(H \setminus \{e'\}) \cup \{e\}$ is a connected spanning subgraph. We denote the "leads to" relation by $\mathcal{R}$.*

Definition 13.28. *An edge-imputation α* agrees *with $\mathcal{R}$ if $\mathcal{P}$ refines $\mathcal{E}(\alpha)$ and for every $P \in \mathcal{P} \setminus D$ that is a parent of $P' \in \mathcal{P} \setminus D$ and $e \in P$, $e' \in P'$ we have $\alpha(e) \geq \alpha(e')$, and for every $e \in D$ we have $\alpha(e) = 0$.*

Theorem 13.29. *An edge-imputation α is a least-core-imputation if and only if it agrees with $\mathcal{R}$.*

Proof. Note that if $|\mathcal{P}| = 1$ then the theorem trivially holds hence we assume $|\mathcal{P}| > 1$.

Let α be a least-core-imputation. By Lemma 13.23 we have $\alpha(e) = 0$ for every $e \in D$. Assume for the sake of contradiction that α does not agree with $\mathcal{R}$. By

Definition 13.17, we have that $\mathcal{P}$ refines α and hence since α does not agree with $\mathcal{R}$ there exist $P \in \mathcal{P} \setminus \{D\}$ that leads to $P' \in \mathcal{P} \setminus \{D\}$, an omni-connected-spanning-subgraph H, $e \in E(H) \setminus P$ and $e' \in P' \cap E(H)$ such that $\alpha(e) < \alpha(e')$ and $H' = (H \setminus \{e'\}) \cup \{e\}$ is a connected spanning subgraph. Since H is an omni-connected-spanning-subgraph and α a least-core-imputation by Proposition 13.26 we have $\alpha(H) = opt$. Thus $\alpha(H') = \alpha(H) + (\alpha(e) - \alpha(e')) < opt$. This is in contradiction to Fact 13.15, which implies that $\alpha(H') \geq opt$ since α is a least-core-imputation.

Assume α is an edge-imputation that agrees with $\mathcal{R}$. We next show that this implies that α is a least-core-imputation.

Let m be the number of the strictly positive weights of α. Assume by way of contradiction that α is not a least-core-imputation. By Theorem 13.13 this can only happen if there exists $i \in \{1, \dots, m\}$ such that $cr_i^\alpha \neq opt$. Let ℓ be the smallest element in $\{1, \dots, m\}$ such that $cr_\ell^\alpha \neq opt$. Let $E' = \cup_{i=1}^{\ell} E_i^\alpha$. We show next that $cr(E') < opt$. If $\ell = 1$ then $cr_\ell^\alpha \leq opt$ since cr_ℓ^α is the cut-rate of E_ℓ in G. Thus in this case $E' = E_\ell$, and the goal is achieved. Assume that $\ell > 1$. By the minimality of ℓ, we have that $cr_i^\alpha = opt$ for $i = 1, \dots, \ell - 1$. If $cr_\ell^\alpha > opt$ then by Fact 3 we have $cr(\cup_{i=1}^{\ell} E_i^\alpha) > opt$ which is a contradiction to the definition of opt. Thus, as $cr_\ell^\alpha \neq opt$ we have $cr_\ell^\alpha < opt$ and hence again by Fact 3 the $cr(E') < opt$.

Let $C_1, \dots, C_s$ be the connected components of $G \setminus E'$. Let H be an omni-connected-spanning-subgraph and let $H_1, \dots, H_r$ be the connected components of $E(H) \setminus E'$. Note that for each $i \in \{1, \dots, r\}$ there exists a unique $j \in \{1, \dots, s\}$ such that $E(H_i) \subseteq E(C_j)$. For each $j \in \{1, \dots, s\}$ set I_j to be the set of all $i \in \{1, \dots, r\}$ such that $E(H_i) \subseteq E(C_j)$. Assume that $s < r$, we shall show afterwards that this is indeed true. By the pigeon-hole principle there exists $j \in \{1, \dots, r\}$ such that $|I_j| > 1$. Since C_j is a connected component and H a connected spanning subgraph of G there exist $x, y \in I_j$ and $e = \{u, v\} \in E(C_j) \setminus \cup_{i=1}^{|I_j|} E(H_i)$ such that $u \in V(H_x)$ and $v \in V(H_y)$. Since H is a connected spanning subgraph there is a path in H between u and v this path contains edges not in $E(C_j)$ since u, v are in different connected components of $H \setminus E'$. Thus this path contains an edge $e' \in E'$ since only edges from E' connect the vertices of C_j to the rest of the graph. Consequently $(H \setminus \{e'\}) \cup \{e\}$ is a connected spanning subgraph of G. Let

$P, P' \in \mathcal{P}$ be such that $e \in P$ and $e' \in P'$. By the above P leads to P'. Yet, this can not be since $\alpha(e) < \alpha(e')$ and α agrees with $\mathcal{R}$.

It remains to be shown that indeed $s < r$. By the definition of cut-rate the number of connected components s in $G \setminus E'$ is $cr(E')|E'|$, which is strictly less than $opt|E'|$. Now as α agrees with $\mathcal{R}$ we know that $\mathcal{P}$ refines E'. Hence E' is the union of sets in $\mathcal{P} \setminus \{D\}$. Consequently, by the definition of a omni-connected-spanning-subgraph, we have $E(H) \cap E' = opt|E'|$. Hence $r = opt|E'|$ because the number of connected components in $H \setminus E'$ is the number of edges in $E(H) \cap E'$. □

By definition, there exists a least-core-imputation β with $\mathcal{E}(\beta) = \mathcal{P}$. By Theorem 13.29, we have that β agrees with $\mathcal{R}$ and hence the following holds.

Proposition 13.30. *The relation $\mathcal{R}$ is acyclic.*

This allows us to define the acyclic parent-child relation, which is a simplification of $\mathcal{R}$ and easy to find.

Definition 13.31. *Let $P, P' \in \mathcal{P} \setminus D$ be distinct. We say that P is a* parent *of P' (conversely P' a* child *of P) if P leads to P' and there is no $P'' \in \mathcal{P}$ such that P leads to P'' and P'' leads to P'. We refer to the relation as the parent-child relation and denote it by* $\mathbf{O}$.

The following is an immediate corollary of Theorem 13.29 and Definition 13.31.

Corollary 13.32. *An edge-imputation α is a least-core-imputation if and only if it agrees with* $\mathbf{O}$.

See Figure 13.1 for an example of an omni-connected-spanning-subgraph and the exchangeability of edges between a parent and child. Corollary 13.32 defines a linear inequality for each parent and child in the relation $\mathbf{O}$. Along with the inequalities that define a probability distribution on edges, this gives a small number of two-variable inequalities describing the least-core-polytope.

Definition 13.33. *We say that $P \in \mathcal{P}$ is an ancestor of $P' \in \mathcal{P}$ if there is a chain in* $\mathbf{O}$ *from P to P'.*

Proposition 13.34. *Let $\mathcal{P}^* \subseteq \mathcal{P} \setminus \{D\}$, and $P^* \in \mathcal{P}^*$ and set $E^* = \cup_{P \in \mathcal{P}^*} P$ then*

- *$cr(E^*) = opt$ if $\mathcal{P}^*$ contains only P^* and all its ancestors.*
- *If $cr(E^*) = opt$ and $\mathcal{P}^*$ contains an element that is not P^* or one of its ancestors then it also contains such a P for which $cr(E^* \setminus P) = opt$.*

Proof. Set $\alpha : E(G) \to \mathbb{R}$ so that $\alpha(e) = \frac{1}{|E^*|}$ if $e \in E^*$ and $\alpha(e) = 0$ otherwise. Now α is an edge-imputation, $\mathcal{P}$ refines $\mathcal{E}(\alpha)$ and $E_1^\alpha = E^*$, $E_2^\alpha = E \setminus E^*$ and $\alpha(e) = 0$ for every $e \in D$.

We now prove the first item. For any parent and its child if the child is in $\mathcal{P}^*$ it is either P^* or one of its ancestors. Thus the parent is also an ancestor of P^* and hence is also in $\mathcal{P}^*$. Consequently α agrees with $\mathbf{O}$ and hence by Theorem 13.29, we have that α is a least-core-imputation. This in turn by Theorem 13.13 implies $cr(E^*) = opt$.

We now prove the second item. Assume $cr(E^*) = opt$ and $\mathcal{P}^*$ contains an element that is not P^* or one of its ancestors. Then there exists $P \in \mathcal{P}^*$ that is not an ancestor of any other element in $\mathcal{P}^*$. Since $cr(E^*) = opt$ by Theorem 13.13 α is a least-core-imputation. Hence by Theorem 13.13 α agrees with $\mathbf{O}$.

Set $\beta : E(G) \to \mathbb{R}$ so that $\beta(e) = \frac{1}{|E^*|}$ if $e \in E^*$ and $\beta(e) = 0$ otherwise. Now β is an edge-imputation, $\mathcal{P}$ refines $\mathcal{E}(\beta)$ and $E_1^\beta = E^*$, $E_2^\beta = E \setminus E^*$ and $\beta(e) = 0$ for every $e \in D$. The only way that β does not agree with $\mathbf{O}$ is if a child of P is in $\mathcal{P}^* \setminus \{P\}$, yet this can not be, since P is not an ancestor of any element in $\mathcal{P}^*$. Thus, β agrees with $\mathbf{O}$ and hence, by Theorem 13.13, β is a least-core-imputation. By Theorem 13.13, this implies that $cr(E^* \setminus P) = opt$.

Theorem 13.35. *The parent-child relation $\mathbf{O}$ can be computed in time polynomial in the size of G.*

Proof. We show that for each $P \in \mathcal{P} \setminus \{D\}$ we can find all of the ancestors of P. Once we know the ancestors of each element $\mathcal{P} \setminus \{D\}$ finding the parent of each such element is easy. An ancestor P of P' is also a parent of P' if there does not exist an P^*, that is neither P nor P', such that P is an ancestor of P^* and P^* is an ancestor of P'. Checking this for each pair element and each one of its ancestors requires running time that is polynomial in the size of G. To achieve our goal we use Proposition 13.34.

We next show how to find the ancestors of P'. Set $\mathcal{P}' = \mathcal{P} \setminus \{D\}$ and $E' = \cup_{P \in \mathcal{P}'} P$. If there exists $P^* \in \mathcal{P}'$ such that $P^* \neq P'$ and $cr(E' \setminus P) = opt$ remove it from $\mathcal{P}'$ and recompute E'. Repeat until no such element is found.

Note that this requires $|V|$ repetitions each taking a polynomial time in the size of G. Consequently, the running time is polynomial in the size of G.

