On the Computation of the Nucleolus of Cooperative Transferable Utility Games

PhD Thesis

by

Balázs Sziklai

Supervisor:
Dr. Tamás Fleiner
Associate Professor, Budapest University of Technology and Economics

Eötvös Loránd University
Doctoral School in Mathematics

Director of the Doctoral School:
Dr. Miklós Laczkovich
Professor, Member of the Hungarian Academy of Sciences

Doctoral Program in Applied Mathematics

Director of the Program:
Dr. György Michaletzky
Professor, Doctor of the Hungarian Academy of Sciences

Operations Research Department
Eötvös Loránd University
Budapest

2015
Declaration of Authorship

I, BALÁZS SZIKLAI, declare that this thesis titled, ‘On the Computation of the Nucleolus of Cooperative Transferable Utility Games’ and the work presented in it are my own. I confirm that:

- This work was done wholly while in candidature for a research degree at this University.
- No part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.

Signed:  

Date:
I would like to express my gratitude to my supervisor Tamás Fleiner for his support and patience. He encouraged me even when the pieces of the proofs did not seem to fit together. He taught me to use mathematical rigor, even when proving trivialities - especially then.

I would like to thank all my colleagues at the Centre for Economic and Regional Studies. I am indebted to László Á. Kóczy who admitted me into the Game Theory Research Group. I thank Tamás Solymosi for lending me his vast knowledge about cooperative games.

I am indebted to many other people for their helpful suggestions that improved this thesis. I am especially thankful to Tamás Kis and Miklós Pintér. I am also grateful to Balázs Kovács, a friend and latex magician, who read a preliminary version of this work, without whom this thesis would contain significantly less commas.

Finally, I would like to express my deepest gratitude to my family. I cannot possibly thank enough the endless support of my parents. I am thankful to my wife who comforted me whenever one of my carefully constructed proofs collapsed and gave me strength to correct it. I dedicate this thesis to my son, Benjamin.
# Contents

Declaration of Authorship ............................ i  
Acknowledgements ................................... ii  
Abbreviations ....................................... v  
Symbols ............................................... vii  

## 1 Introduction  
1.1 Cooperative games ................................ 3  
1.2 Computational complexity .......................... 4

## 2 Game theoretical framework  
2.1 Properties of the characteristic function .......... 7  
2.2 Properties of solutions ................................ 9  
2.3 Shapley-value ........................................ 13  
2.4 Nucleolus ............................................. 14  
2.5 Solution concepts related to the nucleolus .......... 16  
2.6 Cost allocation games .................................. 19  
2.7 Partition function form games ........................ 20

## 3 Computing the nucleolus  
3.1 The nucleolus by linear programming ................. 23  
3.2 Verifying the nucleolus ................................ 25  
3.3 Characterization sets for the nucleolus ............... 28

## 4 Bankruptcy games  
4.1 A riddle in the Talmud .............................. 39  
4.2 Formal definition of bankruptcy games ............... 40  
4.3 Literature overview ..................................... 42  
4.4 Hydraulic rationing ..................................... 44  
4.5 The nucleolus with characterization sets ............... 53

## 5 Directed acyclic graph games  
5.1 Cost sharing on rooted graphs ........................ 59
Contents

5.2 Literature overview ........................................... 61
5.3 Formal definition of directed acyclic graph games ............ 64
5.4 Significance of DAG-games ................................... 67
5.5 The canonization process and its consequences ................. 71
5.6 The core of the canonized DAG-game .......................... 76
5.7 Dually essential coalitions of a DAG-game .................... 88
5.8 The nucleolus with characterization sets ...................... 93
5.9 The road construction algorithm .............................. 96
  5.9.1 The simplified version ................................... 96
  5.9.2 The proof ............................................... 98
  5.9.3 Extending the algorithm ................................. 102
  5.9.4 Applicability of the algorithm .......................... 105
5.10 Time complexity issues ..................................... 106

6 Conclusion .................................................. 109
  6.1 How to find the nucleolus .................................. 109
  6.2 Characterization sets ....................................... 111
  6.3 Bankruptcy games ......................................... 112
  6.4 Directed acyclic graph games .............................. 113
  6.5 Future research ........................................... 115

References .................................................. 121

Index ...................................................... 133
Abbreviations

ADD additivity
AN anonymous
AMON aggregate monotonicity
CEA constrained equal awards rule
CEL constrained equal losses rule
CG contested garment
COV covariance under strategic equivalence
DAG directed acyclic graph
DP dummy player
ISRGP imputation saving reduced game property
ETP equal treatment property
LP linear program
MCST minimum cost spanning tree
mMCST monotonic minimum cost spanning tree
NE non-emptiness
PFF partition function form
PO Pareto optimal
poset partially ordered set
RGP reduced game property
RC road construction
SMON strong monotonicity
TU transferably utility
## Symbols

### General

- $2^N$: power set of $N$  
- $|f(n)|$: absolute value of $f(n)$  
- $f(n) = O(g(n))$: $|f(n)| \leq c \cdot |g(n)|$ for $n$ sufficiently large  
- $N$: set of players / the grand coalition  
- $\mathbb{N}$: set of natural numbers  
- $\mathbb{R}$: set of real numbers  
- $\mathbb{R}^+$: set of positive real numbers  
- $\mathbb{R}_0^+$: set of non-negative real numbers  
- $\mathbb{R}^n$: $n$-dimensional vectors with real components  
- $\mathbb{R}^N$: Real vector space corresponding to the player set $N$  
- $|S|$: cardinality of $S$  
- $S \cup i$: same as $S \cup \{i\}$  
- $S \setminus i$: same as $S \setminus \{i\}$  
- $v(i)$: same as $v(\{i\})$  
- $x(i)$: same as $x(\{i\})$  
- $x(S)$: $\sum_{i \in S} x_i = \sum_{i \in S} x(i)$  
- $x \leq y$: $x_i \leq y_i$ for $i = 1, \ldots, n$  
- $x \preceq y$: $x$ is lexicographically smaller than $y$

### Game Theory

- $c(S)$: characteristic value of coalition $S$ in a cost game  
- $\mathcal{C}(\Gamma)$: core of game $\Gamma$

vii
Symbols

\( \Gamma \) shorthand for either \((N,v)\) or \((N,c)\)  
\((N,v)\) cooperative transferable utility game / profit game  
\( I(N,v) \) imputation set  
\( I^*(N,v) \) set of allocations  
\( \mathcal{N}(\Gamma) \) nucleolus of game \( \Gamma \)  
\( \mathcal{P} \) proper (non-trivial coalitions) \( \{S \in 2^N \mid S \neq 0, N\} \)  
\( sat_\Gamma(S,x) \) satisfaction of coalition \( S \) under payoff vector \( x \) in game \( \Gamma \)  
\( v(S) \) characteristic value of coalition \( S \) in a profit game

Chapter 2

\( C_\varepsilon(\Gamma) \) \( \varepsilon \)-core of game \( \Gamma \)  
\( \mathcal{G} \) a subset of \( \mathcal{V}^N \)  
\( i \sim_v j \) equivalence of player \( i \) and \( j \) in game \( v \)  
\( \mathcal{LC}(\Gamma) \) least core of game \( \Gamma \)  
\( m_i(S,v) \) marginal contribution of player \( i \) to coalition \( S \) in game \( v \)  
\( \mathcal{N}(\Gamma,X) \) nucleolus in game \( \Gamma \) with respect to \( X \)  
\( \mathcal{N}(\Gamma) \) shorthand for \( \mathcal{N}(\Gamma,I(N,v)) \)  
\( \pi \) a permutation  
\( \Phi(v) \) Shapley-value in game \( v \)  
\( \mathbb{R}^v(\mathcal{N}) \) a real vector space with permuted coordinates  
\( \sigma \) a (possibly set-valued) function on the domain \( \mathcal{G} \)  
\( \theta(x) \) satisfaction vector under payoff vector \( x \)  
\( v(S,x) \) reduced game with respect to coalition \( S \) and payoff vector \( x \)  
\( \nu(S,x) \) imputation saving reduced game  
\( V \) partition function  
\( \mathcal{V}^N \) set of all coalition functions on \( 2^N \)

Chapter 3

\( B \) balanced set of coalitions  
\( B_0 \) coalitions with the smallest satisfaction values  
\( B_S \) \( S \)-balanced collection

viii
Symbols

\[ \mathcal{DE}(\Gamma) \] set of dually essential coalitions 31
\[ \mathcal{DS}(\Gamma) \] set of dually saturated coalitions 34
\[ \mathcal{E}(\Gamma) \] set of essential coalitions 30
\[ e_S \] indicator vector of coalition \( S \) 25
\[ (N, v^*) \] dual game given by \( v^*(S) := v(N) - v(N \setminus S) \) for all \( S \subseteq N \).
\[ \mathcal{S}(\Gamma) \] set of saturated coalitions 33
\[ \underline{S} \] lower closure of \( S \) 34

Chapter 4

\( \mathbb{B} \) set of all bankruptcy problems 40
\( d \) claims vector 40
\( (d, E) \) bankruptcy problem 40
\( d(S) \) the total amount claimed by the members of \( S \) 40
\( E \) estate/endowment 40
\( \mathcal{H} \) hydraulic system 45
\( r \) bankruptcy rule 40
\( r^* \) dual of \( r \) 41
\( v_{(d,E)} \) bankruptcy game 40
\( t \) common water level in the connected talmudic hydraulic 49
\( S_{j \bar{i}} \) set of coalitions that contain \( j \) but not \( i \) 49

Chapter 5

\( (*) \) restrictive condition on player networks 67
\( \prec \) partial order defined on the node set 74
\( A \) arc set 64
\( A_p \) arcs that leave node \( p \) 65
\( A(T) \) arc set of subgraph \( T \) 65
\( a_p \) an arc that leaves node \( p \) 76
\( a_s \) a shortcut 86
\( B \) a specific node set, usually a branch 65
\( B_p \) full branch originating from node \( p \) 74
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_p^Q$</td>
<td>a specific branch</td>
</tr>
<tr>
<td>$C(T)$</td>
<td>total cost of the arcs in $T$</td>
</tr>
<tr>
<td>$c_{(D,R)}(S)$</td>
<td>characteristic value of $S$ defined on $(D,R)$</td>
</tr>
<tr>
<td>$c_D(S)$</td>
<td>same as $c_{(D,R)}$</td>
</tr>
<tr>
<td>$D$</td>
<td>directed acyclic graph network</td>
</tr>
<tr>
<td>$\bar{D}$</td>
<td>the DAG-network that is generated after $\mathcal{P}_1$</td>
</tr>
<tr>
<td>$(D,R)$</td>
<td>player network</td>
</tr>
<tr>
<td>$d(p)$</td>
<td>length of the shortest path in $T_N$ from $p$ to $r$</td>
</tr>
<tr>
<td>$\delta(a)$</td>
<td>construction cost of arc $a$</td>
</tr>
<tr>
<td>$F$</td>
<td>set of free nodes</td>
</tr>
<tr>
<td>$G(V,A)$</td>
<td>directed acyclic graph</td>
</tr>
<tr>
<td>$\bar{\Gamma}$</td>
<td>shorthand for $(N,c_D)$</td>
</tr>
<tr>
<td>$N(T)$</td>
<td>residents of subgraph $T$</td>
</tr>
<tr>
<td>$p \prec q$</td>
<td>node $p$ is an ancestor of node $q$</td>
</tr>
<tr>
<td>$P_{q-p}$</td>
<td>a path in $T_N$ that leads from $q$ to $p$</td>
</tr>
<tr>
<td>$\mathcal{P}_1$</td>
<td>the first cycle of the road construction algorithm</td>
</tr>
<tr>
<td>$\pi(p)$</td>
<td>the parent (direct ancestor) of $p$</td>
</tr>
<tr>
<td>$\Pi(p)$</td>
<td>the principal ancestor of $p$</td>
</tr>
<tr>
<td>$r$</td>
<td>the root (sink) of the graph</td>
</tr>
<tr>
<td>$\mathcal{R}$</td>
<td>mapping from the player set to the node set</td>
</tr>
<tr>
<td>$\mathcal{R}(S)$</td>
<td>${\mathcal{R}(i) : i \in S}$</td>
</tr>
<tr>
<td>$\mathcal{S}_0(\Gamma)$</td>
<td>saturated coalitions that have zero satisfaction in the core</td>
</tr>
<tr>
<td>$t_1$</td>
<td>time spent with construction during $\mathcal{P}_1$</td>
</tr>
<tr>
<td>$T_S$</td>
<td>the trunk constructed by coalition $S$</td>
</tr>
<tr>
<td>$\tau(Q,S)$</td>
<td>cost of the arcs in $T_S$ that leave from node set $Q$</td>
</tr>
<tr>
<td>$V$</td>
<td>node set of the graph</td>
</tr>
<tr>
<td>$V(T)$</td>
<td>node set of subgraph $T$</td>
</tr>
<tr>
<td>$w(a)$</td>
<td>workers assigned to arc $a$</td>
</tr>
<tr>
<td>$\hat{x}$</td>
<td>standard allocation</td>
</tr>
<tr>
<td>$z$</td>
<td>outcome of the road construction algorithm</td>
</tr>
<tr>
<td>$\hat{z}$</td>
<td>outcome of the simplified road construction algorithm</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

"Science talks about very simple things, and asks hard questions about them. As soon as things become too complex, science can’t deal with them... But it’s a complicated matter: Science studies what’s at the edge of understanding, and what’s at the edge of understanding is usually fairly simple. And it rarely reaches human affairs. Human affairs are way too complicated."