When $\mathcal{P} = \mathcal{P} \setminus \{D\}$ we have $cr(E') = opt$ because of the following. By definition there exists a least-core-imputation β such that $\mathcal{E}(\beta) = \mathcal{P}$. Note that $\mathcal{P}'$ is all the non-degenerate sets in $\mathcal{E}(\beta)$ and hence $cr(E') = cr(\cup_{i=1}^{m} E_i^{\beta})$. By Theorem 13.13 $cr_i^{\beta} = opt$ for $i = 1, \ldots, m$, where m is the maximal index such that $x_m^{\beta} > 0$. Hence according to Fact 3 we have $cr(\cup_{i=1}^{m} E_i^{\beta}) = opt$.

Finally we show that at the end what remains in $\mathcal{P}'$ is only P' and all its ancestors. The set P' is never removed from $\mathcal{P}'$. By Proposition 13.34 for any ancestor P^* of P' it is the case that $cr(E' \setminus P^*) < opt$ and hence none of the ancestors of P' are ever removed. Also by Proposition 13.34 as long as $\mathcal{P}'$ does not contain only P' and each one of its ancestors there exists a P^* such that $cr(E' \setminus P^*) = opt$ and hence such an element will be removed. Thus only P' and each one of its ancestors are never removed and consequently they are the only elements remaining in $\mathcal{P}'$ at the end of the process. □

From Definition 11.34, we recall that a least-core-imputation α is a *nucleolus-like-imputation* if the number of the coalitions with the worst excess is minimum possible. First we show how to compute nucleolus-like imputations. To do this, we use the relation **O** to characterize nucleolus-like-imputations. The nucleolus-like-imputations are characterized by the following lemma.

The general idea of the proof runs as follows. First we show that for any α that satisfies the condition of the lemma, every minimum connected spanning subgraph is an omni-connected-spanning-subgraph. Hence, using Proposition 13.26, we get that for any α that satisfies the condition of the lemma, a connected spanning subgraph is a minimum connected spanning subgraph of α if and only if it is an omni-connected-spanning-subgraph. These are the only such least-core-imputations, since any least-core-imputation that does not satisfy the condition of the lemma has a parent and its child whose edges get the same weight. Consequently, by the definition of parent and child, it has a minimum connected spanning subgraph that is not an omni-connected-spanning-subgraph.

Lemma 13.36. *An edge-imputation γ is a nucleolus-like-imputation if and only if $\gamma(e) > 0$ for every $e \in E \setminus D$, and for every $P, P' \in \mathcal{P} \setminus \{D\}$ such that P is a parent of P' and every $e \in P$, $e' \in P'$, we have $\gamma(e') > \gamma(e'')$.*

Proof. Let β be a least-core-imputation such that one of the following holds

1. There exists $P \in \mathcal{P} \setminus \{D\}$ that is a parent of $P' \in \mathcal{P} \setminus \{D\}$ such that $\beta(e) = \beta(e')$ for every $e \in P$ and $e' \in P'$.
2. There exist $P \in \mathcal{P} \setminus \{D\}$ such that $\beta(e) = 0$ for every $e \in P$.

We shall show that β has a minimum connected spanning subgraph that is not an omni-connected-spanning-subgraph. Afterwards we shall show that for every γ for which both conditions do not hold, every minimum connected spanning subgraph of γ is an omni-connected-spanning-subgraph. According to Proposition 13.26, every omni-connected-spanning-subgraph is a minimum connected spanning subgraph of γ, this means that such γ are the only least-core-imputations that have the minimum possible number of minimum connected spanning subgraphs.

Assume the first condition holds for β. By the Definition 13.31 there exists an omni-connected-spanning-subgraph H and edges $e_1 \in P \setminus E(H)$, $e_2 \in P' \cap H$ such that $H' = (T \cup \{e_1\}) \setminus \{e_2\}$ is a connected spanning subgraph. Observe that H' is not an omni-connected-spanning-subgraph of β but is a minimum connected spanning subgraph of β since $\beta(H') = \beta(H) = opt$. Assume the second condition holds for β. Let H be an omni-connected-spanning-subgraph. Recall we assumed $opt < 1$ and hence as H is an omni-connected-spanning-subgraph we have $|E(H) \cap P| = opt|P| < |P|$ and therefore there exists $e \in P \setminus E(H)$. Since H is a minimum connected spanning subgraph of β by Proposition 13.26 and $\beta(e) = 0$ we also have $H \cup \{e\}$ is a minimum connected spanning subgraph of β. Note that $H \cup \{e\}$ is not an omni-connected-spanning-subgraph.

Let γ be some least-core-imputation for which the above two conditions do not hold. That is, $\gamma(e) > 0$ for every $e \in E \setminus D$ and $P \in \mathcal{P} \setminus \{D\}$ that is a parent of $P' \in \mathcal{P} \setminus \{D\}$ and every $e \in P$, $e' \in P'$ we have $\gamma(e) > \gamma(e')$.

From here on let H be a minimum connected spanning subgraph of γ. We next show that H is an omni-connected-spanning-subgraph. Let m be the number of distinct strictly positive values of γ and set $\mathcal{P}_i = \{P \in \mathcal{P} \mid P \in E_i^\gamma\}$ for $i = 1, \ldots, m$. Note that by the definition of γ for every $P \in \mathcal{P} \setminus \{D\}$ there exists $i \in \{1, \ldots, m\}$ such that $P \in \mathcal{P}_i$. Assume by way of contradiction that H is not an omni-connected-spanning-subgraph. Let k be the minimum integer such that there exists $P \in \mathcal{P}_k$ for which $|H \cap P| \neq |P|opt$. Since H is a minimum connected spanning subgraph and γ a least-core-imputation by Proposition 13.8 we have $|H \cap E_k^\gamma| = opt|E_k^\gamma|$. Since $\mathcal{P}$ refines $\mathcal{E}(\gamma)$ we also have $|H \cap E_k^\gamma| = \sum_{E' \in \mathcal{P}_k} |H \cap E'|$ and $|E_k^\gamma| = \sum_{E' \in \mathcal{P}_k} |E'|$ and hence

$$\sum_{E' \in \mathcal{P}_k} |H \cap E'| = opt \sum_{E' \in \mathcal{P}_k} |E'|$$

Therefore the fact that $|H \cap P| \neq |P|opt$ implies that there exists $P' \in \mathcal{P}_k$ such that $|H \cap P'| < |P'|opt$. Let P' be such a set.

Let E^* be the union of P' and all its ancestors (see Definition 13.33). We next show that $cr(E^*) = opt$. Set $\alpha : E(G) \to \mathbb{R}$ so that $\alpha(e) = \frac{1}{|E^*|}$ if $e \in E^*$ and $\alpha(e) = 0$ otherwise. Observe that α is an edge-imputation, $\mathcal{P}$ refines $\mathcal{E}(\alpha)$ and $E_1^\alpha = E^*$, $E_2^\alpha = E \setminus E^*$ and $\alpha(e) = 0$ for every $e \in D$. Now for any parent and its child if the child is contained E^* it is either P' or one of its ancestors. Thus the parent is also an ancestor of P' and hence is also in E^*. Consequently, α agrees with $\mathbf{O}$ and hence by Corollary 13.32 we have that α is a least-core-imputation. This in turn by Theorem 13.13 implies $cr(E^*) = opt$.

Note that because of the strict weight inequalities, all the ancestors of P' are elements in one of the sets $\mathcal{P}_1, \ldots, \mathcal{P}_{k-1}$. Thus for any ancestor P^* of P' we have $|H \cap P^*| = opt|P^*|$. Consequently, $|H \cap E^*| < opt|E^*|$ yet this can not be since $opt|E^*|$ is the minimum number of edges a connected spanning subgraph can have in E^*. $\square$

Using this characterization one can easily check whether α is a nucleolus-like-imputation and one can also easily construct a nucleolus-like-imputation.

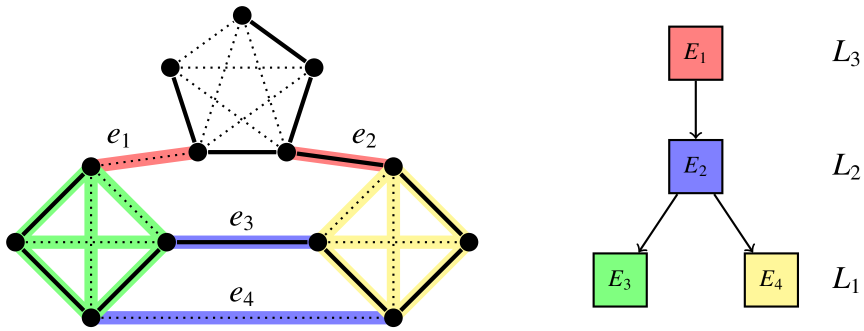

Fig. 13.1. Prime Partition

Example 13.37. Figure 13.1 illustrates the prime-partition $\mathcal{P} = \{E_1, \ldots, E_5\}$. For this graph, $opt = 1/2$. The set $E_1 = \{e_1, e_2\}$, the set $E_2 = \{e_3, e_4\}$, the set E_3 is equal to the edges of the left K_4, the set E_4 is equal to the edges of the right K_4, and the set E_5 is equal to the edges of the K_5. Suppose that least-core-imputation β is such that $\mathcal{E}(\beta) = \mathcal{P}$, and $E_i^\beta = E_i$ for $i = 1, \ldots, 5$. (There will be other least-core-imputations with the same partition in which E_3 and E_4 exchange roles.) Removing E_1 from the graph creates one extra component by removing two edges, so we have $cr_1^\beta = cr(E_1) = opt = 1/2$. Similarly we have $cr_k^\beta = 1/2$ for all $k = 1, \ldots 4$. However, $cr_5^\beta = 4/10 < 1/2$ and so the set E_5 is a degenerate set, as per Lemma 13.23. The Figure 13.1 shows the subgraph H indicated with solid edges. It is an omni-connected-spanning-subgraph, using two edges from each of the K_4's, one edge from the two edges that connect the two K_4's, and one edge from the two edges that connect the two K_4's to the K_5. Within the K_5, an omni-connected-spanning-subgraph can use more than four edges, as this K_5 corresponds to the final element of the prime-partition with any strong linear order and achieves cut-rate $4/10$, which is worse than $opt = 1/2$. The edge e_3 can be replaced with the edge e_1. Thus, the edges in the element of the prime-partition containing e_1 must have weight at least that of the edges in the element of the prime-partition containing e_3. The right part of Figure 13.1 illustrates the partial order $\mathbf{O}$ and its layers $\{L_1, L_2, L_3\}$.

13.7 Nucleolus

We now show how to uniquely maximize the second worst excess of among nucleolus-like-imputations. We are only interested in the case that $opt < 1$, since the graph has $opt = 1$ if and only if it contains a bridge. It is easy to see that bridges are the vetoers of the SCG and the nucleolus divides payoff uniformly among the bridges. From here on we assume the following.

Assumption 1 $opt < 1$.

Before we prove how to compute the nucleolus, we need the following definitions.

Definition 13.38. *We define $\mathcal{L}_1, \mathcal{L}_2, \ldots$ inductively as follows. The set $\mathcal{L}_1$ is all the sinks of $\mathbf{O}$ excluding D. For $j = 2, \ldots,$ we have that $\mathcal{L}_j$ is the set of all the sinks when all elements of $\{D\} \cup (\cup_{i=1,\ldots,j-1}\mathcal{L}_i)$ have been removed from $\mathbf{O}$.*

Note that $\mathbf{O}$ is defined only over nondegenerate elements of $\mathcal{P}$ and hence the degenerate set is not contained in any of $\mathcal{L}_1, \mathcal{L}_2, \ldots$.

Definition 13.39. *The* layers $\mathcal{L} = \{L_1, \ldots, L_t\}$ *of G are $L_i = \cup_{E' \in \mathcal{L}_i} E'$ for $i = 1, \ldots, t$.*

The layers provide a way to partition the edges (except those in the degenerate set). See Figure 13.1 for an example of layers.

The following theorem shows that there is a unique least-core-imputation that maximizes the difference between the payoff of a worst-excess coalition and the payoff of the second worst-excess coalition This unique least-core-imputation is the nucleolus of the spanning connectivity game.

Theorem 13.40. *Set*

$$\kappa = \frac{1}{\sum_{i=1}^{t} i \cdot |L_i|}\ .$$

The nucleolus ν *has* $\nu(e) = i \cdot \kappa$ *for every* $i \in \{1, \ldots, t\}$ *and* $e \in L_i$ *and* $\nu(e) = 0$ *otherwise.*

Proof. The main idea of the proof is that the weight of a connected spanning subgraph with the second smallest weight is $opt + \kappa$, and this must be optimal since all the weights are multiples of κ. For all other nucleolus-like-imputations there is second lightest connected spanning subgraph with weight less than $opt+\kappa$.

Let H be an omni-connected-spanning-subgraph and t the number of layers. Observe that

$$\sum_{e \in E} \nu(e) = \sum_{i=1}^{t} i \cdot |L_i| \cdot \kappa = \kappa \sum_{i=1}^{t} i \cdot |L_i| = \kappa \cdot \kappa^{-1} = 1$$

and hence ν is an edge-imputation. Note that by definition $\mathcal{P}$ refines $\mathcal{E}(\nu)$ and for any $e \in P \in \mathcal{P}$ that is a parent of $e' \in P' \in \mathcal{P}$ we have $\nu(e) > \nu(e')$ and hence ν is a least-core-imputation and specifically a nucleolus-like-imputation.

We now show that the weight of any coalition with the second worst excess is $opt + k$. Afterwards we show that only ν is the only nucleolus-like-imputation for which the weight of any coalition with the second worst excess is at least $opt + k$.

Let $P \in \mathcal{P}$ be such that $P \subseteq L_1$. Since $opt < 1$ Proposition 13.8 implies that there exists $e \in P \setminus E(H)$. By the definition of ν we have $\nu(e) = \kappa$. By Proposition 13.26 $\nu(H) = opt$ and hence $\nu(H \cup \{e\}) = opt + \kappa$. Note that since $\nu(e')$ is a multiple of κ for every $e' \in E$ there does not exist a connected spanning subgraph H' such that $opt < \nu(H') < opt + \kappa$.

Let α be a nucleolus-like-imputation such that the second smallest weight of a connected spanning subgraph is at least $opt + \kappa$. We shall prove by induction on ℓ that $\alpha(e) \geq \ell\kappa$ for every $e \in L_\ell$ and every $\ell = 1, \ldots, t$. Since the only nucleolus-like-imputation that satisfies this conditions is ν this implies that $\alpha = \nu$.

Assume for the sake of contradiction that there exists $e \in P \in L_1$ such that $\alpha(e) < \kappa$. Since $opt < 1$ Proposition 13.8 implies that there exists $e' \in P \setminus E(H)$. By Proposition 13.26 we have $\nu(H) = opt$ and hence because $\alpha(e') = \alpha(e) < \kappa$ we get $\nu(H \cup \{e\}) = opt + \alpha(e') < opt + \kappa$. In addition as α is a nucleolus-like-imputation $\alpha(e') > 0$ and thus $v(H \cup \{e\}) > opt$. Yet the assumption was that for α, any coalition with the second worst excess has payoff at least $opt + \kappa$.

Assume by way of induction that for $\ell - 1$ we have $\alpha(e) \geq (\ell - 1)\kappa$ for every $e \in L_{\ell-1}$. Assume for the sake of contradiction that there exists $e \in P \in L_\ell$ such that $\alpha(e) < \ell \cdot \kappa$. By the definition of L_ℓ, there exists $P' \subseteq L_{\ell-1}$ such that P is a parent of P'. Consequently, there exists an omni-connected-spanning-subgraph H', $e' \in P \setminus E(H')$ and $e'' \in P' \cap E(H')$ such that $H^* = (H' \setminus \{e''\}) \cup \{e'\}$ is and spanning tree of G. Observe that $\alpha(H^*) = \alpha(H') + \alpha(e') - \alpha(e'')$. By Proposition 13.26, we have $\alpha(H') = opt$ and hence $\alpha(H^*) = opt + \alpha(e') - \alpha(e'')$. By the induction assumption, we have $\alpha(e'') \geq (\ell - 1)\kappa$ and therefore as $\alpha(e'') = \alpha(e) < \ell \cdot \kappa$ we get $\alpha(H^*) < opt + \kappa$. In addition as α is a nucleolus-like-imputation we have $\alpha(e') > \alpha(e'')$ and therefore $v(H^* \cup \{e\}) > opt$. Yet the assumption was that for α, any coalition with the second worst excess has payoff at least $opt + \kappa$. □

13.8 Wiretap game

Never interrupt your enemy when he is making a mistake.
- Napoleon Bonaparte

However beautiful the strategy, you should occasionally look at the results.
- Winston Churchill

In this section, we formally define the wiretap game and discusss its connection to the SCG. The strategic form of the wiretap game is defined implicitly by the graph $G = (V, E)$. The wiretapper chooses an edge and the hider chooses a spanning subgraph of G. The wiretapper receives payoff 1 (wins), if the edge he chooses is part of the spanning subgraph chosen by the hider, and receives payoff 0 (loses) otherwise. Thus, the value of the game is the probability that the wiretapper can secure for wiretapping the spanning subgraph chosen by the hider. For this reason we choose to write the game as a constant-sum, rather than zero-sum game.

The pure strategies of the wiretapper are the edges E and the pure strategies of the hider are the set of spanning subgraphs $\mathcal{S}$, with a typical element of $\mathcal{S}$, which is a set of edges, denoted by S.

We could define the wiretap game by only allowing the hider to pick spanning trees. However, our defnition with connected spanning subgraphs allows a clean connection to the spanning connectivity game. Also, this does not change the nucleolus strategy of the wiretapper.

Let $\Delta(A)$ be the set of mixed strategies (probability distributions) on a finite set A, and let $\mathbb{I}_{a\in A}$ be the indicator function that takes value 1 if $a \in A$ and 0 otherwise. By the well-known maxmin theorem for finite zero-sum games, the wiretap game $\Gamma(G)$ has a unique *value*, defined by

$$val(\Gamma) = \max_{x\in\Delta(E)} \min_{S\in\mathcal{S}} \sum_{e\in E} \mathbb{I}_{e\in S} \cdot x_e = \min_{y\in\Delta(\mathcal{S})} \max_{e\in E} \sum_{S\in\mathcal{S}} \mathbb{I}_{e\in S} \cdot y_S \,. \tag{13.26}$$

The equilibrium or *maxmin* strategies of the wiretapper are

$$\{x \in \Delta(E) | \sum_{e\in E} \mathbb{I}_{e\in S} \cdot x_e \geq val(\Gamma) \text{ for all } S \in P\} \,. \tag{13.27}$$

Playing any maxmin strategy guarantees the wiretapper to achieve a probability of successful wiretapping of at least $val(\Gamma)$. The equilibrium or *minmax* strategies of the hider are

$$\{y \in \Delta(P) | \sum_{S\in P} \mathbb{I}_{e\in S} \cdot y_S \leq val(\Gamma) \text{ for all } e \in E\} \,. \tag{13.28}$$

Playing any minmax strategy guarantees the hider to suffer a probability of successful wiretapping of no more than $val(\Gamma)$. We have the following simple observation.

Observation 13.41 *The set of imputations in the least core of the SCG(G) are the maxmin strategies of the wiretap game $\Gamma(G)$.*

The problem of finding a maxmin strategy of $\Gamma(G)$, defined by (13.26) and (13.27), can be written as finding an x to solve

Spanning connectivity game	Wiretap game
player (edge)	pure strategy of wiretapper
winning coalitions	pure strategies of hider
imputation	wiretapper's strategy
least-core imputation	maxmin strategy
nucleolus-like imputation	maxmin strategy minimizing no. of best responses
nucleolus	unique desirable maxmin strategy
1-$\epsilon_1(x)$	min probability of successful wiretap using strategy x
ϵ_1 coalition for imputation x	best response to the maxmin strategy x
ϵ_1 coalitions of nucleolus-like imputations	best responses to every maxmin strategy

Table 13.1. Spanning connectivity game and the wiretap game

$$\begin{aligned} &\max z \\ &\text{s.t.} \quad \textstyle\sum_{e \in E} \mathbb{I}_{e \in S} \cdot x_e \geq z \text{ for all } S \in \mathcal{S} \\ &\qquad x \in \Delta(E) \end{aligned} \tag{13.29}$$

It is easy to see that LP 13.1 and LP 13.29 have the same solution and objective function. Therefore a least core imputation of the SCG corresponds to a maxmin strategy for the wiretapper in the wiretap game and the value of the wiretap game is equal to the *cr*. Interpreted as a maxmin strategy in the wiretap game, the nucleolus of the SCG game has the following desirable properties. A nucleolus-like strategy is a maxmin strategy for the wiretapper which minimizes the number of pure best responses of the hider. The nucleolus is the unique nucleolus-like strategy which maximizes the gain in the probability of a successful wiretap if the hider fails to play a best response. We summarize the relation between the SCG and the wiretap game in Table 13.1.

13.9 Conclusion

We saw that although computing the Banzhaf values and Shapley values of the SCG are #P-complete, computing the nucleolus is in P. The idea of a principal partition which refines the player partition for other least core payoffs may be generalized for other cooperative games. This kind of partition can give valuable information about the relations between the players. An interesting research question will be to characterize, compute or utilize the principal partition for well-known cooperative games.