Avram Noam Chomsky

Fair distribution is one of the oldest problems of society. The earliest known code of laws, the Code of Hammurabi as well as antique religious texts such as the Talmud devote long passages on fair distribution (Hammurabi and Johns, 2000; Elishakoff, 2011). As the quote from Chomsky points out, it is very hard to seize such a human concept as 'fairness' and especially hard to translate it to the language of mathematics. Axiomatic economics bypasses this problem. It does not elect a single solution as universally good, instead it searches for desirable properties (such as efficiency, envy-freeness, etc.) and investigates which property is met by which solution. Then it leaves the task of choosing the appropriate solution to us.

Cooperative game theory offers one way to deal with fair distribution problems where participants (i.e. the players) can interact with each other by means of forming coalitions. In such cases stability of the solution is always a crucial question. The Shapley-value and the nucleolus are the two most popular solution concepts that game theorist (re-)invented. Interestingly enough both solution concepts were
used prior to their 'invention' in various fields such as agriculture (Aadland and Koplin, 1998) and religious law (Aumann and Maschler, 1985). This however does not decrease the value of their discovery. On the contrary! It shows that both of these solutions capture something fundamental and despite how complicated they look, people like to use them.

The nucleolus was developed by Schmeidler (1969) and became one of the most applied solution concepts of cooperative game theory. Despite its good properties it lost some popularity in the last 20 or so years. The general opinion shifted partly because it was categorized to be a too complicated solution concept. It is not only difficult to compute, but it is also hard to interpret, since its axiomatization is less straightforward as for instance the Shapley value's. Not to mention that the latter one has many different axiomatizations while the nucleolus has basically just one. Even verifying whether an allocation is the nucleolus or not is conjectured to be \( NP \)-hard\(^2\) in general (Faigle, Kern, and Kuipers, 1998).

Ironically in this period there has been several breakthroughs regarding the computational aspect of the nucleolus that somehow escaped the attention of many theoreticians and researchers. The aim of this thesis is threefold.

- To aggregate the known results related to the computation of the nucleolus.
- To analyze the various existing methods in practice.
- To expand the theory of characterization sets and show their usefulness.

We begin with the basic terminology of game theory and computer science. In the second chapter we spend some time with the properties of the characteristic function, we introduce the Shapley-value and the nucleolus formally and describe some axioms. The third chapter elaborates on the computation of the nucleolus in general. In the fourth chapter we examine the nucleolus of bankruptcy games – an important class of games that has already been extensively studied. We will also introduce directed acyclic graph (DAG) games, a generalization of standard tree games that is another popular fair distribution problem. We analyze the core of DAG-games and provide an algorithm that computes the nucleolus for some

\(^1\) Another famous example from game theory is the Gale-Shapley algorithm that was used in the US National Resident Matching Program before Gale and Shapley published their celebrated paper.

\(^2\) This concept is defined formally in Section 1.2.
families of this game class. In the last chapter we summarize the results, discuss the limitations of our work and point out interesting research problems. In this last chapter we also make an inventory of the author’s contributions.

1.1 Cooperative games

Game theory studies strategic decision making when players have interest of conflict and/or can create value by cooperating with each other. As this vague definition suggests game theory is a vast field. Traditionally game theory is divided into two branches, called the cooperative and non-cooperative branch. These two branches differ in how they formalize interdependence among the players.

In cooperative games the outcome is uniquely determined by how the players team up with each other, individual strategy choices are not visible. A common assumption is that the utilities of different coalitions can be compared with each other. For example, the value of different coalitions can be expressed in terms of money, and the preference for money is the same for every player. In such cases we speak of transferable utility or simply TU-games.

A cooperative game with transferable utility is an ordered pair \((N, v)\) consisting of the player set \(N = \{1, 2, \ldots, n\}\) and a characteristic function \(v : 2^N \to \mathbb{R}\) with \(v(\emptyset) = 0\). The value \(v(S)\) represents the worth of coalition \(S\). To put it differently, no matter how other players behave if the players of \(S\) work together they can secure themselves \(v(S)\) amount of payoff. The set \(N\) – when viewed as a coalition – is called the grand coalition.

Definition 1.1. A cooperative game \((N, v)\) is called superadditive if

\[
(S, T \subset N, S \cap T = \emptyset) \Rightarrow v(S) + v(T) \leq v(S \cup T).
\]

In superadditive games two disjoint coalitions can always merge without losing money. Hence we can assume that players form the grand coalition. The main question is then how to distribute \(v(N)\) among the players in some fair way. A solution is a vector \(x \in \mathbb{R}^N\) that represents the payoff of each player. The next chapter is dedicated solely to the formal introduction of the properties and solution concepts of cooperative games. For more on the subject the reader is referred to Peleg and Sudhölter (2007).


1.2 Computational complexity

As for an \( n \) person game there are \( 2^n \) number of possible coalitions, a cooperative game theoretical solution might be hard to compute. Our main goal is to show different ways how the nucleolus of various classes of games can be calculated efficiently in polynomial time. For this we will need some basic notions from computer science.

**Definition 1.2.** *(Big O notation)* Let \( f \) and \( g \) be two functions \( f, g : \mathbb{N} \rightarrow \mathbb{R}^+ \). We say that \( f(n) = O(g(n)) \) if positive integers \( C \) and \( n_0 \) exists so that for every integer \( n > n_0 \)

\[
|f(n)| \leq C|g(n)|.
\]

Intuitively this means that \( f \) does not grow faster than \( g \). That is, with the big O notation we can express the order of magnitude of a function.

*Time complexity* or *running time* of an algorithm quantifies the amount of time taken by an algorithm to run as a function of the length of the string representing the input (Sipser, 1997). The running time is not measured in units of time, but rather in how many elementary operations (addition, multiplication, comparison, etc.) the algorithm needs, to solve a problem instance. Time complexity of an algorithm is commonly expressed using big O notation. Since an algorithm’s performance may vary with different inputs of the same size, it is customary to use the worst-case time complexity of an algorithm, denoted by \( T(n) \), which is defined as the maximum amount of time taken on any input of size \( n \). An algorithm is said to be of polynomial time if its running time is upper bounded by a polynomial expression in the size of the input for the algorithm, i.e. \( T(n) = O(n^k) \) for some constant \( k \). Similarly an algorithm is said to be exponential time, if \( T(n) \) is upper bounded by \( 2^{poly(n)} \), where \( poly(n) \) is some polynomial in \( n \), more formally if \( T(n) = O(2^{n^k}) \) for some constant \( k \).

A *decision problem* is a question in a formal system that can be answered only by "yes" or "no". We say that a decision problem \( D \) is solvable if there exists an algorithm that eventually comes up with an answer upon receiving \( D \) as an input. The complexity class \( EXP \) contains all the decision problems that are solvable in exponential time while \( P \) gathers those that are solvable in polynomial
1.2 Computational complexity

time\(^3\). Problems in \(P\) are considered to be 'easy' in comparison with problems in \(\text{EXP} \setminus P\), as solving large instances from the latter class is computationally infeasible.

Another important complexity class is \(NP\) that collects those decision problems that are verifiable in polynomial time. That is, if the answer is "yes" for the decision problem then there exist a proof that supports this claim which can be checked in polynomial time. For instance, for a given graph \(G\) the question of "Does \(G\) have a Hamiltonian cycle (a cycle that contains every node of \(G\) exactly once)?" belongs to \(NP\). Obviously \(P \subseteq NP\). Whether \(P = NP\) or not is considered by many to be the most important open problem of computer science. It is one of the seven Millennium Prize Problems selected by the Clay Mathematics Institute to carry a US$ 1,000,000 prize for the first correct solution. It is widely assumed that actually \(P \neq NP\) is true.

An important subclass of \(NP\) is called \(NP\)-complete. A decision problem \(D\) is said to be \(NP\)-complete if \(D\) is in \(NP\) and every other problem in \(NP\) is reducible to \(D\) in polynomial time. Intuitively the \(NP\)-complete problems are the hardest in \(NP\) as if there is a polynomial time algorithm for any \(NP\)-complete problem it follows immediately that \(P = NP\).

Finally, an optimization problem is called \(NP\)-hard if there exists an \(NP\)-complete decision problem that can be polynomially reduced to it. An example for an \(NP\)-hard optimization problem is the Steiner tree problem. Given an undirected graph \(G = (V, E)\) with non-negative edge weights and a subset \(R \subset V\) of required vertices the task is to find the cheapest tree that spans all the nodes in \(R\). The corresponding \(NP\)-complete decision problem is to determine for a fixed \(k\) if there exist a Steiner tree of total weight at most \(k\).

The book of Chalkiadakis, Elkind, and Wooldridge (2012) covers some topics on complexity of cooperative games. Deng and Papadimitriou (1994) and more recently Deng and Fang (2008) and Bilbao, Fernández, and López (2014) also survey some result on the computational aspects of cooperative game theoretical solutions. There is no comprehensive study that focuses solely on the computational complexity of the nucleolus in general, however there are many partial results that we will discuss in Chapter 3.

\(^3\)Here we do not want to dive into the depth of theory of computation. A formal definition of complexity classes based on languages and Turing machines can be found in the excellent book of Sipser (1997). Another influential textbook in the subject is (Garey and Johnson, 1979).
Chapter 2

Game theoretical framework

Some of the results presented in this chapter are based on the articles (Biró, Kóczy, and Sziklai, 2013) and (Csercsik and Sziklai, 2013).

2.1 Properties of the characteristic function

In this section we review some basic properties of the characteristic function.

Definition 2.1. A cooperative game \((N, v)\) is said to be monotonic if

\[ S \subseteq T \subseteq N \Rightarrow v(S) \leq v(T). \]

Monotonicity is a weaker property than superadditivity in the sense that every non-negative superadditive game is necessarily monotonic. Note that monotonicity and \(v(\emptyset) = 0\) together implies that the values of the characteristic function are non-negative. It is indeed hard to come up with a cooperative game that models a real life situation for which the characteristic function is not monotonic. If a payoff is feasible for a coalition \(S\), then it should be feasible for any enlargement of \(S\) – by ignoring the new members the coalition should be not worse off. This property is fundamental for cost games as well (see Section 2.6). Further variants of this property and their relationship is discussed in (Fürgő, Pintér, Simonovits, and Solymosi, 2006) and in (Brânzei, Dimitrov, and Tijs, 2005). For sake of simplicity we will omit the braces hereinafter from the following expressions \(S \cup \{i\}, S \setminus \{i\}, v(\{i\})\) and simply write \(S \cup i, S \setminus i\) and \(v(i)\).
**Definition 2.2.** A cooperative game \((N, v)\) is called *strongly superadditive* if

\[
(S, T \subset N) \Rightarrow v(S) + v(T) \leq v(S \cup T).
\]

Note that this version of superadditivity (cf. Definition 1.1) does not require for \(S\) and \(T\) to be disjoint. An even stronger property is convexity.

**Definition 2.3.** A cooperative game \((N, v)\) is *convex* if the characteristic function is supermodular, i.e.

\[
(S, T \subseteq N) \Rightarrow v(S) + v(T) \leq v(S \cup T) + v(S \cap T).
\]

Deriving results for convex games is a less challenging task than in general due to their nice structure. Here we only mention a result of Shapley (1971), namely that the core of a convex game is not empty (cf. Eq. (2.2)). In the upcoming sections we will review further results that were developed for this class of games.

**Definition 2.4.** Two games \((N, u)\) and \((N, v)\) are *strategically equivalent* if there exist \(\alpha > 0\) and \(\beta \in \mathbb{R}^N\) such that \(u(S) = \alpha v(S) + \beta(S)\) for all \(S \subseteq N\).

Strategic equivalence is conform with our intuition on how transferable utility games work. From strategic point of view, it should not matter whether a player is involved in \(u\) or \(v\), since his utility for one game can be obtained by applying an affine transformation on the utility he gained in the other game.

**Definition 2.5.** A game \((N, v)\) is *zero-normalized* if \(v(i) = 0\) for all \(i \in N\).

Clearly every game is strategically equivalent to some zero-normalized game.

**Definition 2.6.** A game \((N, v)\) is *weakly superadditive* if \(v(S \cup i) \geq v(S) + v(i)\) for all \(S \subseteq N, i \notin S\).

Every weakly superadditive game is strategically equivalent to a zero-normalized monotonic game. Therefore weakly superadditive games are also called *zero-monotonic*.
2.2 Properties of solutions

Let \((N, v)\) be a cooperative game and \(x \in \mathbb{R}^N\) a corresponding solution. For convenience, we introduce the following notations \(x(S) = \sum_{i \in S} x_i\) for any \(S \subseteq N\), and instead of \(x(\{i\})\) we simply write \(x(i)\). For \(x, y \in \mathbb{R}^n\), we say that \(x \geq y\) if \(x_i \geq y_i\) for \(i = 1, 2, \ldots, n\). Furthermore, let \(\mathcal{P} = 2^N \setminus \{\emptyset, N\}\) denote the family of the non-trivial – or as sometimes called proper – coalitions.