Just as there is a corresponding wiretap game for a SCG, a corresponding zero-sum game can be formalized for any simple coalitional game. The maximizer player in the zero-sum game chooses to 'control' a player whereas the minimizer player chooses a secret winning coalition. The maximizer gets value one if his controlled player is in the winning coalition chosen by the minimizer and gets value zero otherwise. Just like in the wiretap game, the least core is a maxmin strategy and the nucleolus is a highly desirable maxmin strategy.

There are a number of natural extensions to the wiretap game. The problem changes if the wiretapper is allowed to pick multiple edges. If the number of edges to be tapped is an input of the modified wiretap game, then it is interesting to investigate the complexity of the problem.

Part V

Conclusion

14

Concluding remarks

When a scientist says: "This is the very end, nobody can do anything more here," then he is no scientist.
- L. Gould

Not every end is the goal. The end of a melody is not its goal, and yet if a melody has not reached its end, it has not reached its goal. A parable.
- Friedrich Nietzsche

Abstract In the thesis, open problems and conclusions for each chapter were discussed individually. In the final chapter, relevant broad research issues are briefly mentioned.

In this thesis, the aim was to integrate approaches from theoretical computer science, multiagent systems, social choice theory and cooperative game theory with respect to simple games. Voting and resource allocation are activities not only restricted to human societies. Virtual rational agents may also be present in multiagent systems. Therefore, concepts from social choice theory and cooperative game theory were shown to be highly relevant in computer science and multiagent systems. A key conclusion of the thesis is that the algorithmic lens is fundamental in examining models and solutions in game theory. Also, computational complexity is an important consideration in proposing solutions and designing mechanisms. An algorithmic perspective in game theory promises to play a greater role as various game theoretic concepts are used in large multiagent systems.

The contributions of the thesis range from theoretical algorithmic and computational results to practical implementations of algorithms and applied analysis of real world social choice models. The thesis also partially or fully answers a number of open questions regarding computation of cooperative game solutions for simple games. In Part II, simple games were examined in detail from the perspective of algorithmic voting theory. This included the study of computing the influence of players, classifying which WVGs are tractable and designing WVGs. In Part III, we contributed to a growing line of work where computational complexity is considered as a barrier to manipulations in voting systems. For many cases, bounds of how much manipulation can help or harm were also presented In Part IV, a broad survey of computational complexity of computing cooperative game solutions for simple games was presented. Chapter 11 includes structural results on cooperative game solutions for monotonic cooperative game. The results may shed light on computation of solutions for specific representations of monotonic cooperative games. In Chapters 12 and 13, we also introduced and studied a natural cooperative game on graphs called the spanning connectivity game.

Throughout the thesis, conclusions and open problems have been mentioned at the end of each chapter. We saw that in general, the computation of solutions becomes easier as we scan the following list from left to right: Shapley-Shubik index, Banzhaf index, nucleolus, Holler index, Deegan-Packel index, least-core and core. It is an open question to construct a representation of a simple game, where the complexity of computing the Banzhaf value is more than the complexity of computing the Banzhaf index. Similarly, it is an open question to construct a representation of a simple game, where the complexity of computing the Shapley-Shubik index is more than the complexity of computing the Banzhaf value.

In the thesis, we restricted ourselves to coalitional games with transferable utility. Coalition games with non-transferable utility have not been considered. With respect to computational complexity of manipulation, we have considered worst case complexity. This may not be a sufficient safeguard against manipulation in the average case. Further research in average time complexity for manipulation in social choice is a recent and important direction of research.

An important assumption in the evaluation of power indices is that each coalition has the same probability of forming. In reality, players have ideological preferences or communication constraints. This may impact the actual voting power of a player. Also, when we considered cooperative game solutions, we assumed that the grand coalition forms and it is the value of the grand coalition which is distributed among the players. Although, this is a natural assumption, it may not be true in case players partition themselves.

In the thesis, we saw that computational complexity for coalitional games depends on the representation of the game. There is much work to be done in devising coalitional games which are not only expressive but also compact. One possible direction is to use graphs as compact ways to represent dependencies among players, forbidden coalitions or consistency orderings of political positions. There is scope to utilize the combination of graphs and WVGs to model complex decision-making scenarios. Moreover, it will be useful to develop models which incorporate uncertainty of information such as values of coalitions in the coalitional game. Some progress has already been made in that direction (see for instance [110]). On the economics front, voting power theorists are still grappling with the independence-of-players assumptions during the voting power computation. In real life, players have varying preferences in forming coalitions with different players. It is an interesting challenge for mathematical economists to formalize this tension in a satisfactory way.

Another area of future research is the formulation of approximate notions of cooperative game solutions. Computer scientists have come up with approximations of the Nash equilibria which may allow for easier computation. Future algorithmic work in coalitional games promises to follow a similar methodology where the trade-off between the quality of solution and computational ease is considered. Within the area of coalition formation, one direction for future work is to study the computational complexity of forming stable coalitions. Although the different stability-based cooperative game solutions predict the coalition formation process, only recently have algorithmic aspects of coalition formation been examined. Also, most of the work in equilibria convergence considers individual players and not coalitions.

In most settings in cooperative game theory, it is assumed that the coalitional game is already known and represented. However, it may be the case that the values of all coalitions are not known a priori. A relevant research problem is that of learning a coalitional game while minimizing the number of queries. In general, a coalitional game requires a query per coalition. However, when the coalitional game contains some structure (like monotonicity) or belongs to a particular class of coalitional games, there is a need to devise efficient algorithms to efficiently construct the game exactly or approximately. The research will involve using the latest tools from learning theory and applying them to cooperative game theory.

Part IV of the thesis examined problems of resource allocation on networks. We considered settings where the value of each coalition is already known and public. In many cases such as auctions, players and coalitions have private values. In resource allocation, there is a growing line of research on how to design resource allocation where players have incentive to provide truthful valuations. There is huge scope for further research on combinatorial optimization games and cost sharing on networks. This will better inform us how to incentivize cooperation in peer-to-peer settings and also how to share and price resources on networks. This economics-driven approach is likely to make an impact in future engineering and design issues in networks.

Part VI

Appendices

A

MWVG Program

(Based on Chapter 5)

```
(*:Mathematica Version:5.2, Package Version:1.10 *)
(*:Name:Compute_Banzhaf_Indices_of_MWVG *)
(*:Authors:Haris Aziz (haris.aziz@warwick.ac.uk) *)
(*:Summary: The program takes as input a multiple
    weighted voting game with integer weights and quotas. It uses the
    generating functions to compute the Banzhaf index of every player*)
(*:References: Computing power indices in weighted multiple majority games
    by E.Algaba, Mathematical Social Sciences 46 (2003) pages 63-80.*)

w = {{5, 2, 1, 1}, {3, 2, 1, 1}}; q = {{7}, {5}};
Print["weights: ", MatrixForm[w]]; Print["quotas: ", MatrixForm[q]];
m = Part[Dimensions[w], 1]; Print["There are ", m, " weighted voting games"]
n = Part[Dimensions[w], 2]; Print["There are ", n, " players"];

Array[symmwithprevious, n]; symmwithprevious[1] = False;
For[i = 2, i < n + 1, i ++, symmwithprevious[i] = True;];
For[i = 2, i < n + 1, i ++ , For[j = 1, j < m + 1, j ++,
   If[w[[j, i]] != w[[j, i - 1]], symmwithprevious[i] = False;,]]];
For[i = 1, i < n + 1, i ++, If[ symmwithprevious[i],
   Print["Player ", i, " has same weights as player ", i - 1],
   Print["Player ", i, " does not have same weights as player ", i - 1] ]];

Bfunction = Product[1 + Product[x[i] ^w[[i, j]], {i, 1, m}], {j, 1, n}];
longBfunction = Expand[Bfunction]; Print["Bfunction = ", Bfunction];
Print[Array[x, m]]; maincoefmatrix = CoefficientList[longBfunction, Array[x, m]];
Print["CoefficientMatrix for the main GF is ", MatrixForm[maincoefmatrix]];

For[j = 1, j < n + 1, j ++ , b[j] = Bfunction / (1 + Product[x[i] ^w[[i, j]], {i, 1, m}]);
 longb[j] = Expand[b[j]]; Print["Generating Function of player ", j, "=", b[j]] ]

kk = {}; For[j = 1, j < m + 1, j ++, kk = Append[kk, {q[[j, 1]] + 1, Total[w[[j]]] + 1}]];
winningmatrix = Take[maincoefmatrix, Part[kk, 1], Part[kk, 2]];
numofwinningcoalitons = Total[winningmatrix, m];
Array[x, m]; Array[coefmatrix, m];

coefmatrix[1] = CoefficientList[longb[1], Array[x, m]];
Print["Coefficient Matrix of player", 1, " =", MatrixForm[coefmatrix[1]]];
For[j = 2, j < n + 1, j ++ , If[symmwithprevious[j], coefmatrix[j] = coefmatrix[j - 1],
   coefmatrix[j] = CoefficientList[longb[j], Array[x, m]];];
  Print["Coefficient Matrix of player", j, " =", MatrixForm[coefmatrix[j]]]];

d = Table[0, {m}, { n}];
For[t = 1, t < m + 1, t ++, For[i = 1, i < n + 1, i ++, d[[t, i]] = q[[t, 1]] - w[[t, i]]]];
e = Table[0, {m}, {n}];
For[t = 1, t < m + 1, t ++,
  For[i = 1, i < n + 1, i ++, e[[t, i]] = Total[w[[t]]] - w[[t, i]]]];

For[i = 1, i < n + 1, i ++, ll[i] = {};];
For[i = 1, i < n + 1, i ++, For[t = 1, t < m + 1, t ++,
  ll[i] = Append[ll[i], {d[[t, i]] + 1, Part[Dimensions[coefmatrix[i]], t]}];]]

Print ("Computing Small1 matrices");
small1[1] = Take[coefmatrix[1], Part[ll[1], 1], Part[ll[1], 2] ];
Print["small1[", 1, "] = ", MatrixForm[small1[1]]]
```

```
For[i = 2,  i < n + 1, i ++, If[symmwithprevious[i], small1[i] = small1[i - 1],
   small1[i] = Take[coefmatrix[i], Part[ll[i], 1], Part[ll[i], 2] ]];
  Print["small1[", i, "] = ", MatrixForm[small1[i]]]];

For[i = 1, i < n + 1, i ++, sum1[i] = Total[small1[i], Infinity];
  Print["sum1[", i, "] = ",  sum1[i]]];
g = Table[0, {m}, {n}];
For[t = 1, t < m + 1, t ++,
  For[i = 1, i < n + 1, i ++, g[[t, i]] = Total[w[[t]]] - w[[t, i]] + 1;]];

mm[1] = {}; For[i = 1, i < n + 1, i ++,  mm[i] = {};] ;

Array[errorcheck, n];
For[z = 1, z < n + 1, z ++, errorcheck[z] = 0;];
For[i = 1, i < n + 1, i ++,
  For[t = 1,  t < m + 1, t ++,  mm[i] = Append[mm[i], {q[[t, 1]] + 1, g[[t, i]]}];
   If[q[[t, 1]] + 1 > g[[t, i]], errorcheck[i] = 1;,];] ];

For[i = 1,  i < n + 1, i ++, If[errorcheck[i] == 1, small2[i] = {},
   small2[i] = Take[coefmatrix[i], Part[mm[i], 1], Part[mm[i], 2] ]];
  Print["small2[", i, "] = ", MatrixForm[small2[i]]]];
For[i = 1, i < n + 1, i ++, If[small2[i] == {}, sum2[i] = 0,
   sum2[i] = Total[small2[i], Infinity]]; Print["sum2[", i, "] = ",  sum2[i]]];

totalswings = 0;
For[i = 1, i < n + 1, i ++, swings[i] = sum1[i] - sum2[i];
  totalswings =  totalswings + swings[i]; Print["swings[", i, "] = ", swings[i]]];
For[i = 1, i < n + 1, i ++, banzhafindex[i] = swings[i] / totalswings;
  Print["Banzhaf Index of player", i, " is ",  banzhafindex[i]]];

vetoplayerlist = {};
For[i = 1, i < n + 1,  i ++, isvetoplayer = False; For[j = 1,  j < m + 1,  j ++,
  If[ (Total[w[[j]]] - w[[j, i]]) < q[[j, 1]], isvetoplayer = True;, ] ];
 If[isvetoplayer,  vetoplayerlist = Append[vetoplayerlist, i];
  Print[i, " has veto powers"], Print[i, " does not have veto powers"]]]

Print["Number of winning coalitions = ", numofwinningcoalitons];
```

weights: $\begin{pmatrix} 5 & 2 & 1 & 1 \\ 3 & 2 & 1 & 1 \end{pmatrix}$

quotas: $\begin{pmatrix} 7 \\ 5 \end{pmatrix}$

There are 2 weighted voting games

There are 4 players

Player 1 does not have same weights as player 0

Player 2 does not have same weights as player 1

Player 3 does not have same weights as player 2

Player 4 has same weights as player 3

Bfunction = $(1 + x[1]\, x[2])^2\, (1 + x[1]^2\, x[2]^2)\, (1 + x[1]^5\, x[2]^3)$

{x[1], x[2]}

CoefficientMatrix for the main GF is $\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$

Generating Function of player 1= $(1 + x[1]\, x[2])^2\, (1 + x[1]^2\, x[2]^2)$

Generating Function of player 2= $(1 + x[1]\, x[2])^2\, (1 + x[1]^5\, x[2]^3)$

Generating Function of player 3= $(1 + x[1]\, x[2])\, (1 + x[1]^2\, x[2]^2)\, (1 + x[1]^5\, x[2]^3)$

Generating Function of player 4= $(1 + x[1]\, x[2])\, (1 + x[1]^2\, x[2]^2)\, (1 + x[1]^5\, x[2]^3)$

Coefficient Matrix of player1 = $\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$

Coefficient Matrix of player2 = $\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$

Coefficient Matrix of player3 = $\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$

Coefficient Matrix of player4 = $\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$

small1[1] = $\begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

small1[2] = $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

small1[3] = $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

small1[4] = $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

```
sum1[1] = 5

sum1[2] = 4

sum1[3] = 3

sum1[4] = 3

small2[1] = {}

small2[2] = ( 1 )
```

small2[3] = $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

small2[4] = $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

```
sum2[1] = 0

sum2[2] = 1

sum2[3] = 2

sum2[4] = 2

swings[1] = 5

swings[2] = 3

swings[3] = 1

swings[4] = 1
```

Banzhaf Index of player1 is $\frac{1}{2}$

Banzhaf Index of player2 is $\frac{3}{10}$

Banzhaf Index of player3 is $\frac{1}{10}$

Banzhaf Index of player4 is $\frac{1}{10}$

```
1 has veto powers

2 does not have veto powers

3 does not have veto powers

4 does not have veto powers

Number of winning coalitions = 5
```

Part VII

Backmatter

References

1. E. Algaba, J. M. Bilbao, and J. R. Fernandez, *The distribution of power in the European Constitution*, European Journal of Operational Research **176** (2007), no. 3, 1752–1755. [16, 78, 108]
2. E. Algaba, J. M. Bilbao, J. R. Fernandez Garcia, and J. J. Lopez, *Computing power indices in weighted multiple majority games*, Mathematical Social Sciences **46** (2003), no. 1, 63–80. [24, 73]
3. N. Alon and P. Edelman, *The inverse Banzhaf problem*, Social Choice and Welfare **10.1007/s00355-009-0402-8** (2009). [84]
4. J.M Alonso-Meijide, *Generating functions for coalition power indices: An application to the IMF*, Annals of Operations Research **137** (2005), 21–44. [78]
5. G. Arcaini and G. Gambarelli, *Algorithm for automatic computation of the power variations in share tradings*, Calcolo **23** (1986), no. 1, 13–19. [78, 108]
6. S. Arnborg, D. G. Corneil, and A. Proskurowski, *Complexity of finding embeddings in a k-tree*, SIAM J. Algebraic Discrete Methods **8** (1987), no. 2, 277–284. [194]
7. S. Arnborg and A. Proskurowski, *Linear time algorithms for NP-hard problems restricted to partial k-trees*, Discrete Appl. Math. **23** (1989), no. 1, 11–24. [194]
8. K. J. Arrow, *Social choice and individual values (2nd edition)*, Wiley, New York, 1963. [69]
9. R. J. Aumann, B. Peleg, and P. Rabinowitz, *A method for computing the kernel of n-person games*, Mathematics of Computation **19** (1965), no. 92, 531–551. [165, 177]
10. H. Aziz, *Complexity of comparison of influence of players in simple games*, Proceedings of the Second International Workshop on Computational Social Choice (COMSOC 2008), 2008, pp. 61–72. [XVII, 17]
11. ———, *Simple games: complexity of length, duality and bribery*, European Workshop on Multiagent Systems (2008). [XVII, 20]
12. H. Aziz, F. Brandt, and P. Harrenstein, *Monotone cooperative games and their threshold versions*, Submitted (2009). [171]
13. H. Aziz, O. Lachish, M. Paterson, and R. Savani, *Power indices of spanning connectivity games*, The 5th International Conference on Algorithmic Aspects in Information and Management (AAIM), 2009. [XVII, 21]
14. ———, *Wiretapping a hidden network*, Tech. report, http://arxiv.org/abs/0909.5293, 2009. [XVII, 21]

15. H. Aziz and M. Paterson, *Computational and combinatorial aspects of multiple weighted voting games*, Algorithms and Complexity in Durham - Proceedings of the Third ACiD Workshop, Texts in Algorithmics, King's College, London, 2007. [XVII, 18]
16. ______, *Classification of computationally tractable weighted voting games*, Lecture Notes in Engineering and Computer Science, World Congress on Engineering **1** (2008), 129–134. [XVII, 18, 67, 132]
17. ______, *Complexity of some aspects of control and manipulation in weighted voting games*, Annales du Lamsade, Proceedings of the DIMACS-LAMSADE Workshop on Algorithmic Decision Theory 2008, 2008. [XVII, 19]
18. ______, *False name manipulations in weighted voting games: splitting, merging and annexation*, AAMAS '09: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems, 2009, pp. 409–416. [XVII, 20]
19. H. Aziz, M. Paterson, and D. Leech, *Efficient algorithm for designing weighted voting games*, Multitopic Conference, 2007. INMIC 2007. IEEE International (2007), 1–6. [XVII, 19]
20. R. Bacher, *Determinants of matrices related to the pascal triangle*, J. Théor. des Nombres Bordeaux **14** (2002), 19–41. [49, 191]
21. Y. Bachrach and E. Elkind, *Divide and conquer: False-name manipulations in weighted voting games*, The Seventh International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008) (Estoril, Portual), May 2008, pp. 975–982. [108, 116, 117, 118, 122, 124, 131, 132]
22. Y. Bachrach, E. Markakis, A. D. Procaccia, J. S. Rosenschein, and A. Saberi, *Approximating power indices*, AAMAS, 2008, pp. 943–950. [66, 197, 198]
23. Y. Bachrach, R. Meir, M. Zuckerman, J. Rothe, and J. S. Rosenschein, *The cost of stability in weighted voting games (extended abstract)*, The Eighth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009) (Budapest, Hungary), May 2009. [181, 184]
24. Y. Bachrach and J. S. Rosenschein, *Computing the Banzhaf power index in network flow games*, AAMAS '07: Proceedings of the 6th International Joint conference on Autonomous agents and multiagent systems (New York, NY, USA), ACM, 2007, pp. 1–7. [187]
25. Y. Bachrach and J. S. Rosenschein, *Coalitional skill games*, AAMAS, 2008, pp. 1023–1030. [51]
26. ______, *Power in threshold network flow games*, Journal of Autonomous Agents and Multi-Agent Systems **18** (2009), no. 1, 106–132. [51, 156]
27. Y. Bachrach, J. S. Rosenschein, and E. Porat, *Power and stability in connectivity games*, AAMAS, 2008, pp. 999–1006. [51, 187]
28. R. A. Bailey, *Surveys in combinatorics*, Cambridge University Press, 1997. [190]
29. M. O. Ball, *Computational complexity of network reliability analysis: An overview*, IEEE Transactions on Reliability **35** (1986), no. 3, 230–239. [187, 189, 190]
30. M. O. Ball and J. S. Provan, *Disjoint products and efficient computation of reliability*, Oper. Res. **36** (1988), no. 5, 703–715. [47]
31. J. F. Banzhaf, *Weighted voting doesn't work*, Rutgers Law Review **19** (1965), 317–343. [25]
32. J. Bartholdi, III and J. Orlin, *Single transferable vote resists strategic voting.*, Social Choice and Welfare **8** (1991), no. 4, 341–354. [124]
33. D. Bean, J. Friedman, and C. Parker, *Simple majority achievable hierarchies*, Theory and Decision **65** (2008), no. 4, 285–302. [79]