A solution \(x\) is called an allocation if it is efficient, i.e. \(x(N) = v(N)\). Similarly, we say that \(x\) is individually rational if \(x(i) \geq v(i)\) for all \(i \in N\). The imputation set of the game \(I(N, v)\) consists of the efficient and individually rational solutions, formally,

\[
I(N, v) = \{ x \in \mathbb{R}^N \mid x(N) = v(N), \; x(i) \geq v(i) \text{ for all } i \in N \}.
\]

The set of allocations is synonymously called preimputation set and denoted by \(I^*(N, v)\). Obviously, \(I(N, v)\) is non-empty if and only if \(\sum_{i \in N} v(i) \leq v(N)\). Individual rationality composes a very mild restriction on the set of allocations. If we cannot guarantee for each player its individual worth, then the stability of the grand coalition is questionable. In such cases there exists a player who gets better off by deviating. The concept of core extends this idea a little further. To define it we will need a notion for profit.

Given a game \(\Gamma = (N, v)\) and an allocation \(x \in \mathbb{R}^N\), we define the satisfaction of a coalition \(S\) as

\[
sat_\Gamma(S, x) := x(S) - v(S).
\]

Clearly the satisfaction can be considered as the measurement of profit that the coalition achieved\(^1\). When there is no confusion we omit the lower index from \(sat_\Gamma\).

\(^1\)In the standard literature the (dis)satisfaction of a coalition is actually measured by the so-called excess, which is defined in the opposite way, i.e. \(exc(S, x) := v(S) - x(S)\) (see (Peleg and Sudhölter, 2007; Osborne and Rubinstein, 1994)). In my opinion this is an example of a conceptual mistake that was born early and got stuck. It is much more intuitive to associate dissatisfaction of a coalition with negative values. Many attempted to deviate from this convention. Raghavan (1997) used the same approach as we apply here with the exception that he denoted the satisfaction function with \(f\). I sympathise with the approach of Serrano (2005) and Benoît (1997) who simply redefined excess as \(exc(S, x) := x(S) - v(S)\). To avoid confusion we stick with the above definition.
The core of the game \((N, v)\) is a set-valued solution where all the satisfaction values are non-negative. Formally,

\[ C(N, v) = \{ x \in \mathbb{R}^N \mid x(N) = c(N), \ x(S) \geq v(S) \text{ for all } S \subseteq N \}. \tag{2.2} \]

A game is called balanced if its core in non-empty. If \(x \in C(N, v)\) then no coalition can improve upon \(x\), in the sense, that each coalition gets at least as much as it could gain on its own. Thus, each member of the core is a highly stable payoff distribution. Note that in the \(n\)-dimensional payoff space the core allocations form a convex polyhedron. Furthermore, if \(v\) is monotonic then the non-negativity of any core allocation \(x\) follows from the simple observation that \(0 \leq v(i) \leq x(i)\) is true for any \(i \in N\), where the first inequality follows from the fact that \(v\) is monotonic and \(v(\emptyset) = 0\), while the second from the non-negativity of satisfaction values.

Let \(\mathcal{V}^N\) be the set of all coalition functions on \(2^N\) and \(\mathcal{G} \subseteq \mathcal{V}^N\). A solution \(\sigma\) on the game set \(\mathcal{G}\) is said to be Pareto optimal (PO) if it consists only of allocations. Another basic requirement is non-emptiness (NE). A solution \(\sigma\) has this property on game set \(\mathcal{G}\) if for any \(v \in \mathcal{G}\), \(\sigma(v) \neq \emptyset\). In the upcoming sections we will show characterizations of the Shapley-value and the nucleolus. For these we need further axioms.

**Definition 2.7.** A single-valued solution \(\sigma\) on \(\mathcal{G}\) is additive (ADD) if

\[ \sigma(u) + \sigma(v) = \sigma(u + v) \text{ when } u, v, u + v \in \mathcal{G}, \]

where \(u + v\) denotes the component-wise addition of \(u\) and \(v\).

Additivity describes a situation when the same set of players play two different games. Then the payoff of any coalition for the combined game should be the sum of what the coalition received for those games individually. A similar technical axiom is covariance.

**Definition 2.8.** A solution \(\sigma\) on \(\mathcal{G}\) is covariant under strategic equivalence (COV) if the following is true: If \(u, v \in \mathcal{G}, \alpha > 0, \beta \in \mathbb{R}^n\) and \(u = \alpha v + \beta\), then

\[ \sigma(u) = \alpha \sigma(v) + \beta. \]
2.2 Properties of solutions

The COV property requires that if two games \((N, u)\) and \((N, v)\) are strategically equivalent then their solution sets should be related by the same transformation on the utilities of the players.

Let \(v \in V^N\) be fixed. Then we say that \(i, j \in N\) players are equivalent, denoted by \(i \sim_v j\) if \(v(S \cup i) = v(S \cup j)\) is true for all \(i, j \not\in S \subseteq N\). It is easy to check that \(\sim_v\) is indeed an equivalence relation.

**Definition 2.9.** A solution \(\sigma\) on \(G\) has the equal treatment property (ETP) if the following is satisfied. For all \(v \in G\) and \(x \in \sigma(v)\) if \(i, j\) satisfy \(i \sim_v j\) then \(x(i) = x(j)\).

The principle of equal treatment is one of the most basic properties that can be imposed to achieve fairness. It ensures that if the strategic positions of two players are symmetric then so are their payoffs. A slightly stronger requirement is anonymity. Let \(\mathcal{U}\) be a set of players and \(\pi : N \to N\) be an injection, then the game \((\pi(N), \pi v)\) is defined by \(\pi v(\pi(S)) = v(S)\) for all \(S \subseteq N\). Also if \(x \in \mathbb{R}^N\), then \(y = \pi(x) \in \mathbb{R}^{\pi(N)}\) is given by \(y_{\pi(i)} = x_i\) for all \(i \in N\).

**Definition 2.10.** Let \(\sigma\) be a solution on \(G\). We say that \(\sigma\) is anonymous (AN) if the following condition is satisfied: If \((N, v) \in G\), \(\pi : N \to N\) is an injection and if \((\pi(N), \pi v) \in G\), then \(\sigma(\pi(N), \pi v) = \pi(\sigma(N, v))\).

Anonymity simply states that exchanging the names of the players will not alter the solution. Clearly ETP follows from AN. Another straightforward axiom is the dummy-player property. Let \((N, v)\) be a game. A player \(i\) is a null- or dummy-player if \(v(S) = v(S \cup i)\) for every \(S \subseteq N\).

**Definition 2.11.** A solution \(\sigma\) on \(G\) has the dummy-player property (DP) if for all \(v \in G\) and \(x \in \sigma(v)\), if \(i \in N\) is a dummy player then \(x(i) = 0\).

Another desirable property is monotonicity, which has many variant in the literature. Here we will state two.

**Definition 2.12.** A single-valued solution \(\sigma\) on \(G\) is strongly monotonic (SMON) if the following condition is satisfied: If \(u, v \in G\), \(i \in N\), and

\[ u(S \cup i) - u(S) \geq v(S \cup i) - v(S) \text{ for all } S \subseteq N, \]

then \(\sigma_i(u) \geq \sigma_i(v)\).
Strong monotonicity was propagated by Young (1985), who used it to characterize the Shapley-value.

**Definition 2.13.** A solution $\sigma$ on $G$ satisfies *aggregate monotonicity* (AMON) if for all games $u,v \in G$, such that $u(N) \geq v(N)$ and $u(S) = v(S)$ for all $S \subseteq N$, moreover for all $x \in \sigma(v)$, there exits $y \in \sigma(u)$ such that $y \geq x$.

Aggregate monotonicity describes what should happen when it turns out there is more to share. It asserts that nobody is worse off just because the size of the cake increased. The next axiom is related to the nucleolus.

**Definition 2.14.** Let $(N,v)$ be a game, $\emptyset \neq S \subseteq N$ a coalition, and $x \in \mathbb{R}^n$ an arbitrary allocation. The *reduced game* with respect to $S$ and $x$ is the game $(S,v_{(S,x)})$ defined by

$$
 v_{(S,x)}(T) = \begin{cases} 
 0 & \text{if } T = \emptyset, \\
 v(N) - x(N \setminus S) & \text{if } T = S, \\
 \max_{Q \subseteq N \setminus S}(v(T \cup Q) - x(Q)) & \text{otherwise}. 
\end{cases}
$$

The reduced game describes a situation when all the players agree that coalition $N \setminus S$ should get $x(N \setminus S)$. These players leave the game with their payoff. However, the remaining players may still cooperate with them. Then, for every non-empty $T \subsetneq S$ coalition, the amount $v_{(S,x)}(T)$ is the maximal total payoff that the coalition may hope to get. However, the expectations of different disjoint subcoalitions may not be compatible with each other, because they may require the cooperation of the same subset of $N \setminus S$. Thus, $(S,v_{(S,x)})$ is not a game in the ordinary sense, it serves only to determine the distribution of $v_{(S,x)}(S)$ to the members of $S$. For an example see (Peleg and Sudhölter, 2007).

**Definition 2.15.** Let $G \subseteq \bigcup N V^N$ be a set of games. A solution $\sigma$ on $G$ satisfies the *reduced game property* (RGP) if for every $(N,v), v \in G$ and for every $\emptyset \neq S \subseteq N$ the following condition is satisfied: If $x \in \sigma(v)$, then $v_{(S,x)} \in G$ and $x^S \in \sigma(S,v_{(S,x)})$.

The reduced game property was introduced by Davis and Maschler (1965). Later Snijders (1995) augmented the concept.
**Definition 2.16.** Let \((N, v)\) be a game, \(\emptyset \neq S \subseteq N\) a coalition, and \(x \in \mathbb{R}^n\) an arbitrary allocation. The *imputation saving reduced game* with respect to \(S\) and \(x\) is the game \((S, v^{(S,x)})\) defined as follows: If \(|S| = 1\), then \(v^{(S,x)} = v(S, x)\). If \(|S| \geq 2\), then

\[
v^{(S,x)}(T) = \begin{cases} 
  v(S,x)(T) & \text{if } T \subseteq S \text{ and } |T| \neq 1, \\
  \min\{x(j), v(S,x)(j)\} & \text{if } T = \{j\} \text{ for some } j \in S.
\end{cases}
\]

The aim of both reduced game concepts is to introduce self-consistency in the model. Young (1987) advocated that every part of a fair solution should be fair. The corresponding property is defined as follows.

**Definition 2.17.** Let \(G \subseteq \bigcup_N \mathcal{V}^N\) be a set of games. A solution \(\sigma\) on \(G\) satisfies the *imputation saving reduced game property* (ISRGP) if for every \((N, v), v \in G\) and for every \(\emptyset \neq S \subseteq N\) the following condition is satisfied: If \(x \in \sigma(v)\), then \(v^{(S,x)} \in G\) and \(x^S \in \sigma(S, v^{(S,x)})\).

There are many other fairness axioms, yet we stop listing them here. Our aim is to compare the nucleolus with the Shapley-value. For this the above introduced properties are more than enough. For further information the reader is referred to the works quoted above.

## 2.3 Shapley-value

Shapley (1953) was the first to propose a single value for \(TU\)-games. His idea was to measure the productivity of a player by its *marginal contribution*.

**Definition 2.18.** Let \(v \in \mathcal{V}^N\). For each \(i \in N\) and for each \(S \subseteq N \setminus \{i\}\), the marginal contribution of player \(i\) to the coalition \(S\) is \(m_i(S, v) = v(S \cup i) - v(S)\).

Consider that players arrive one after another in a room. Each player is rewarded by it’s marginal contribution upon arrival. The Shapley-value assumes that each order of players is equally likely and takes the expected value of the different allocations.
Definition 2.19. The Shapley-value $\Phi$ on $\mathcal{V}^N$ is given by

$$
\Phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(n-|S|-1)}{n!} (m_i(S,v))
$$

for every $v \in \mathcal{V}^N$ and for every $i \in N$.

Although, the Shapley-value is extremely popular among game theorists, it has some drawbacks. First and foremost it is not necessarily a core allocation, even when the core is non-empty. It does not satisfy individual rationality and RGP in general. Some recent papers criticize the Shapley-value for relying too much on the productivity of players, leaving no room for solidarity (Casajus and Huettner, 2013). On the positive side there are many known characterizations, here we mention two of them.

Theorem 2.20. (Shapley, 1953) There is a unique solution $\Phi \in \mathcal{V}^N$ that satisfies NE, ETP, DP, and ADD.

Theorem 2.21. (Young, 1985) Let $\sigma$ be a single valued solution on $\mathcal{V}^N$. If $\sigma$ satisfies SMON, ETP and PO, then $\sigma$ is the Shapley-value.

For this latter theorem Pintér (2012) recently provided a simpler proof. Although, by definition it takes exponential time in the number of players to compute the Shapley-value, in practice this is usually a much easier task due to the various explicit formulas and the many existing axiomatizations. Another important well-known fact is that the Shapley-value is in the core if the game is convex. Furthermore, it satisfies anonymity, aggregate monotonicity and numerous other axioms. For a comprehensive list the reader is referred to (Peleg and Sudhölter, 2007).

2.4 Nucleolus

We say that a vector $x \in \mathbb{R}^m$ lexicographically precedes $y \in \mathbb{R}^m$ (denoted by $x \preceq y$) if, either $x = y$, or there exists a number $1 \leq j \leq m$ such that $x_i = y_i$ if $i < j$ and $x_j < y_j$.

Definition 2.22. Let $\Gamma = (N, v)$ be a game, $X \subseteq \mathbb{R}^N$ a set of payoff vectors, $x$ an allocation, and let $\theta^P(x) \in \mathbb{R}^{2n}$ be the satisfaction vector that contains the
2.4 Nucleolus

$2^{n-2}$ satisfaction values of the non-trivial coalitions in a non-decreasing order. The nucleolus of $\Gamma$ with respect to $X$ is the subset of those payoff vectors $x \in X$ that lexicographically maximize $\theta^P(x)$ over $X$. Formally,

$$\mathcal{N}(\Gamma, X) = \{x \in X \mid \theta^P(y) \preceq \theta^P(x) \text{ for all } y \in X\}.$$ 

It is well known that if $X$ is nonempty and compact then $\mathcal{N}(\Gamma, X) \neq \emptyset$ and if, in addition, $X$ is convex then $\mathcal{N}(\Gamma, X)$ consist of a single point (for proof see (Schmeidler, 1969)). Furthermore, the nucleolus is a continuous function of the characteristic function. If $X$ is chosen to be the set of allocations, we speak of the prenucleolus of $\Gamma$, if $X$ is the set of imputations then we speak of the nucleolus of $\Gamma$. Throughout the thesis we will use the shorthand notation $\mathcal{N}(\Gamma)$ for $\mathcal{N}(\Gamma, I(N, v))$.

The axiomatization of the nucleolus proved to be harder than that of the Shapley-value. The first result is due to Sobolev (1975) who managed to find a characterization for the prenucleolus.

**Theorem 2.23.** (Sobolev, 1975) There is a unique solution on $\cup_N \mathcal{V}^N$ that is single valued and satisfies COV, AN, and RGP, and it is the prenucleolus.

Later Potters (1990) provided an axiomatization of the nucleolus on the set of games for which the nucleolus and the prenucleolus coincide. Finally, Snijders (1995) settled the question for all cooperative games with finite player set that have a nucleolus.

**Theorem 2.24.** (Snijders, 1995) Let $\mathcal{G}^I \subseteq \cup_N \mathcal{V}^N$ be the set of games with finitely many players that have a non-empty imputation set. There is a unique solution on $\mathcal{G}^I$ that is single valued and satisfies COV, AN, and ISRGP, and it is the nucleolus.

The nucleolus does not satisfy strong or aggregate monotonicity. On the other hand it satisfies DP and ETP, and it is always in the core if the core is non-empty. As core stability is crucial in many applications some researchers prefer the nucleolus over the Shapley-value.

Unlike to the Shapley-value there is no explicit formula for the nucleolus for general class of games. Also verifying properties like RGP is a complicated task\footnote{For instance the reduced game of an irrigation game is not an irrigation game (Szatzker, 2012).}.
Therefore many researchers consider the nucleolus as a nice but too complex solution concept, which does not worth the effort to deal with. However, in the recent years there has been great progress in the computation aspect of the problem. The next chapter is dedicated to the various methods that were developed to the computation of the nucleolus.

2.5 Solution concepts related to the nucleolus

Although, the core, the Shapley-value and the nucleolus are undoubtedly the most popular cooperative solution concepts, there are countless others that are studied and applied throughout in the literature. Here we survey a few that are related to the nucleolus in one or other way. First let us review two set-valued solution concepts that contain the nucleolus.

**Definition 2.25.** Let \( \varepsilon \) be a real number. The \( \varepsilon \)-core of the game \( \Gamma = (N, v) \), \( C_\varepsilon \), is defined by

\[
C_\varepsilon = \{ x \in I^*(N, v) \mid \text{sat}(S, x) \geq \varepsilon \text{ for all } \emptyset \neq S \subset N \}.
\]

The least-core of the game \( \Gamma \), denoted by \( \mathcal{L}C(\Gamma) \) is the intersection of all non-empty \( \varepsilon \)-cores of \( \Gamma \).

The least core can be also defined in the following way. Let \( \varepsilon_0 \) be the smallest \( \varepsilon \) such that \( C_\varepsilon(\Gamma) \neq \emptyset \), that is,

\[
\varepsilon_0 = \max_{x \in I^*(N, v) \emptyset \neq S \subset N} \min \text{sat}(S, x).
\]

Then \( \mathcal{L}C(\Gamma) = C_{\varepsilon_0}(\Gamma) \). The \( \varepsilon \)-core can be interpreted as the set of efficient payoff vectors that cannot be improved upon by any coalition if forming a coalition entails a 'cost' of \( \varepsilon \) (or a bonus' of \( -\varepsilon \), if \( \varepsilon \) is negative). If the core of a game is not empty (i.e. if \( \varepsilon \geq 0 \)), then the least-core is a centrally-located locus (or point) within the core. If the core is empty (i.e. if \( \varepsilon < 0 \)), then the least-core reveals the latent position of the core. Maschler, Peleg, and Shapley (1979) proved that the least core is in the imputation set whenever the game is zero-monotonic. Furthermore, the nucleolus is contained in any non-empty \( \varepsilon \)-core, in particular the nucleolus is an element of the least-core.
2.5 Solution concepts related to the nucleolus

Definition 2.26. Let \((N,v)\) a game and \(x \in I^*(N,v)\). The maximum surplus of player \(i\) over player \(j\) with respect to \(x\) is

\[
s_{ij}(x) = \max \{v(S) - x(S) : S \subseteq N \setminus \{j\}, i \in S\}.
\]

The maximum surplus is a way to measure one player’s bargaining power over another. The kernel of \(v\) is the set of imputations \(x\) that satisfy

\[
(s_{ij}(x) - s_{ji}(x)) \cdot (x(j) - v(j)) \leq 0
\]

for every pair of players \(i\) and \(j\). The kernel was introduced by Davis and Maschler (1965). Intuitively, player \(i\) has more bargaining power than player \(j\) with respect to imputation \(x\) if \(s_{ij}(x) > s_{ji}(x)\), but player \(j\) is immune to player \(i\)’s threats if \(x(j) = v(j)\), because he can obtain this payoff on his own. The kernel contains all imputations where no player has this bargaining power over another. Schmeidler (1969) proved that if the imputation set is not empty then the nucleolus belongs to the kernel. Maschler, Peleg, and Shapley (1972) showed that the kernel is a single point for convex games, which is therefore the nucleolus. For more on the least-core and the kernel the reader is referred to (Maschler, Peleg, and Shapley, 1979) and (Peleg and Sudhölter, 2007).

There are many nucleolus inspired single valued solution concepts. The average lexicographic value (Alexia) is the average of all lexicographic maxima of the core. Alexia value coincides with the Shapley value for convex games, and with the nucleolus for both strongly compromise admissible games and big boss games (Tijs, Borm, Lohmann, and Quant, 2011). The nucleon, which was introduced by Faigle, Kern, Fekete, and Hochstädtler (1998) may be viewed as the multiplicative analogue of the nucleolus. It maximizes the satisfaction ratio \((x(S)/v(S))\) instead of the satisfaction values. The per capita nucleolus is based on the maximal dissatisfaction per player for each coalition, that is the satisfaction values are normalized with the sizes of the coalitions. This solution concept was introduced by Grotte (1970) but it did not become that popular until recently. In the past few years however it is getting more attention (see for instance (Huijink, Borm, Reijnierse, and Kleppe, 2013)). Derks and Haller (1999) extended the concept of the per capita nucleolus and considered the lexicographic optimization of arbitrarily weighted satisfaction values. An even more general setup can be found in (Potters
and Tijs, 1992). Sudhölter (1997) provided yet another variant of the nucleolus. The modified nucleolus takes into account both the 'power', that is, the worth, and the 'blocking power' of a coalition, i.e. the amount which the coalition cannot be prevented from by the complement coalition. Finally, Hinojosa, Már mol, and Thomas (2005) extended the notion of the core and nucleolus for multiple scenario cooperative games, where the characteristic function is vector-valued.

Lexicographic solution concepts are used in non-cooperative games as well. In the so-called apportionment problem there is a parliament with a fixed number of seats and administrative regions with different number of voters. It is a universal principle with respect to democratic elections that voters should have equal influence on the outcome, which in turn require equally sized electoral districts. The question is — assuming that electoral districts are of equal size within each of the administrative regions — how to distribute the seats among the administrative regions to minimize the differences between the voters' influences.

Balinski and Young (1982) offer a comprehensive historical overview on this matter, discussing many apportionment rules and their properties. A scenario where increasing the Parliament size would decrease the number of seats allotted to an administrative region is often considered undesirable, perhaps even paradoxical. An apportionment rule where this is possible is said to exhibit the Alabama paradox referring to a historical occurrence of the phenomenon in the case of state Alabama. A rule is said to be house-monotonic if it does not suffer from such weakness. Note that house monotonicity is basically the appropriate conversion of aggregate monotonicity. This is not the only connection between cooperative games and apportionment problems. Biró, Kóczy, and Sziklai (2013) defines an apportionment rule that lexicographically minimizes the differences between the populations of the constituencies to the average constituency size. This so-called leximin rule resembles to the nucleolus in more than one way. For instance, the leximin rule does not satisfy house-monotonicity similarly to the nucleolus which does not satisfy aggregate monotonicity\(^3\). A short description of the leximin rule can be found in Biró, Sziklai, and Kóczy (2012).

\(^3\)One could argue that the leximin rule is more like the nucleon, as it is based on ratios and not on differences. However, the nucleon — unlike to the nucleolus — is aggregate monotonic.
2.6 Cost allocation games

In many economic situations cooperation results in cost saving rather than profit growth. For instance, such situations occur when customers would like to gain access to some public service or public facility. The question is then, how to share the incurring costs. Cost allocation games can be modeled in a similar fashion as cooperative TU-games.

A cooperative cost game is an ordered pair \((N, c)\) consisting of the player set \(N = \{1, 2, \ldots, n\}\) and a characteristic cost function \(c : 2^N \to \mathbb{R}\) with \(c(\emptyset) = 0\). The value \(c(S)\) represents how much cost coalition \(S\) must bear if it chooses to act separately from the rest of the players. In most cases \(c\) is monotonic and subadditive. That is, the more people use the service the more it costs, however there is also an increase of efficiency. It is possible to associate a profit game \((N, v)\) with a cost game \((N, c)\), called the savings game, which is given by \(v(S) = \sum_{i \in S} c(i) - c(S)\) for all \(S \subseteq N\).

**Definition 2.27.** A cost game \((N, c)\) is called subadditive if

\[(S, T \subseteq N, S \cap T = \emptyset) \Rightarrow c(S) + c(T) \geq c(S \cup T)\]

and concave if,

\[c(S) + c(T) \geq c(S \cup T) + c(S \cap T), \quad \forall S, T \subseteq N.\]

Note that a cost game is subadditive (concave\(^4\)) if and only if the corresponding savings game is superadditive (convex). Similarly, the relationship is reversed in all the previously listed concepts such as individual rationality, satisfaction, the core, and so on. Let \(\Gamma = (N, c)\) be a cost game and \(x \in \mathbb{R}^N\) an arbitrary allocation. The satisfaction of a coalition \(S\) is defined as

\[\text{sat}_\Gamma(S, x) := c(S) - x(S).\]

Note that in case of cost games the order of the characteristic function and the payoff vector is reversed (cf. Eq. 2.1 and the corresponding footnote), hence the

\(^4\)Sometimes submodular cost games are called convex instead of concave in the same way we usually speak of the core of a cost game instead of its anti-core. This terminology is appealing since for instance the result of Kuipers (1996) on convex games naturally extends to concave cost games.
Game theoretical framework

formula for \( \text{sat}_\Gamma \) depends on whether \( \Gamma \) is a cost or profit game\(^5\). Again the satisfaction of \( S \) indicates the contentment of the coalition. We say that \( x \) is individually rational if \( x(i) \leq c(i) \) for all \( i \in N \) and \( x \) is a core member if all the satisfaction values are non-negative. Notice that core vectors of any monotonic cost game \( \Gamma = (N, c) \) are non-negative. Indeed,

\[
x(i) = x(N) - x(N \setminus i) \geq c(N) - c(N \setminus i) \geq 0
\]

for any \( i \in N \) and \( x \in C(\Gamma) \). Henceforward we will spare the reader of defining every concept and stating every theorem both for profit and cost games, except for the most crucial ones. In most cases the translation can be done simply by changing the sign of an expression or reversing the inequality sign. We will always indicate where more elaboration is needed.

2.7 Partition function form games

In their pioneer work, Neumann and Morgenstern (1944) argue that the union of two disjoint coalitions should not obtain less, than what the two coalitions could achieve on their own. The rationale behind this is that any strategies that were available for the two coalitions separately is also available for the merged coalition. This implies the superadditivity of the characteristic function (or subadditivity in case of cost games). Moreover, it allows us to assume that the grand coalition will form.