34. B. Ben-Moshe and Y. Dinitz, *Fast additive constant approximation algorithms for the safe deposit boxes problem with two and three currencies*, Proceedings of the 19th Annual Canadian Conference on Computational Geometry, CCCG, 2007, pp. 53–56. [140]
35. T. Bietenhader and Y. Okamoto, *Core stability of minimum coloring games*, Math. Oper. Res. **31** (2006), no. 2, 418–431. [166]
36. J. M. Bilbao, J. R. Fernandez, A. Jimenez Losada, and J. J. Lopez, *Generating functions for computing power indices efficiently*, citeseer.ist.psu.edu/452269.html (2000). [64, 65, 67, 81, 124]
37. J. C. Bioch and T. Ibaraki, *Complexity of identification and dualization of positive boolean functions*, Inf. Comput. **123** (1995), no. 1, 50–63. [144]
38. P. Borm, H. Hamers, and R. Hendrickx, *Operations research games: A survey*, TOP: An Official Journal of the Spanish Society of Statistics and Operations Research **9** (2001), no. 2, 139–199. [155, 187]
39. S. F. Brams and P. J. Affuso, *Power and size: A new paradox*, Theory and Decision **7** (1976), 29–56. [65, 67]
40. U. Brandes and T. Erlebach, *Network analysis: Methodological foundations*, Springer, 2005. [198]
41. F. Carreras, *On the design of voting games*, Mathematical Methods of Operations Research **59** (2004), no. 1, 503–515. [80]
42. F. Carreras and J. Freixas, *Complete simple games*, Mathematical Social Sciences **32** (1996), no. 2, 139–155. [36]
43. N. Chakravarty, A. M. Goel, and T. Sastry, *Easy weighted majority games*, Mathematical Social Sciences **40** (2000), no. 2, 227–235. [61, 63, 64, 67]
44. V. Chandru and M. R. Rao, *Linear programming*, Algorithms and Theory of Computation Handbook, CRC Press, 1999. [141]
45. E. Cheng and W. H. Cunningham, *A faster algorithm for computing the strength of a network*, Inf. Process. Lett. **49** (1994), no. 4, 209–212. [219]
46. Y. Chevaleyre, P. E. Dunne, U. Endriss, J. Lang, M. Lemaitre, N. Maudet, J. Padget, S. Phelps, J. A. Rodríguez-Aguilar, and P. Sousa, *Issues in multiagent resource allocation*, Informatica **30** (2006), no. 1, 3–31. [155]
47. Y. Chevaleyre, U. Endriss, J. Lang, and N. Maudet, *A short introduction to computational social choice*, Proceedings of the 33rd Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM-2007), LNCS, vol. 4362, Springer-Verlag, January 2007, pp. 51–69. [10]
48. C. K. Chow, *On the characterization of threshold functions*, FOCS, 1961, pp. 34–38. [27]
49. J. S. Coleman, *Control of collectivities and the power of a collectivity to act*, Social Choice (B. Lieberman, ed.), New York, 1971, pp. 269–300. [26]
50. V. Conitzer and T. Sandholm, *Complexity of manipulating elections with few candidates*, Eighteenth National Conference on Artificial Intelligence (Menlo Park, CA, USA), American Association for Artificial Intelligence, 2002, pp. 314–319. [145]
51. P. G. D. Cortona, C. Manzi, A. Pennisi, F. Ricca, and B. Simeone, *Evaluation and optimization of electoral systems (Monographs on Discrete Mathematics and Applications)*, Society for Industrial Mathematics, 1987. [145]
52. Y. Crama and L. Leruth, *Control and voting power in corporate networks: Concepts and computational aspects*, European Journal of Operational Research **178** (2007), no. 3, 879–893. [187]

53. W. H. Cunningham, *Optimal attack and reinforcement of a network*, J. ACM **32** (1985), no. 3, 549–561. [201, 205, 219]
54. I. Curiel, *Cooperative game theory and applications: Cooperative games arising from combinatorial optimization problems*, Springer, 1997. [155, 187]
55. B. B. de Mesquita, *Minimum winning coalition, in politics*, Neil J. Smelser and Paul B. Baltes, Editor(s)-in-Chief, International Encyclopedia of the Social & Behavioral Sciences (2001), 9889–9891. [27]
56. J. Deegan and E. W. Packel, *A new index of power for simple n-person games*, International Journal of Game Theory **7** (1978), no. 2, 113–123. [27]
57. V. G. Deineko and G. J. Woeginger, *On the dimension of simple monotonic games*, European Journal of Operational Research **170** (2006), no. 1, 315–318. [36, 37, 71]
58. X. Deng and Q. Fang, *Algorithmic cooperative game theory*, Pareto Optimality, Game Theory And Equilibria, Springer Optimization and Its Applications, vol. 17, Springer, 2008, pp. 159–185. [13, 155, 172, 183]
59. X. Deng, Q. Fang, and X. Sun, *Finding nucleolus of flow game*, SODA '06: Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms (New York, NY, USA), ACM, 2006, pp. 124–131. [174, 201]
60. X. Deng and C. H. Papadimitriou, *On the complexity of cooperative solution concepts*, Math. Oper. Res. **19** (1994), no. 2, 257–266. [46, 155, 184]
61. E. A. Dinic and A. V. Karzanov, *Boolean optimization problems with uniform constraints*, Institute for Control Sci. (1978), Reprint. [139]
62. T. Driessen, *Cooperative games, solutions and applications*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1988. [155, 160]
63. P. Dubey and L. S. Shapley, *Mathematical properties of the Banzhaf power index*, Mathematics of Operations Research **4** (1979), no. 2, 99–131. [27, 118, 143]
64. E. Einy, *The desirability relation of simple games*, Mathematical Social Sciences **10** (1985), no. 2, 155–168. [36]
65. E. Einy and D. Wettstein, *Equivalence between bargaining sets and the core in simple games*, International Journal of Game Theory **25** (1996), no. 1, 65–71. [162, 179]
66. T. Eiter, K. Makino, and G. Gottlob, *Computational aspects of monotone dualization: A brief survey*, Discrete Applied Mathematics **156** (2008), no. 11, 2035–2049. [144]
67. E. Elkind, *Private communication*, 2009. [120]
68. E. Elkind, L. A. Goldberg, P. Goldberg, and M. Wooldridge, *Computational complexity of weighted threshold games*, AAAI-07 (Twenty-Second National Conference on Artificial Intelligence) (2007), 718–723. [155, 158, 163, 164, 165, 167, 173, 175, 180, 182, 184, 201]
69. ———, *On the dimensionality of voting games*, Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI-2008) (Chicago, Illinois, USA), July 13-17 2008. [37, 108]
70. E. Elkind and D. Pasechnik, *Computing the nucleolus of weighted voting games*, SODA '09: Proceedings of the Nineteenth Annual ACM -SIAM Symposium on Discrete Algorithms (Philadelphia, PA, USA), 2009, pp. 327–335. [155, 174, 180, 201]
71. K. Engel, *Sperner theory*, Cambridge University Press, New York, NY, USA, 1997. [16]

72. A. Fabrikant, C. H Papadimitriou, and K. Talwar, *The complexity of pure Nash equilibria*, STOC '04: Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing (New York, NY, USA), ACM, 2004, pp. 604–612. [11]
73. U. Faigle, W. Kern, and J. Kuipers, *Computing the nucleolus of min-cost spanning tree games is NP-hard*, Int. J. Game Theory **27** (1998), no. 3, 443–450. [174, 201]
74. ______, *Computing an element in the lexicographic kernel of a game*, Mathematical Methods of Operations Research **63** (2006), no. 3, 427–433. [178]
75. P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe, *A richer understanding of the complexity of election systems*, Fundamental Problems in Computing: Essays in Honor of Professor Daniel J. Rosenkrantz (S. Ravi and S. Shukla, eds.), 2009. [145]
76. P. Faliszewski and L. A. Hemaspaandra, *The complexity of power-index comparison*, AAIM, 2008, pp. 177–187. [46]
77. Piotr Faliszewski, Edith Elkind, and Michael Wooldridge, *Boolean combinations of weighted voting games*, AAMAS '09: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems, 2009, pp. 185–192. [69]
78. Q. Fang, *Nucleolus*, Encyclopedia of Algorithms, 2008. [173]
79. Q. Fang and L. Kong, *Core stability of vertex cover games*, WINE, 2007, pp. 482–490. [166]
80. S. S. Fatima, M. Wooldridge, and N. R. Jennings, *A randomized method for the Shapley value for the voting game*, AAMAS '07: Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems (New York, NY, USA), ACM, 2007, pp. 1–8. [66]
81. J. Feigenbaum, D. C. Parkes, and D. M. Pennock, *Computational challenges in e-commerce*, Commun. ACM **52** (2009), no. 1, 70–74. [10]
82. D. Felsenthal and M. Machover, *The measurement of voting power*, Edward Elgar Publishing, Cheltenham, UK, 1998. [53, 82, 117, 126, 130, 131]
83. D. S. Felsenthal and M. Machover, *Analysis of QM rules in the draft constitution for Europe proposed by the European Convention, 2003*, Social Choice and Welfare **23** (2004), no. 1, 1–20. [78]
84. D. S. Felsenthal and M. Machover, *Analysis of QM rule adopted by the council of the european union, 23 june 2007*, Tech. report, 2007. [94]
85. R. Fidytek, A. W. Mostowski, R. Somla, and A. Szepietowski, *Algorithms counting monotone boolean functions*, Inf. Process. Lett. **79** (2001), no. 5, 203–209. [16]
86. M. L. Fredman and L. Khachiyan, *On the complexity of dualization of monotone disjunctive normal forms*, J. Algorithms **21** (1996), no. 3, 618–628. [144]
87. J. Freixas, *Bounds for Owen's multilinear extension*, J. Appl. Probab. **44** (2007)), no. 4, 852–864. [66]
88. J. Freixas, X. Molinero, M. Olsen, and M. J. Serna, *The complexity of testing properties of simple games*, CoRR **abs/0803.0404** (2008). [34, 37]
89. J. Freixas and M. A. Puente, *Amplitude of weighted majority systems: strict representations*, Qüestiió **23** (1999), no. 1, 43–60. [109, 110]
90. ______, *A note about games-composition dimension*, Discrete Appl. Math. **113** (2001), no. 2-3, 265–273. [71]
91. J. Freixas and M. A. Puente, *Dimension of complete simple games with minimum*, European Journal of Operational Research **127** (2008), no. 2, 555–568. [37]