Although this argument is compelling, there are some situations where it does not apply. Consider for instance a parliament where there is large middle party and two smaller ones which represent the radical and the liberal side. The radical and liberal side swore not to be on the same platform with each other. The representatives of these two parties are therefore boycotting any coalition where the other one is present. In this case the grand coalition is worse off than the middle party together with one of the smaller ones.

\(^5\)It is standard in the literature to define the excess of a cost game the same way as profit games, that is, \( \text{exc}_\Gamma(S, x) := c(S) - x(S) \) (see for instance (Maschler, Potters, and Reijnierse, 2010) or (Granot and Huberman, 1984)). However, the meaning of \( \text{exc}_\Gamma(S, x) \) changes from the measure of dissatisfaction to the measure of satisfaction of \( S \). As we will discuss both profit and cost games we rather change the formula of \( \text{sat}_\Gamma \) than changing its meaning.
Another example is provided in (Csercsik and Sziklai, 2013), where multiple navigation companies route their clients on the same traffic network. Sharing information and cooperation with other agents may result in a more efficient utilization of the network capacities. However, the formation of a coalition may affect other players costs via the increased edge loads in the traffic network, thus externalities may arise. Furthermore, each coalition optimizes its routing according to its current knowledge. Only the grand coalition possesses every relevant information, hence the merging of two coalitions may result in extra costs rather than in cost saving\(^6\).

In general when externalities are present the superadditivity property often fails. To handle externalities in a cooperative framework Thrall and Lucas (1963) introduced partition function form (PFF) games. In a PFF game the value of a coalition \(S\) not only depends on the members of \(S\), but also on the partition in which \(S\) is embedded. Thus, the characteristic function \(v\) is replaced by a partition function \(V : \Pi \rightarrow (2^N \rightarrow \mathbb{R})\) that assigns a characteristic function to each partition (here \(\Pi\) denotes the partitions of \(N\)).

While for the generalisation of the core to partition function form games Kóczy (2007) introduced a concept, the recursive core that is well motivated and mathematically appealing, for the nucleolus no well established generalisation exists. Current attempts include two unpublished papers by McCain (2011) and Tripathi and Amit (2014).

\(^6\)It seems paradoxical that having 'more' information causes the coalition to choose a less optimal strategy. How a strategy fares, however, depends also on the information the coalition does not possess. It may happen that an 'ignorant' strategy fares better.
Chapter 3

Computing the nucleolus

Section 3.2 and 3.3 of this chapter is based on (Solymosi and Sziklai, 2015).

3.1 The nucleolus by linear programming

Kopelowitz (1967) was the first\(^1\) to propose a series of linear programs to compute the nucleolus. With this method the number of linear programs that have to be solved can raise up to \(2^{n-1}\). Kohlberg (1972) offered a unique minimization problem which in exchange suffers from another problem: It has astronomical \(O(2^n!)\) number of constraints. Owen (1974) showed how this LP can be reduced to a 'more tractable' size of \(O(4^n)\) constraints, although the constraint coefficients are too large to use the program for practical purposes. Here we present the idea of Maschler, Peleg, and Shapley (1979). This sequential LP was the first that was suitable for computational purposes.

Let \(X^0 := I(N,v)\) and \(\Sigma^0 := \mathcal{P}\). For \(r = 1, \ldots, \varrho\) we define recursively

\[
\begin{align*}
\varepsilon^r &:= \max_{x \in X^{r-1}} \min_{S \in \Sigma^{r-1}} \text{sat}(S, x), \\
X^r &:= \{x \in X^{r-1} : \min_{S \in \Sigma^{r-1}} \text{sat}(S, x) = \varepsilon^r\}, \\
\Sigma_r &:= \{S \in \Sigma^{r-1} : \text{sat}(S, x) = \varepsilon^r, \text{ for all } x \in X^r\}, \\
\Sigma^r &:= \Sigma^{r-1} \setminus \Sigma_r,
\end{align*}
\]

\(^{1}\)Oddly enough, Schmeidler introduced the nucleolus in 1969, whereas Kopelowitz's working paper came out in 1967.
where $\varrho$ is the first value of $r$ for which $\Sigma^r = \emptyset$. Maschler, Peleg, and Shapley (1979) also described the geometric properties of the nucleolus and gave interpretation of above sequential linear program. In the $n$-dimensional payoff space the satisfaction values of the different coalitions each correspond to a hyperplane. During the $r^{th}$ linear program $\varepsilon^r$ is reached by pushing the hyperplanes to their limit, i.e. till the point that any further push would empty the solution set. The satisfaction values of the coalitions whose hyperplane collided ($\Sigma^r$) becomes fixed. This process continues till the lexicographic center of a game - which turns out to be the nucleolus - is reached. Note that in general this method needs $O(4^n)$ number of linear programs with constraint coefficients in $\{-1, 0, 1\}$.

Since then, there have been many attempts to improve the computation process either by restraining the number of LPs or by finding a unique minimization problem with minimal number of constraints. Sankaran (1991) provided a method that needs $O(2^n)$ number of LPs with constraint coefficients in $\{-1, 0, 1\}$. Later Fromen (1997) improved his results by reducing the number of linear programs used in the computation to $O(n)$. Potters, Reijnierse, and Ansing (1996) proposed a fast algorithm to find the nucleolus of any game with non-empty imputation set. This algorithm is based on solving a prolonged simplex algorithm. It requires solving $n - 1$ linear programs with $O(2^n)$ number of rows and columns. The most recent result is by Puerto and Perea (2013). They offered a unique minimization problem, which complexity is similar to Owen’s algorithm. This method again uses $O(4^n)$ constraints, however the constraints coefficients are from the set $\{-1, 0, 1\}$. Further methods can be found in Wolsey (1976); Dragan (1981) and Hallefjord, Helming, and Jørnsten (1993).

An interesting addition to this topic is provided by Guajardo and Jörnsten (2014). They collect examples from the literature where the nucleolus was miscalculated and analyze what went wrong. Kido (2008) suggested a completely different approach in the form of a non-linear program.

As there are $2^n$ number of characteristic function values we cannot hope for a linear program which is polynomial in the number of players. However, depending on the game’s characteristics we may identify coalitions that are redundant in the computation process. In this way we may give an efficient LP for at least some classes of cooperative games. The most fruitful approach of this kind is by Granot, Granot, and Zhu (1998) who mix the linear programming technique with

$^2$Observe that $X^1$ is the least-core of $(N, v)$. 

24
the concept of characterization sets. We will discuss this method in details in the upcoming sections.

3.2 Verifying the nucleolus

Computing the nucleolus is a notoriously hard problem, even $NP$-hard for some classes of games. While $NP$-hardness has been proven for minimum cost spanning tree games (Faigle, Kern, and Kuipers, 1998), voting games (Elkind, Goldberg, Goldberg, and Wooldridge, 2009) and flow and linear production games (Deng, Fang, and Sun, 2009), it is still unknown whether the corresponding decision problem – i.e. deciding whether an allocation is the nucleolus or not – belongs to $NP$. Faigle, Kern, and Kuipers (1998) conjecture the decision problem: "given $x^* \in \mathbb{R}^N$, is $x^*$ the nucleolus?'' to be $NP$-hard in general. The most resourceful tool regarding this problem is the criterion developed by (Kohlberg, 1971).

A collection of coalitions $B \subseteq 2^N$ is said to be balanced if there exist positive weights $\lambda_S$, $S \in B$, such that $\sum_{S \in B} \lambda_{S} e_{S} = e_{N}$, where $e_{S} \in \{0, 1\}^n$ denotes the indicator vector of coalition $S$. The following theorem is called the Kohlberg-criterion.

**Theorem 3.1.** (Kohlberg, 1971) Let $\Gamma = (N, v)$ be a game with non-empty core and let $x \in I(\Gamma)$. Then $x = N(\Gamma)$ if and only if for all $y \in \mathbb{R}$ the collection $\{\emptyset \neq S \subset N \mid \text{sat}(S, x) \leq y\}$ is balanced or empty.

The Kohlberg-criterion is often used to verify the nucleolus in practical computation when the size of the player set is not too large. As the following LP shows it is easy to tell whether a given collection of coalitions is balanced or not$^3$. Let $S_1, \ldots, S_m$ be the collection which balancedness is in question and let $q \in [0, 1]^m$. For $k = 1, \ldots, m$ let

\[
p_k^* = \max_{q_k} \quad \sum_{i=1}^{m} q_i e_{S_i} = e_{N} \\
q_1, \ldots, q_m \geq 0.
\]

---

$^3$I would like to thank professor Tamás Kis for the help in designing this linear program.
Lemma 3.2. The collection $S_1, \ldots, S_m$ is balanced if and only if $p_k^* > 0$ for each $k = 1, \ldots, m$.

**Proof.** Trivially $p_k^* > 0$ is a necessary condition. Let $a_1, \ldots, a_m \in [0, 1]$ be arbitrary reals such that $a_1 + \cdots + a_m = 1$. Furthermore, let $q^k$ be an optimal solution of the $k^{\text{th}}$ LP. We claim that $\lambda = \sum_{i=1}^{m} a_i q^i_j$ is a vector of balancing weights. Note that $\lambda_j = \sum_{i=1}^{m} a_i q^i_j > 0$ as $q^i_j = p^i_j > 0$ and $\sum_{j=1}^{m} q^i_j e_{S_i} = e_N$ for all $i = 1, \ldots, m$ by (3.1). Then

$$\sum_{j=1}^{m} \lambda_j e_{S_j} = \sum_{j=1}^{m} \sum_{i=1}^{m} a_i q^i_j e_{S_i} = \sum_{i=1}^{m} a_i \sum_{j=1}^{m} q^i_j e_{S_j} = (a_1 + \cdots + a_m) e_N = e_N.$$

The Kohlberg-criterion naturally extends to the prenucleolus\(^4\), where instead of $x \in I(N,v)$ we require only that $x \in I^*(N,v)$. A direct consequence of the extension is that the prenucleolus of monotonic games is non-negative.

**Theorem 3.3.** Let $\Gamma = (N,v)$ be a monotonic game and let $z$ denote its prenucleolus, then $z(i) \geq 0$ for all $i \in N$.

**Proof.** By contradiction suppose that $z(i) < 0$ for some $i \in N$. Let $\mathcal{B}_0$ contain the coalitions with the smallest satisfaction values under $z$. By the Sobolev-criterion $\mathcal{B}_0$ is a balanced collection. For every $S \in \mathcal{B}_0$, $i \in S$ otherwise $S \cup i$ would have an even smaller satisfaction due to the monotonicity of the characteristic function and the fact that $z(i) < 0$. By balancedness of $\mathcal{B}_0$, $\sum_{S \in \mathcal{B}_0} \lambda_S e_S = e_N$. As $i \in S$ for all $S \in \mathcal{B}_0$ this also means that $\sum_{S \in \mathcal{B}_0} \lambda_S = 1$. Then for all $j \neq i$ and for all $S \in \mathcal{B}_0$, $S$ must contain $j$. Thus the only coalition in $\mathcal{B}_0$ is the grand coalition. However, the grand coalition has zero satisfaction under any allocation, while coalition $\{i\}$ has negative satisfaction under $z$, which contradicts that $\mathcal{B}_0$ contains the coalitions with the smallest satisfaction values.

\(^4\)Some authors attribute this result to Sobolev (1975) and consequently refer to it as the Sobolev-criterion.

In recent years several polynomial time algorithms were proposed to find the nucleolus of important families of cooperative games, like standard tree (Maschler, Potters, and Reijnierse, 2010), assignment (Solymosi and Raghavan, 1994), matching (Kern and Paulusma, 2003) and bankruptcy games (Aumann and Maschler, 2003).
3.2 Verifying the nucleolus

In addition Kuipers (1996) showed that there exists an efficient algorithm to compute the nucleolus for convex games. Arin and Inarra (1998) also developed a method that can be used in case of convex games.

The main breakthrough came from another direction. In their seminal paper Maschler, Peleg, and Shapley (1979) described the geometric properties of the nucleolus, which became the basis of many future works. They also devised a computational framework, which we presented in the previous section. Although, this sequential linear program consists of exponentially many inequalities it can be solved efficiently if one knows which constraints are redundant. Huberman (1980); Granot, Granot, and Zhu (1998); Reijnierse and Potters (1998) provided methods to identify coalitions that correspond to non-redundant constraints.

**Definition 3.4.** Let \( 
\Gamma^F = (N, F, v) \) be a cooperative game with coalition formation restrictions, where \( \emptyset \neq F \subseteq P \) consists of all coalitions deemed permissible besides the grand coalition \( N \). Let \( \theta^F(x) \in \mathbb{R}^{|F|} \) be the restricted vector that contains the satisfaction values \( sat_{\Gamma}(S, x) \), \( S \in F \) in a non-decreasing order. Furthermore, let \( \mathcal{N}(\Gamma^F) \) be defined as the set of allocations that lexicographically maximize \( \theta^F(x) \). Then \( F \) is called a characterization set for the nucleolus of the game \( \Gamma = (N, v) \), if \( \mathcal{N}(\Gamma^F) = \mathcal{N}(\Gamma) \).

Note the different roles of \( N \) and the coalitions in \( F \) in the above optimization. The satisfaction of the grand coalition is required to be constantly zero (defining the feasible set), whereas the satisfactions of the smaller permissible coalitions form the objective function.

Interestingly this concept was already used by Megiddo (1974), but somehow went unnoticed at that time. Later Granot, Granot, and Zhu (1998) and Reijnierse and Potters (1998) re-introduced the idea almost simultaneously. The above definition originates from Granot, Granot, and Zhu (1998) who constructed a sequential LP process whose input is a characterization set and the characteristic function values for the coalitions contained therein, and the output is the nucleolus of the game. They showed that if the size of the characterization set is polynomially bounded in the number of players, then the nucleolus of the game can be computed in strongly polynomial time. In addition Reijnierse and Potters (1998) proved that for every game \((N, v)\) there exists a collection with at most \(2(n-1)\) coalitions that determine the nucleolus. Although, finding these coalitions is as hard as computing the nucleolus itself. Unfortunately this result does not make the verification of the
nucleolus belong to $NP$. Even if somehow we could effortlessly put the satisfaction values of the $2^n$ coalition in increasing order. It can happen that these $2(n-1)$ coalitions are scattered among the different balanced coalition arrays and we have to evaluate exponential many of them before we could confirm that the given allocation is indeed the nucleolus.

The Kohlberg-criterion applied to games with coalition formation restrictions yields the following theorem.

**Theorem 3.5.** (Maschler, Potters, and Tijs, 1992) Let $F$ be a characterization set and $x$ be an imputation of the game $\Gamma$ with $C(\Gamma) \neq \emptyset$. Then $x = N(\Gamma)$ if and only if for all $y \in \mathbb{R}$ the collection $\{S \in F \mid \text{sat}(S, x) \leq y\}$ is balanced or empty.

A similar criterion appears in (Groote Schaarsberg, Borm, Hamers, and Reijnierse, 2013). With the help of the Kohlberg-criterion the problem of finding the nucleolus is reduced to finding the right characterization set.

### 3.3 Characterization sets for the nucleolus

In this section we present some known characterization sets and introduce two new ones (dually essential and dually saturated coalitions) and point out some relations between them. The first is a fundamental result\(^5\) of Granot, Granot, and Zhu (1998). This theorem will be crucial in the proof of many theorems of this thesis.

**Theorem 3.6.** (Granot, Granot, and Zhu, 1998) Let $\Gamma$ be a cooperative (cost or profit) game, $F \subseteq \mathcal{P}$ a non-empty collection, and $x$ an element of the nucleolus of $\Gamma^F$. Collection $F$ is a characterisation set for the nucleolus of $\Gamma$ if for every $S \in \mathcal{P} \setminus F$ there exists a non-empty subcollection $F_S$ of $F$, such that

i. $\text{sat}_\Gamma(T, x) \leq \text{sat}_\Gamma(S, x)$ for every $T \in F_S$,

ii. $e_S$ can be expressed as a linear combination of the vectors in $\{e_T : T \in F_S \cup \{N\}\}$.

\(^5\)A very similar result appears in (Reijnierse and Potters, 1998). It is remarkable that two such closely related and important paper appeared in the same year.
Observe that the above conditions are sufficient but not at all necessary. Take for example the (superadditive and balanced) profit game with four players \( N = \{1, 2, 3, 4\} \) and the following characteristic function: \( v(i) = 0, v(i, j) = 1, v(i, j, k) = 3 \) for any \( i, j, k \in N \) and \( v(N) = 4 \). Then the 2-player coalitions and the grand coalition are sufficient to determine the nucleolus, which is given by \( z(i) = 1 \) for all \( i \in N \). However, the 3-player coalitions have smaller satisfaction values at \( z \), thus the first condition of Theorem 3.6 is violated. Notice that in this game the 3-player coalitions and the grand coalition are also sufficient to determine the nucleolus.

In general neither the 2-player nor the 3-player coalitions (and the grand coalition) characterize the nucleolus. The fact that in this example they did was due to the particular choice (most importantly the symmetry) of the coalitional function. We would like to identify properties of coalitions that characterize the nucleolus independently of the realization of the coalitional function, at least for a large class of games.

A straightforward corollary of Theorem 3.6 is that we can enlarge characterization sets arbitrarily.

**Corollary 3.7.** Let \( F \subseteq P \) be a characterization set that satisfies both conditions of Theorem 3.6. Then \( T \) is a characterization set for any \( F \subseteq T \subseteq P \).

Now we focus on characterization sets for balanced games. As Granot, Granot, and Zhu (1998) remark, in order to apply Theorem 3.6 we need not find an allocation in the nucleolus of the restricted game \( \Gamma^F \) if we know a superset \( Y \) of \( N(\Gamma^F) \) such that condition i. holds for all elements of \( Y \). Since core allocations provide non-negative satisfaction for all coalitions, the following corollary of Theorem 3.6 is easily seen.

**Corollary 3.8.** Let \( \Gamma \) be a cooperative (cost or profit) game with a non-empty core. The non-empty collection \( F \subseteq P \) is a characterization set for the nucleolus of \( \Gamma \) if for every \( S \in P \setminus F \) there exists a non-empty subcollection \( F_S \) of \( F \), such that

i. \( sat_T(T, x) \leq sat_T(S, x) \) for all \( x \in C(\Gamma) \), whenever \( T \in F_S \),

ii. \( e_S \) can be expressed as a linear combination of the vectors in \( \{e_T : T \in F_S \cup \{N\}\} \).
Now we present four characterization sets. The first is due to Huberman (1980).

**Definition 3.9 (Essential coalitions).** Let $N$ be a set of players, $(N, v)$ a profit, $(N, c)$ a cost game. Coalition $S \in \mathcal{P}$ is called *essential* in profit game $\Gamma = (N, v)$ if it cannot be partitioned as $S = S_1 \cup \cdots \cup S_k$ with $k \geq 2$ and $S_j \neq \emptyset$ for all $1 \leq j \leq k$ such that
\[
v(S) \leq v(S_1) + \cdots + v(S_k).
\]
Similarly, coalition $S \neq \emptyset$ is called essential in cost game $\Gamma = (N, c)$ if it cannot be partitioned as $S = S_1 \cup \cdots \cup S_k$ with $k \geq 2$ and $S_j \neq \emptyset$ for all $1 \leq j \leq k$ such that
\[
c(S) \geq c(S_1) + \cdots + c(S_k).
\]
The set of essential coalitions is denoted by $\mathcal{E}(\Gamma)$, where $\Gamma$ is either $(N, v)$ or $(N, c)$. A coalition that is not essential is called *inessential*.

Notice that we call only a non-trivial coalition essential, although $\emptyset$ would always qualify since it cannot be partitioned at all, and $N$ could also qualify in certain games. On the other hand, by definition, the singleton coalitions are always essential in every game. It is easily seen that in a profit / cost game each inessential coalition has a weakly majorizing / weakly minorizing partition which consists exclusively of essential coalitions. Moreover, the core is determined by the efficiency equation $\text{sat}_\Gamma(N, x) = 0$ and the $\text{sat}_\Gamma(S, x) \geq 0$ inequalities corresponding to the essential coalitions, all the other inequalities can be discarded from the core system.

**Theorem 3.10.** (Huberman, 1980) If the core of the game is non-empty then the essential coalitions form a characterization set for the nucleolus.

The theorem means that in a balanced game the essential coalitions and the grand coalition are sufficient to determine the nucleolus. This observation helps us to eliminate large coalitions which are redundant for the nucleolus. To detect small coalitions that are unnecessary for the nucleolus, we need the concept of dual game.

**Definition 3.11.** The *dual game* $\Gamma^* = (N, g^*)$ of game $\Gamma = (N, g)$ is defined by the coalitional function $g^*(S) := g(N) - g(N \setminus S)$ for all $S \subseteq N$, where $\Gamma$ is either $(N, v)$ or $(N, c)$. 
Notice that $g^*(\emptyset) = 0$, $g^*(N) = g(N)$, and $(g^*)^*(S) = g(S)$ for all $S \subseteq N$. It will be useful to think of the dual game of a profit game as a cost game and vice versa. It can be easily checked that $g$ is monotonic if and only if $g^*$ is monotonic, and $g$ is supermodular if and only if $g^*$ is submodular. However, there is no such dualization relation between superadditivity and subadditivity.

We can identify small redundant coalitions, if we apply Huberman’s argument to the dual game.

**Definition 3.12** (Dually essential coalitions). Let $N$ be a set of players, $(N, v)$ a profit, $(N, c)$ a cost game. Coalition $S \in \mathcal{P}$ is called dually essential in game $(N, v)$ if its complement cannot be partitioned as $N \setminus S = (N \setminus T_1) \cup \cdots \cup (N \setminus T_k)$ with $k \geq 2$ and $T_j \neq N$ for all $1 \leq j \leq k$ such that

$$v^*(N \setminus S) \geq v^*(N \setminus T_1) + \cdots + v^*(N \setminus T_k),$$

or equivalently,

$$v(S) \leq v(T_1) + \cdots + v(T_k) - (k - 1)v(N).$$

Similarly, $S \neq \emptyset, N$ is called dually essential in cost game $(N, c)$ if its complement cannot be partitioned as $N \setminus S = (N \setminus T_1) \cup \cdots \cup (N \setminus T_k)$ with $k \geq 2$ and $T_j \neq N$ for all $1 \leq j \leq k$ such that

$$c^*(N \setminus S) \leq c^*(N \setminus T_1) + \cdots + c^*(N \setminus T_k),$$

or equivalently,

$$c(S) \geq c(T_1) + \cdots + c(T_k) - (k - 1)c(N).$$

The set of dually essential coalitions is denoted by $\mathcal{DE}(\Gamma)$, where $\Gamma$ is either $(N, v)$ or $(N, c)$. A coalition that is not dually essential is called dually inessential.

Notice that we call only a non-trivial coalition dually essential, although the grand coalition would always, and the empty coalition could sometimes qualify. On the other hand, all $(n-1)$-player coalitions are dually essential in any game. If $S \in \mathcal{P}$ is dually inessential then it is contained in each of the coalitions $T_1, \ldots, T_k \in \mathcal{P}$ in the above expression, but every player in $N \setminus S$ appears exactly $k - 1$ times in this family. We call such a system of coalitions an overlapping decomposition of $S$. For a more general definition, where the complements of the overlapping
coalitions need not form a partition of the complement coalition, see e.g. Brânzei, Solymosi, and Tijs (2005) and the references therein.

It is clear that if $S, T \in P$ are dually inessential coalitions and $T$ appears in an overlapping decomposition of $S$, then $S \subset T$ so $S$ cannot appear in an overlapping decomposition of $T$. Consequently, in a profit / cost game each dually inessential coalition has a weakly majorizing / weakly minorizing overlapping decomposition which consists exclusively of dually essential coalitions. Moreover, the core of $\Gamma$ is also determined by the dual efficiency equation $\text{sat}_{\Gamma^*}(N,x) = 0$ and the $\text{sat}_{\Gamma^*}(S,x) \geq 0$ dual inequalities corresponding to the complements of the dually essential coalitions, all the other dual inequalities can be discarded from the dual core system.

The main feature of dually essential coalitions for the nucleolus is that together with the grand coalition they are sufficient to determine the nucleolus in balanced games. The next theorem is the dual counterpart of Theorem 3.10.

**Theorem 3.13.** If $C(\Gamma) \neq \emptyset$, then the grand coalition and the dually essential coalitions form a characterization set for $N(\Gamma)$.

**Proof.** There are various ways to derive this result. A formal proof can be obtained by applying the arguments of Huberman (1980) to the dual game. Here we pursue another way and deduce it from Corollary 3.8. We only give the proof for profit games, for cost games it is analogous.

Let $S \in P$ be a dually inessential coalition in the balanced profit game $\Gamma = (N,v)$. As remarked earlier, $S$ has a weakly majorizing overlapping decomposition $T_1, \ldots, T_k$ ($k \geq 2$) which consists exclusively of dually essential coalitions. Hence ii. of Corollary 3.8 follows immediately. To see condition i., let $x \in C(\Gamma)$. Then

$$v(S) \leq v(T_1) + \cdots + v(T_k) - (k - 1)v(N),$$
$$v(S) - x(S) \leq v(T_1) + \cdots + v(T_k) - (k - 1)x(N) - x(S),$$
$$-\text{sat}_{\Gamma}(S,x) \leq -(\text{sat}_{\Gamma}(T_1,x) + \cdots + \text{sat}_{\Gamma}(T_k,x)),$$
$$\text{sat}_{\Gamma}(S,x) \geq \text{sat}_{\Gamma}(T_1,x) + \cdots + \text{sat}_{\Gamma}(T_k,x),$$
$$\text{sat}_{\Gamma}(S,x) \geq \text{sat}_{\Gamma}(T_j,x) \geq 0 \quad j = 1, \ldots, k,$$

where the second inequality comes from $v(N) = x(N)$, while the third from the identity $x(T_1) + \cdots + x(T_k) = (k - 1)x(N) + x(S)$ implied by $N \setminus S = (N \setminus T_1) \cup$
3.3 Characterization sets for the nucleolus

\[ \cdots \cup (N \setminus T_k), \] and the last one from the non-negativity of the satisfaction values at any core allocation.

The next characterization set was proposed by Granot, Granot, and Zhu (1998) for balanced monotonic cost games.