92. A. Freville, *The multidimensional 0-1 knapsack problem: An overview*, European Journal of Operational Research **127** (2004), no. 1, 1–21. [138]
93. S. Fujishige, *Submodular functions and optimization*, Annals of Discrete Mathematics, vol. 58, Elsevier, 2005. [201]
94. G. Gambarelli, *Power indices for political and financial decision making: A review*, Annals of Operations Research **51** (1994), 1572–9338. [15, 78, 108]
95. M. R. Garey and D. S. Johnson, *Computers and intractability, a guide to the theory of NP-completeness*, Freeman, 1979. [111]
96. G. Gens and E. Levner, *Complexity of approximation algorithms for combinatorial problems: a survey*, SIGACT News **12** (1980), no. 3, 52–65. [139, 141]
97. A. Gibbard, *Manipulation of voting schemes: A general result*, Econometrica **41** (1973), no. 4, 587–601. [108]
98. D. B. Gillies, *Solutions to general non-zero-sum games*, Contributions to the Theory of Games IV (A. W. Tucker and R. D. Luce, eds.), (Annals of Mathematics Studies), vol. 40, Princeton University Press, 1959, pp. 47–85. [159]
99. C. Godsil and G. Royle, *Algebraic graph theory*, Springer, April 2001. [197]
100. P. W. Goldberg, *A bound on the precision required to estimate a boolean perceptron from its average satisfying assignment*, SIAM J. Discret. Math. **20** (2006), no. 2, 328–343. [48]
101. D. Granot and M. Maschler, *Spanning network games*, International Journal of Game Theory **27** (1998), no. 4, 467–500. [187]
102. D. Granot, M. Maschler, G. Owen, and W. R. Zhu, *The kernel/nucleolus of a standard tree game*, International Journal of Game Theory **25** (1996), no. 2, 219–44. [174, 201]
103. D. Gusfield, *Connectivity and edge-disjoint spanning trees*, Inf. Process. Lett. **16** (1983), no. 2, 87–89. [200, 205]
104. E. Hemaspaandra and L. A. Hemaspaandra, *Dichotomy for voting systems*, J. Comput. Syst. Sci. **73** (2007), no. 1, 73–83. [145]
105. M. Hilliard, *Weighted voting theory and applications*, Tech. Report No. 609, School of Operations Research and Industrial Engineering, Cornell University, 1983. [42]
106. M. J. Holler, *Forming coalitions and measuring voting power*, Political Studies **30** (1982), no. 2, 262–271. [27]
107. R. Holzman, E. Lehrer, and N. Linial, *Some bounds for the Banzhaf index and other semivalues*, Math. Oper. Res. **13** (1988), no. 2, 358–363. [66]
108. E. Horowitz and S. Sahni, *Computing partitions with applications to the knapsack problem*, J. ACM **21** (1974), no. 2, 277–292. [54, 73]
109. S-T. Hu, *Threshold logic*, University of California Press, Berkeley and Los Angeles, 1965. [16, 37, 109, 110]
110. S. Ieong and Y. Shoham, *Bayesian coalitional games*, AAAI, 2008, pp. 95–100. [243]
111. J. Bartholdi III, C. Tovey, and M. Trick., *The computational difficulty of manipulating an election.*, Social Choice and Welfare **6** (1989), no. 3, 227–241. [124]
112. ———, *Voting schemes for which it can be difficult to tell who won the election.*, Social Choice and Welfare **6** (1989), no. 2, 157–165. [124]

113. A. Iwasaki, D. Kempe, Y. Saito, M. Salek, and M. Yokoo, *False-name-proof mechanisms for hiring a team*, WINE, 2007, pp. 245–256. [117]

114. III J. Bartholdi, C. Tovey, and M. Trick., *How hard is it to control an election?*, Mathematical and Computer Modeling **16** (1992), no. 8/9, 27–40. [115, 124]

115. D. Johnson, *Challenges for theoretical computer science*, http://www.research.att.com/~dsj/nsflist.html (2000). [15]

116. G. Kalai and S. Safra, *Threshold phenomena and influence: perspectives from mathematics, computer science, and economics*, Computational complexity and statistical physics, St. Fe Inst. Stud. Sci. Complex., Oxford Univ. Press, New York, 2006, pp. 25–60. MR MR2208732 [14, 33]

117. R. M. Karp, *Combinatorics, complexity, and randomness*, Commun. ACM **29** (1986), no. 2, 98–109. [9]

118. W. Kern and D. Paulusma, *Matching games: The least core and the nucleolus*, Math. Oper. Res. **28** (2003), no. 2, 294–308. [182]

119. J. Kleinberg, *The convergence of social and technological networks*, Commun. ACM **51** (2008), no. 11, 66–72. [11]

120. D. Kleitman, *On Dedekind's problem: The number of monotone boolean functions*, Proceedings of the American Mathematical Society **21** (1969), no. 3, 677–682. [15]

121. B. Klinz and G. J. Woeginger, *Faster algorithms for computing power indices in weighted voting games*, Mathematical Social Sciences **49** (2005), no. 1, 111–116. [54, 67, 73]

122. A. Kopelowitz, *Computation of the kernels of simple games and the nucleolus of n-person games*, Math. Dept., The Hebrew University of Jerusalem (1967), no. RM-31. [174]

123. B. Korte and J. Vygen, *Combinatorial optimization: Theory and algorithms*, 3rd ed., Springer, Germany, 2006. [50, 191]

124. I. Krohn and P. Sudholter, *Directed and weighted majority games*, Mathematical Methods of Operations Research **42** (1995), no. 2, 189–216. [180]

125. J. Kuipers, *A polynomial algorithm for computing the nucleolus of convex games.*, Tech. rep., Univ. of Maastricht **M 96-12** (1996). [174, 201]

126. K. Jeroen Kuipers, U. Faigle, and W. Kern, *On the computation of the nucleolus of a cooperative game*, International Journal of Game Theory **30** (2001), no. 1, 79–98. [174, 201]

127. D. M. Kilgour L. Fang, K. W. Hipel, *Interactive decision making: The graph model for conflict resolution (wiley series in systems engineering and management)*, John Wiley & Sons, New York, 1993. [12]

128. Lamsade and Dimacs, *Dimacs-Lamsade partnership*, 2008, `http://www.lamsade.dauphine.fr/dimacs/`. [14]

129. J-E Lane and R. Maeland, *Constitutional analysis: The power index approach*, European Journal of Political Research **37** (2000), 31–56. [78]

130. A. Laruelle and M. Widgren, *Is the allocation of voting power among EU states fair?*, Public Choice **94** (1998), no. 3-4, 317–39. [78, 79, 108]

131. D. Leech, *Computation of power indices*, Warwick Economic Resarch Papers (2002), no. 644. [66]

132. ———, *An empirical comparison of the performance of classical power indices*, Political Studies **50** (2002), no. 1, 1–22. [53]

133. ______, *Designing the Voting System for the Council of the European Union*, Public Choice **113** (2002), no. 3-4, 437–64. [89]
134. ______, *Voting Power in the Governance of the International Monetary Fund*, Annals of Operations Research **109** (2002), no. 1, 375–397. [78, 108]
135. ______, *Computing power indices for large voting games*, Management Science **49** (2003), no. 6, 831–837. [66, 80]
136. ______, *Power indices as an aid to institutional design: the generalised apportionment problem*, Yearbook on New Political Economy, edited by M. Holler, H.Kliemt, D. Schmidtchen and M. Streit. (2003). [79]
137. D. Leech and H. Aziz, *The double majority voting rule of the EU reform treaty as a democratic ideal for an enlarging union: an appraisal using voting power analysis*, W. Slomczynski and K. Zyczkowski eds., Distribution of Power and voting procedures in the European Union, Ashgate, 2009. [XVII, 19]
138. N. Linial, *Game-theoretic aspects of computing*, Handbook of Game Theory with Economic Applications (R. J. Aumann and S. Hart, eds.). [10]
139. F. W. Lucas, *Von Neumann-Morgenstern Stable Sets*, Handbook of Game Theory **1** (1962), 543–590. [160]
140. ______, *The proof that a game may not have a solution*, Transactions of the American Mathematical Society **136** (1969), no. 2, 219–229. [160]
141. W. F. Lucas, *Von neumann-morgenstern stable sets*, Handbook of Game Theory with Economic Applications (R. J. Aumann and S. Hart, eds.). [159]
142. Maschler M., B. Peleg, and L. S. Shapley, *Geometric properties of the kernel, nucleolus, and related solution concepts*, Math. Oper. Res. **4** (1979), 303–338. [159, 177]
143. H. S. Wilf M. Petkovsek and D. Zeilberger, $A = b$, A K Peters, 1997. [73]
144. M. Machover and D. S. Felsenthal, *The treaty of nice and qualified majority voting*, Social Choice and Welfare **18** (2001), no. 3, 431–464. [89]
145. ______, *The Treaty of Nice and qualified majority voting*, Social Choice and Welfare **18** (2001), no. 3, 431–464. [89]
146. ______, *Annexations and alliances: When are blocs advantageous a priori?*, Social Choice and Welfare **19** (2002), no. 2, 295–312. [118]
147. K. Makino, *A linear time algorithm for recognizing regular boolean functions*, J. Algorithms **43** (2002), no. 2, 155–176. [41, 43]
148. I. Mann and L. S Shapley, *Values of Large Games IV: Evaluating the electoral college by montecarlo techniques*, The Rand Corporation, Santa Monica. **RM-2651**. [65, 66]
149. M. Maschler, *The bargaining set, kernel, and nucleolus*, Handbook of Game Theory with Economic Applications (R. J. Aumann and S. Hart, eds.). [159]
150. M. Maschler and B. Peleg, *A characterization, existence proof and dimension bounds for the kernel of a game.*, Pacific J. Math **18** (1966), no. 2, 289–328. [35]
151. T. Matsui and Y. Matsui, *A survey of algorithms for calculating power indices of weighted majority games*, Journal of the Operations Research Society of Japan **43** (2000), no. 7186. [38, 40, 46, 54, 64, 66, 67, 73, 124, 144]
152. Y. Matsui and T. Matsui, *NP-completeness for calculating power indices of weighted majority games*, Theoretical Computer Science **263** (2001), no. 1-2, 305–310. [78]

153. H. Meinhardt, *An LP approach to compute the pre-kernel for cooperative games*, Comput. Oper. Res. **33** (2006), no. 2, 535–557. [180]

154. S. Muroga, *Threshold logic and its applications*, Wiley Interscience, New York, 1971. [16, 42]

155. H. Nagamochi, D-Z Zeng, N. Kabutoya, and T. Ibaraki, *Complexity of the minimum base game on matroids*, Math. Oper. Res. **22** (1997), no. 1, 146–164. [187]

156. H. Narayanan, *Theory of matroids and network analysis*, Ph.D. thesis, Indian Institute of Technology, Bombay, 1974. [201]

157. C. St.J. A. Nash-Williams, *Edge-disjoint spanning trees of finite graphs*, J. London Math. Soc. **36** (1961), 445–450. [201]

158. N. Nisan, *Introduction to mechanism design*, Algorithmic Game Theory (N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, eds.), Cambridge University Press, 2007, pp. 209–242. [10]