**Definition 3.14** (Saturated coalitions). Let \((N, c)\) be a monotonic cost game. A coalition \(S \subseteq N\) is said to be saturated if \(c(S) = c(S \cup i)\) implies \(i \in S\).

In other words if \(S\) is a saturated coalition then every new member will impose extra cost on the coalition. Let \(S^*(\Gamma)\) denote the set of all saturated coalitions and

\[ S(\Gamma) = S^*(\Gamma) \cup \{N \setminus i \mid i \in N\} \cup \{N\}. \]

**Theorem 3.15.** (Granot, Granot, and Zhu, 1998) Let \(\Gamma = (N, c)\) be a monotonic cost game with a non-empty core. Then \(S(\Gamma)\) forms a characterization set for \(N(\Gamma)\).

In fact Granot, Granot, and Zhu (1998) proved a slightly stronger statement, namely, that the intersection of saturated and essential coalitions forms a characterization set in monotonic cost games with a non-empty core. Similarly to the other characterization sets, \(S(\Gamma)\) also induces a representation of the core \(C(\Gamma)\) as well. Let us mention here that just because a collection of coalitions determines the core it does not necessarily characterize the nucleolus of the game. Maschler, Peleg, and Shapley (1979) presented two games with the same core, but with different nucleoli.

We now convert the concept of saturatedness to monotonic profit games based on the dualization correspondence between profit and cost games.

**Definition 3.16** (Dually saturated coalitions). Let \((N, v)\) be a monotonic profit game and \(S \subseteq N\) be an arbitrary coalition. We say that \(S\) is dually saturated if \(v(S \setminus i) < v(S)\) for any \(i \in S\).

In other words every member contributes to the worth of coalition \(S\). Notice that \(S\) is saturated in the monotonic cost game \(c\) if and only if \(N \setminus S\) is dually saturated.
Computing the nucleolus

in the monotonic profit game $c^*$. Let $\mathcal{DS}^*(\Gamma)$ denote the set of all dually saturated coalitions and

$$\mathcal{DS}(\Gamma) = \mathcal{DS}^*(\Gamma) \cup \{i \mid i \in N\} \cup \{N\}.$$ 

The following definition is needed for our next theorem. Let $(N, v)$ be a monotonic game and $S \subseteq N$ a dually non-saturated coalition, then we say that $\underline{S} \neq \emptyset$ is a lower closure of $S$ if $\underline{S} \subset S$, $v(\underline{S}) = v(S)$ and $\underline{S}$ is a dually saturated coalition. Note that if $S$ has no lower closure, then no member contributes to the worth of $S$ or to any subset of $S$. Hence $v(S) = v(i) = v(\emptyset) = 0$ for any $i \in S$.

**Theorem 3.17.** Let $\Gamma = (N, v)$ be a monotonic game with a non-empty core, then $\mathcal{DS}(\Gamma)$ forms a characterization set for $\mathcal{N}(\Gamma)$.

**Proof.** Again we will use Theorem 3.6. Let $S$ be a dually non-saturated coalition. If $S$ has no lower closure then $v(S) = v(i) = 0$ for any $i \in S$. From this observation also follows that $sat_\Gamma(\{i\}, x) \leq sat_\Gamma(S, x)$ for any $i \in S$ and for any allocation $x$. Since all the singleton coalitions are included in $\mathcal{DS}(\Gamma)$ by Theorem 3.6, $S$ can be discarded. Finally, let $\underline{S}$ be a lower closure of $S$ and let $T = S \setminus \underline{S}$, then

$$sat_\Gamma(\underline{S}, x) + x(T) = x(\underline{S}) + x(T) - v(\underline{S}) = x(S) - v(S) = sat_\Gamma(S, x).$$

Since core vectors are non-negative this also means $sat_\Gamma(\underline{S}, x) \leq sat_\Gamma(S, x)$ for any $x \in \mathcal{C}(\Gamma)$. Now we show that $sat_\Gamma(\{i\}, x) \leq sat_\Gamma(S, x)$ for any $i \in T$.

$$sat_\Gamma(\{i\}, x) = x(i) - v(i) \leq x(i)$$

$$= x(S) - x(S \setminus i) + v(S \setminus i) - v(S)$$

$$= sat_\Gamma(S, x) - sat_\Gamma(S \setminus i, x) \leq sat_\Gamma(S, x)$$

We have shown that for any $S \in 2^N \setminus \mathcal{DS}(\Gamma)$ there exist a subcollection $\mathcal{F}$ of $\mathcal{DS}(\Gamma)$, such that $\mathcal{F}$ fulfills both conditions of Theorem 3.6. Hence $\mathcal{DS}(\Gamma)$ is a characterization set for $\mathcal{N}(\Gamma)$.

Next we show a relationship between dually essential and saturated coalitions.
Lemma 3.18. Let $\Gamma = (N, c)$ be a monotonic cost game, then $DE(\Gamma) \subseteq S(\Gamma)$

Proof. The grand coalition and the $n - 1$ player coalitions are all members of both $S(\Gamma)$ and $DE(\Gamma)$. Let $S$ be a non-saturated coalition with at most $n - 2$ players. We will show that $S$ is dually inessential. As $S$ is not saturated there exists $i \in N \setminus S$ such that $c(S) = c(S \cup i)$. Let $S_1 := S \cup i$ and $S_2 := N \setminus i$. Then $S_1 \cup S_2 = N$ and $S_1 \cap S_2 = S$ therefore we can use Definition 3.12 since

$$
c(N) \geq c(N \setminus i),
$$
$$
c(S) \geq c(S) + c(N \setminus i) - c(N),
$$
$$
c(S) \geq c(S_1) + c(S_2) - c(N).
$$

In other words $S$ appears in an overlapping decomposition of $S_1$ and $S_2$, therefore it cannot be dually essential.

The above lemma suggests that dually essential coalitions are more useful since they define a smaller characterization set, which in turn implies a smaller LP. However, usually it is also harder to determine whether a coalition is dually essential or not. Saturatedness on the other hand can be checked easily. For instance for airport games\footnote{This class of games was introduced by Littlechild and Owen (1973).} there exist at most $n$ saturated coalitions, which can be easily determined from the characteristic function. In fact it is enough to know the value of the singleton coalitions to identify the saturated coalitions, which gives us an alternative way to derive an efficient algorithm for the nucleolus.

There is a symmetrical result for essential and dually saturated coalitions.

Lemma 3.19. Let $\Gamma = (N, v)$ be a monotonic profit game, then $E(\Gamma) \subseteq DS(\Gamma)$.

Proof. Observe that the singleton coalitions are all members of both $E(\Gamma)$ and $DS(\Gamma)$. Let $S$ be a dually non-saturated coalition such that $|S| > 1$. Then there exists $i \in S$ such that $v(S) = v(S \setminus i)$. By monotonicity $v(i) \geq 0$, hence $v(S) \leq v(S \setminus i) + v(i)$. Thus $S$ is inessential.

If we look for an as small characterisation set as possible for balanced games, in light of Theorems 3.10 and 3.13 it seems natural to eliminate both inessential coalitions and dually inessential coalitions at the same time. However, this might
not be possible, the intersection of essential and dually essential coalitions need not yield a characterization set. For example, let $\Gamma = (N, v)$ be an additive game, that is $v(S) = \sum_{i \in S} v(i)$ for all $S \subseteq N$. Then only the singleton coalitions are essential, and since $v^* = v$, only the $(n - 1)$-player coalitions are dually essential. Thus $\mathcal{E}(\Gamma) \cap \mathcal{D}E(\Gamma) = \emptyset$ cannot possibly be a characterization set. The problem lies in what we call a cycle in the decomposition. This occurs when coalition $S$ can be discarded because of a collection that contains coalition $T$, and $T$ can also be discarded because of a collection that contains $S$ or another coalition $R$ that can be eliminated because of a collection that contains $S$, etc. As remarked after both definitions, such a cycle in the decomposition cannot occur if we apply only essentiality or only dual essentiality. Before we show when the two types of concepts can be mixed we need some preparation.

Let us extend the previously introduced balancedness concept. A collection of coalitions $\mathcal{B}_S \subseteq \mathcal{P}$ is said to be $S$-balanced if there exist positive weights $\lambda_T$ for $T \in \mathcal{B}_S$, such that $\sum_{T \in \mathcal{B}_S} \lambda_T e_T = e_S$. An $N$-balanced collection is simply called balanced. A coalition $S$ is called vital if for any $S$-balanced collection $\mathcal{B}_S$ and any system $(\lambda_T)_{T \in \mathcal{B}_S}$ of balancing weights for $\mathcal{B}_S$, $\sum_{T \in \mathcal{B}_S} \lambda_T v(T) < v(S)$. Trivially, every vital coalition is essential, but the converse does not hold. The concept was introduced by Gillies (1959) and further analyzed by Shellshear and Sudhölter (2009). It is easily seen that the family of vital coalitions (just like essential coalitions) and the grand coalition are sufficient to determine the core. However, Maschler, Peleg, and Shapley (1979) demonstrated that vital coalitions (unlike essential coalitions) and the grand coalition do not necessarily characterize the nucleolus in a balanced game.

The next theorem provides a sufficient condition for $\mathcal{E}(\Gamma) \cap \mathcal{D}E(\Gamma)$ to be a characterization set for the nucleolus in a balanced game.

**Theorem 3.20.** Let $\Gamma = (N, v)$ be a game with a non-empty core. If the grand coalition is vital, the collection $\mathcal{E}(\Gamma) \cap \mathcal{D}E(\Gamma)$ forms a characterization set of $N(\Gamma)$.

**Proof.** Suppose coalition $S$ is redundant, that is, either inessential or dually inessential (or both). Then there is a series of coalitions $S_1, \ldots, S_k$ in $\mathcal{E}(\Gamma)$ or in $\mathcal{D}E(\Gamma)$

---

36

---

7 Strongly essential coalitions discussed by Brânzei, Solymosi, and Tijs (2005) are not immune to this kind of failure, hence they might not form a characterization set for the nucleolus in a balanced game in which the grand coalition is not vital.
that have smaller satisfaction value than $S$ and whose membership vectors span $e_S$. Lexicographically maximizing the satisfactions of $S_1, \ldots, S_k$ the satisfaction of $S$ becomes fixed. If there is no cycle in the decomposition then there exists a decomposition of $S_i$ for $i = 1, \ldots, k$ that consist entirely of coalitions $T_1^i, \ldots, T_r^i$ that belong to $\mathcal{E}(\Gamma) \cap \mathcal{DE}(\Gamma)$. Lexicographically maximizing the satisfactions of $T_1^i, \ldots, T_r^i$ the satisfaction of $S_i$ becomes fixed. Thus indirectly the satisfaction of $S$ becomes fixed. We conclude that, if there is no cycle in the decomposition, then by Theorem 3.6 $\mathcal{E}(\Gamma) \cap \mathcal{DE}(\Gamma)$ is a characterization set.

By contradiction suppose that the grand coalition is vital, but there is a cycle in the decomposition caused by coalitions $T_1, T_2, \ldots, T_r$. Note that we may assume without loss of generality that the series alternates between inessentiality and dual inessentiality. If $T_\ell$ and $T_{\ell+1}$ were deemed redundant for the same reason (e.g. they are both inessential) then the inequality that shows inessentiality of $T_\ell$ can be refined by the inequality that shows inessentiality of $T_{\ell+1}$. Let us assume that $T_1$ is inessential – the proof is the same if $T_1$ is dually inessential. Thus using the definition of essentiality and dual essentiality

$$v(T_1) \leq v(T_2) + \sum_{j=1}^{k_1} v(S_1^j), \quad (3.2)$$

$$v(T_2) \leq v(T_3) + \sum_{j=1}^{k_2} v(S_2^j) - k_2 \cdot v(N), \quad (3.3)$$

$$v(T_3) \leq v(T_4) + \sum_{j=1}^{k_3} v(S_3^j), \quad (3.4)$$

$$\vdots$$

$$v(T_r) \leq v(T_1) + \sum_{j=1}^{k_r} v(S_r^j) - k_r \cdot v(N), \quad (3.5)$$

In words $T_1$ is inessential because of the collection $T_2, S_1^1, \ldots, S_1^{k_1}$ (these are all essential coalitions, thus the inequality cannot be refined any more). Then $T_2$ is dually inessential because of the collection $T_3, S_2^2, \ldots, S_2^{k_2}$ compose an overlapping decomposition of $T_2$ (and these are all dually essential). And so on until finally $T_r$ is deemed redundant because of $T_1, S_1^n, \ldots, S_r^{k_r}$. Note that there may be coalitions
among \( S^1_1, \ldots, S^2_1, \ldots, S^r_1, \ldots, S^r_{k_r} \) that coincide. Using indicator functions and the conditions of inessentiality and dual inessentiality.

\[
e_{T_1} = e_{T_2} + \sum_{j=1}^{k_1} e_{S^1_j}, \tag{3.6}
\]

\[
e_{T_2} = e_{T_3} + \sum_{j=1}^{k_2} e_{S^2_j} - k_2 \cdot e_N, \tag{3.7}
\]

\[
e_{T_3} = e_{T_4} + \sum_{j=1}^{k_3} e_{S^3_j}, \tag{3.8}
\]

\[
\vdots
\]

\[
e_{T_r} = e_{T_1} + \sum_{j=1}^{k_r} e_{S^r_j} - k_r \cdot e_N, \tag{3.9}
\]

Thus by summing (3.2)-(3.5) we obtain that

\[
v(N) \leq \frac{1}{k_2 + k_4 + \cdots + k_r} \sum_{i=1}^{r} \sum_{j=1}^{k_i} v(S^i_j),
\]

while from (3.6)-(3.9) we gather that

\[
e_N = \frac{1}{k_2 + k_4 + \cdots + k_r} \sum_{i=1}^{r} \sum_{j=1}^{k_i} e_{S^i_j},
\]

i.e. the collection \( S^1_1, \ldots, S^2_1, \ldots, S^r_1, \ldots, S^r_{k_r} \) is balanced. This contradicts the fact that the grand coalition is vital. \( \square \)
Chapter 4

Bankruptcy games

This chapter is partly based on (Fleiner and Sziklai, 2012), while the last section relies on (Solymosi and Sziklai, 2015).