159. L. Nordmann and H. Pham, *Weighted voting systems*, IEEE Transactions on Reliability **48** (1999), no. 1, 42–49. [108]

160. H. Nurmi, *On the strategic properties of some modern methods of group decision making*, Behavioral Science **29** (1984), no. 4, 248–257. [107]

161. N. Ohta, V. Conitzer, Y. Satoh, A. Iwasaki, and M. Yokoo, *Anonymity-proof Shapley value: extending Shapley value for coalitional games in open environments*, AAMAS, 2008, pp. 927–934. [117]

162. N. Ohta, A. Iwasaki, M. Yokoo, K. Maruono, V. Conitzer, and T. Sandholm, *A compact representation scheme for coalitional games in open anonymous environments*, AAAI, 2006. [117]

163. M. J. Osborne and A. Rubinstein, *A course in game theory*, first edition ed., The MIT Press, Massachusetts, USA, 1994. [155, 158, 165]

164. G. Owen, *Multilinear extensions of games*, Management Science **18** (1972), no. 5, 64–79. [65]

165. G. Owen, *Multilinear extensions and theBanzhaf value*, Naval Research Logistics Quarterly **22** (1975), no. 4, 741–750. [65]

166. C. H. Papadimitriou, *Computational Complexity*, Addison-Wesley, Reading, Massachusetts, 1994. [136]

167. ______, *Algorithms, games, and the internet*, STOC '01: Proceedings of the Thirty-Third annual ACM Symposium on Theory of Computing (New York, NY, USA), ACM, 2001, pp. 749–753. [10, 185]

168. ______, *The search for equilibrium concepts*, SAGT, 2008, pp. 1–3. [13]

169. C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization; algorithms and complexity*, Dover Publications, 1998. [17]

170. C. H. Papadimitriou and M. Yannakakis, *On complexity as bounded rationality (extended abstract)*, STOC '94: Proceedings of the Twenty-Sixth Annual ACM symposium on Theory of Computing (New York, NY, USA), ACM, 1994, pp. 726–733. [124]

171. B. Parhami, *Voting: A paradigm for adjudication and data fusion in dependable systems*, Dependable Computing Systems: Paradigms, Performance Issues, & Applications (H. B. Diab and A. Y. Zomaya, eds.), Wiley, 2005, pp. 87–114. [16, 108]

172. S. Patkar and H. Narayanan, *Fast on-line/off-line algorithms for optimal reinforcement of a network and its connections with principal partition*, FST TCS 2000 (London, UK), Springer-Verlag, 2000, pp. 94–105. [201]

173. D. Paulusma, *Complexity aspects of cooperative games (PhD-thesis)*, University of Twente, 2001. [17]

174. U. N. Peled and B. Simeone, *Polynomial-time algorithms for regular set-covering and threshold synthesis*, Discrete Appl. Math **12** (1985), 57–69. [37, 41]
175. B. Peleg, *On weights of constant-sum majority games*, SIAM Journal on Applied Mathematics **16** (1968), no. 3, 527–532. [175]
176. B. Peleg and P. Sudholter, *Introduction to the theory of cooperative games*, first edition ed., Kluwer Academic, Boston, 2003. [155]
177. L. S. Penrose, *The elementary statistics of majority voting*, Journal of Royal Statistical Society **109** (1946), 40–63. [82, 91]
178. ———, *On the objective study of crowd behavior*, H. K. Lewis and Co., London, 1952. [91]
179. K. Prasad and J. S. Kelly, *NP-completeness of some problems concerning voting games*, Int. J. Game Theory **19** (1990), no. 1, 1–9. [46, 54, 116]
180. A. D. Procaccia and J. S. Rosenschein, *Junta distributions and the average-case complexity of manipulating elections*, AAMAS '06: Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (New York, NY, USA), ACM, 2006, pp. 497–504. [145]
181. J. S. Provan and M. O. Ball, *The complexity of counting cuts and of computing the probability that a graph is connected*, SIAM J. Comput. **12** (1983), no. 4, 777–788. [47]
182. Ryan R. O'Donnell and R. A. Servedio, *The Chow parameters problem*, STOC '08: Proceedings of the 40th annual ACM symposium on Theory of computing (New York, NY, USA), ACM, 2008, pp. 517–526. [84]
183. T. E. S. Raghavan and T. Solymosi, *Assignment games with stable core*, International Journal of Game Theory **30** (2001), no. 2, 177–185. [166]
184. K. G. Ramamurthy, *Coherent structures and simple games (theory and decision library)*, first edition ed., Kluwer Academic Publishers, Netherlands, 1990. [108, 137]
185. H. Reijnierse and J. Potters, *The β-nucleolus of TU-games*, Games and Economic Behavior **24** (1998), no. 1-2, 77–96. [177]
186. F. S. Roberts, *Computer science and decision theory*, Annals of Operations Research **163** (2008), no. 1, 209–253. [9, 12]
187. J. S. Rosenschein and A. D. Procaccia, *Voting in cooperative information agent scenarios: Use and abuse*, Cooperative Information Agents X, The Tenth International Workshop on Cooperative Information Agents, Edinburgh (CIA 2006) (M. Klusch, M. Rovatsos, and T. R. Payne, eds.), Lecture Notes in Artificial Intelligence, no. 4149, Springer-Verlag, Berlin, September 2006, pp. 33–50. [10]
188. A. E. Roth, *The economist as engineer: Game theory, experimentation, and computation as tools for design economics*, Econometrica **70** (2002), no. 4, 1341–1378. [11]
189. D. Schmeidler, *The nucleolus of a characteristic function game*, SIAM J. Appl. Math. **17** (1969), no. 6, 1163–1170. [159, 160, 173]
190. A. Schrijver, *Combinatorial optimization : Polyhedra and efficiency (algorithms and combinatorics)*, Springer, 2004. [202, 203]
191. L. S. Shapley, *Cores of convex games*, Journal International Journal of Game Theory **1** (1971), no. 1, 11–26. [167]
192. ———, *A value for n person games*, A. E. Roth, editor, The Shapley value (1988), 31–40. [25]
193. L. S. Shapley and M. Shubik, *Quasi-cores in a monetary economy with non-convex preferences*, Econometrica **34** (1966), 805–827. [159]

194. Y. Shoham, *Computer science and game theory*, Communications of the ACM **51** (2008), no. 5, 74–79. [14]

195. Y. Shoham and K. Leyton-Brown, *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*, Cambridge University Press, 2008. [13]

196. M. Shubik, *Game theory in the social sciences: Concepts and solutions*, MIT Press. [155]

197. W. Slomczynski and K. Zyczkowski, *From a toy model to the double square root voting system*, ArXiv Physics e-prints (2007). [83, 84, 91, 94]

198. T. Solymosi and T. E. S. Raghavan, *An algorithm for finding the nucleolus of assignment games*, Int. J. Game Theory **23** (1994), no. 2, 119–143. [174, 201]

199. S. H. Strogatz, *Exploring complex networks.*, Nature **410** (2001), no. 6825, 268–276. [185]

200. X. Sun and Q. Fang, *Core stability of flow games*, China-Japan Conference on Discrete Geometry, Combinatorics and Graph Theory, 2005, pp. 189–199. [166]

201. K. Takamiya and A. Tanaka, *Computational complexity in the design of voting games*, The Institute of Social and Economic Research, Osaka University **653** (2006). [80]

202. A. Taylor and W. Zwicker, *Simple games: Desirability relations, trading, pseudoweightings*, first edition ed., Princeton University Press, New Jersey, 1999. [14, 15, 16, 36, 37, 71, 108]

203. M. Tennenholtz, *Game theory and artificial intelligence*, Selected papers from the UKMAS Workshop on Foundations and Applications of Multi-Agent Systems (London, UK), Springer-Verlag, 2002, pp. 49–58. [10]

204. J. Tolle, *Power distribution in four-player weighted voting systems*, Mathematics Magazine **76** (2003), no. 1, 33–39. [79]

205. N. Tomizawa, *Strongly irreducible matroids and principal partition of a matroid into strongly irreducible minors*, Electron. and Commun. Japan **59** (1976), no. A, 1–10. [201]

206. V. A. Trubin, *Strength of a graph and packing of trees and branchings*, Cybernetics and Systems Analysis **29** (1993), no. 3, 379–384. [219]

207. W. T. Tutte, *On the problem of decomposing a graph into n connected factors*, J. London Math. Soc. **36** (1961), 221–230. [201]

208. A. B. Urken, *Social choice theory and distributed decision making*, Proceedings of the ACM SIGOIS and IEEECS TC-OA 1988 Conference on Office Information Systems (New York, NY, USA), ACM Press, 1988, pp. 158–168. [10, 16, 108]

209. L. G. Valiant, *The complexity of enumeration and reliability problems*, SIAM Journal on Computing **8** (1979), no. 3, 410–421. [187]

210. A. van den Nouweland, S. Tijs, and M. Maschler, *Monotonic games are spanning network games*, International Journal of Game Theory **21** (1993), no. 4, 419–27. [187]

211. van der Laan G., *A Banzhaf share function for cooperative games in coalition structure*, Theory and Decision **53** (August 2002), 61–86. [118]

212. J. von Neumann, *Probabilistic logics and synthesis of reliable organisms from unreliable components*, Automata Studies (C. Shannon and J. McCarthy, eds.), Princeton University Press, 1956, pp. 43–98. [16, 108]

213. J. von Neumann and O. Morgenstern, *Theory of games and economic behavior*, Princeton University Press, 1944. [15, 34, 159, 160]

214. H. Wallace and F. Hayes-Renshaw, *The council of ministers*, Mathematics Department, Princeton University, Basingstoke: Palgrave Macmillan, 2006. [90]

215. M. P. Wellman, *The economic approach to artificial intelligence*, ACM Comput. Surv. **27** (1995), no. 3, 360–362. [11]

216. R. O. Winder, *Threshold Logic, Ph.D. thesis*, Mathematics Department, Princeton University, New Jersey, 1962. [16, 42]

217. L. A. Wolsey, *The nucleolus and kernel for simple games or special valid inequalities for 0-1 linear integer programs*, International Journal of Game Theory **5** (1976), no. 4, 227–238. [179, 180]

218. M. Yokoo, *False-name bids in combinatorial auctions*, SIGecom Exch. **7** (2007), no. 1, 48–51. [117]

219. M. Yokoo, V. Conitzer, T. Sandholm, N. Ohta, and A. Iwasaki, *Coalitional games in open anonymous environments*, AAAI, 2005, pp. 509–515. [117]

220. M. Zuckerman, P. Faliszewski, Y. Bachrach, and E. Elkind, *Manipulating the quota in weighted voting games*, The Twenty-Third National Conference on Artificial Intelligence (Chicago, Illinois), July 2008. [108, 117]

Index