4.1 A riddle in the Talmud

The bankruptcy problem is one of the oldest in the history of economics. In its simplest form, we have a bankrupt firm, and creditors that wish to collect their claims. The total demand exceeds the firm’s liquidation value. A natural question is how to divide this value among the claimants. Depending on the notion of fairness, one can impose many rules for such a division.

The Talmud is a 2000-year old document that forms the basis for Jewish civil, criminal and religious law. It introduces the problem of bankruptcy with the following example. A man who had three wives dies. According to the marriage contracts he owes 100, 200, and 300 zuz (contemporary currency) to them respectively. The Talmud explains how the estate should be divided among them by analyzing three different cases (See Table 4.1).

It is clear that in the first case – when the estate worth only 100 zuz – equal division was ruled, and the estate was divided proportionally in the third case. However, in the second case the ruling is neither equal nor proportional. As the original concept was forgotten, new and new interpretations of the Mishna\footnote{The specific section of the Talmud.} were
born. Rabbis and economists argued literally for centuries. Many questioned that there is a universal rule behind these cases. A few even suggested that there is an error in the transcription (Lewy, 1908). Finally, in their seminal paper Aumann and Maschler (1985) unraveled the mystery. They not only showed there is a common rationale behind these cases, but they also proved that the Talmud-rule coincides with the nucleolus of the corresponding bankruptcy game. Of course it is unlikely that the sages who wrote the Mishna were familiar with game theoretical concepts. Therefore Aumann and Maschler (1985) also provided justifications that were well within the reach of the rabbis.

### 4.2 Formal definition of bankruptcy games

Let \( N = \{1, 2, \ldots, n\} \) be the set of creditors. The bankruptcy problem\(^2\) is defined as a pair \( (d, E) \) where \( E \in \mathbb{R}^+ \) represents the firm’s liquidation value (or estate/endowment) and \( d \in (\mathbb{R}^+)^n \) is the collection of claims with \( \sum_{i=1}^n d_i > E \). Again we employ the shorthand notation \( d(S) = \sum_{i \in S} d_i \). Let \( \mathcal{B} \) denote the class of such problems. A solution of a bankruptcy problem is a vector \( x \in (\mathbb{R}_0^+)^n \) such that \( \sum_{i=1}^n x_i = E \). A rule \( r : \mathcal{B} \to \mathbb{R}^n \) is a mapping that assigns a unique solution to each bankruptcy problem.

The characteristic function corresponding to the bankruptcy problem \( (d, E) \) is

\[
v_{(d,E)}(S) = \max\{ E - d(N \setminus S), 0 \}
\]

By definition, this is the value that is left from the firm’s liquidation value \( E = v_{(d,E)}(N) \) after the claim of each player of the complement coalition \( N \setminus S \) has been satisfied. Coalition \( S \) can achieve \( v_{(d,E)}(S) \) without any effort. Note that \( v_{(d,E)} \)

\(^2\)Sometime it is referred as the claims or rationing problem.

<table>
<thead>
<tr>
<th>Size of the estate</th>
<th>Wife A</th>
<th>Wife B</th>
<th>Wife C</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>33(\frac{1}{3})</td>
<td>33(\frac{1}{3})</td>
<td>33(\frac{1}{3})</td>
</tr>
<tr>
<td>200</td>
<td>50</td>
<td>75</td>
<td>75</td>
</tr>
<tr>
<td>300</td>
<td>50</td>
<td>100</td>
<td>150</td>
</tr>
</tbody>
</table>

Table 4.1: Marriage contract rulings according to the Talmud (Kethubot 93a)
is non-negative and supermodular. Hence bankruptcy games are convex, which implies that they are superadditive and monotonic as well. Now we formalize some rules and their properties that we will need later on.

The dual of a rule \( r \) (denoted by \( r^* \)) assigns awards in the same way as \( r \) assigns losses, namely \( r^*(d, E) = d - r(d, d(N) - E) \). A self-dual rule is one with \( r^* = r \), such rule treats losses and awards the same way.

The constrained equal awards (CEA) rule assigns equal awards to each agent subject to no one receiving more than his claim. The dual of this rule is the constrained equal losses (CEL) rule. In this case losses are distributed as equally as possible subject to no one receives a negative amount. Formally:

**Constrained equal awards:** For all \( (d, E) \in \mathbb{B} \) and \( i \in N \), \( CEA_i(d, E) = \min(d_i, \lambda) \) where \( \lambda \) solves \( \sum_{i=1}^{n} \min(d_i, \lambda) = E \).

**Constrained equal losses:** For all \( (d, E) \in \mathbb{B} \) and \( i \in N \), \( CEL_i(d, E) = \min(0, d_i - \lambda) \) where \( \lambda \) solves \( \sum_{i=1}^{n} \min(0, d_i - \lambda) = E \).

Another well-known rule is the random arrival rule. Suppose the claims arrive sequentially and they are fully compensated until the money runs out. The random arrival rule computes award vectors for every possible order of claims and then takes their average. Hence it produces the same award vector as the Shapley-value applied to the corresponding bankruptcy game O’Neill (1982).

The Contested Garment principle is a division formula that can be derived from the Talmud. There are two claimants and a divisible piece of good. According the principle, each claimant should give the part of the good that he does not contest to the other claimant. Then the rest is split up equally.

![Contested Garment Principle](image)

**Figure 4.1:** The Contested Garment Principle

Note that the CG-principle does not distinguish between claims that exceed the size of the estate. Claims that are larger than the estate are treated as if they were 'only' claiming the whole and not more.
**Definition 4.1.** Let \( r(d, E) \) be a division rule defined on each \((d, E) \in \mathbb{B}\). We say that \( r \) is pairwise consistent if for all \( N' \subset N \), where \(|N'| = 2\), if \( x \equiv r(d, E) \), then \( x_{N'} = r(d_{N'}, \sum_{N} x_i) \).

The stronger version, \textit{consistency}\(^3\) is obtained by dropping the restriction \(|N'| = 2\).

For more on this topic see (Hokari and Thomson, 2008) and (Young, 1987). We refer to the pairwise consistent rule that divides the endowment between any two agent by the Contested Garment Principle as the \textit{CG-consistent solution}.

**Definition 4.2.** Let \((d, E)\) be a bankruptcy problem. A solution is called \textit{CG-consistent}, if for all \( i \neq j \) the division of \( x(i) + x(j) \) prescribed by the contested garment principle for claims \( d_i, d_j \) is \((x(i), x(j))\).

**Theorem 4.3.** (Aumann and Maschler, 1985) Each bankruptcy problem \((d, E)\) has a unique \textit{CG-consistent solution}, which is given by the following formula

\[
T_i(d, E) = \begin{cases} 
\min \{d_i/2, \lambda\} & \text{if } E \leq \frac{1}{2}d(N), \\
\max \{d_i/2, d_i - \mu\} & \text{if } E > \frac{1}{2}d(N),
\end{cases}
\]

where \( \lambda \) and \( \mu \) are chosen so that \( \sum_{i \in N} T_i(d, E) = E \).

Observe that the \textit{CG-consistent rule} is the combination of the constrained equal awards and the constrained equal losses rules. In fact it is a self-dual rule that produces the same estate division as the examples in the Talmud. For this reason it is also called the Talmud-rule.

### 4.3 Literature overview

The rationing problem and the concept of bankruptcy game were introduced in (O’Neill, 1982). This influential paper spawned a tremendous amount of literature, especially after the discovery made by Aumann and Maschler (1985). Despite its (mathematical) simplicity the bankruptcy game captures the essence of fair division problems. It is one of most popular topics in cooperative game theory, hence it would be a futile attempt to list all the relevant papers. Nevertheless we will try to outline the main research directions.

\(^3\)In the paper of Aumann and Maschler (1985) this property is called \textit{self-consistency}.
Some researchers focused on developing new solution concepts, while others worked on the axiomatic foundations of the rationing problem. Some stuck to the original model and presented new variants of the Talmud-rule, while others extended the model itself. Of course these research directions are intertwined with each other. We will try to give examples for all of them. For a comprehensive survey see (Thomson, 2003) and (Thomson, 2015).

Herrero, Maschler, and Villar (1999) introduced the so called rights-egalitarian solution. This is an extension of the divorce rule that is similar to CEL, but allows players to have negative entitlement. They also showed that if we reinterpret the bankruptcy situation, and define the characteristic function in a slightly different way then the rights-egalitarian solution coincides with the Shapley-value and the prenucleolus of that particular game. Chun, Schummer, and Thomson (2001) on the other hand suggested the constrained egalitarian rule, which is closely related to CEA. In fact if the estate is less than or equal to half of the claims it distributes the money in the same way. If the estate is larger, then it gradually dispenses the remaining part of the estate by giving the money to the players with the smallest claims. Moreno-Ternero and Villar (2006) generalized the Talmud-rule and introduced a family of rules by combining the CEA and CEL-rule in an arbitrary ratio. In a more recent paper Huijink, Borm, Reijnierse, and Kleppe (2013) provided an algorithm for the per capita nucleolus in case of bankruptcy games. They showed that the per capita nucleolus is also a combination of the CEA and CEL-rules, more precisely some part of the claims is distributed by CEA and the rest is by CEL.

Most of the above listed papers contain axiomatic justifications as well. Herrero and Villar (2001) compares the CEA, CEL the proportional and the Talmud rule from axiomatic point of view. Another characterization of the Talmud rule was provided by Moreno-Ternero and Villar (2004), while the so called reverse Talmud rule (cf. Chun, Schummer, and Thomson (2001)) was analyzed by van den Brink, Funaki, and van der Laan (2013). Further axiomatic research, that focused on the different versions of consistency was provided by Hokari and Thomson (2008).

Moulin and Sethuraman (2013) extend the standard rationing context to bipartite context. That is, there are more than one bankrupt firms and players can have different claims toward different firms. The model can be easily depicted by a bipartite graph, where the two node sets represent creditors and estates, while edges
Bankruptcy games

represent claims. İlkilici and Kayi (2012) use the same setup and provide an algorithm for the egalitarian and proportional solution. Another graph-based concept appears in (Bjørndal and Jörnsten, 2010), where the bankruptcy game is generalized to flow sharing games. Casas-Méndez, Fragnelli, and García-Jurado (2011) introduce weighted bankruptcy games and use the idea to solve the museum-pass problem, that is how to share the profit of a pass that is valid for several museums.

Finally, let us mention a methodical novelty that revolutionized the visualisation of the bankruptcy problem. Kaminski (2000) used a hydraulic framework (vessels and water) to represent the claims and the estate. More and more authors borrow the concept to illustrate their result, see for instance (Aumann, 2003; Fleiner and Sziklai, 2012; Huijink, Born, Reijnierse, and Kleppe, 2013; Timoner and Izquierdo, 2014; Gonczarowski and Tennenholtz, 2014). In the next section we will show that the hydraulic representation not only helps to interpret notions but it is an effective tool in developing new concepts and proving the right statements.

4.4 Hydraulic rationing

The result of Aumann and Maschler (1985) – that the Talmud rule is in fact the nucleolus of the corresponding bankruptcy game – is a fascinating discovery and a landmark of cooperative game theory, but it came with a somewhat complicated proof. It used complex notions like the reduced game and the kernel. Later Benoît (1997) gave a proof that used only elementary operations but it was still long and technical, therefore the desire to obtain a simple proof for Aumann and Maschler’s result remain. Here we show a short proof employing Kaminski’s idea and construct a specific hydraulic for this reason.

In this hydraulic, every claim is represented by a vessel while the firm’s liquidation value corresponds to the amount of water we pour into this system. Our vessels have a peculiar hourglass-shape with the following characteristics:

4 Proof techniques that use the principles of mechanics were very common in the ancient times. Archimedes wrote to Eratosthenes: "I thought it to write out for you and explain in detail... a certain method, by which it will be possible for you... to investigate some of the problems in mathematics by means of mechanics. This procedure is, I am persuaded, no less useful even for the proof of the theorems themselves...". The increasing number of examples in the literature shows that these are no less useful today. Just to mention some well-known instances: the shortest path in a directed graph can be found using a system of strings and knots, congestion games can be represented by electric circuits, and - as in our case - rationing problems can be modeled using hydraulic systems.
4.4 Hydraulic rationing

- Each vessel has an upper and lower tank of a shape of a cylinder.
- The upper and lower tanks have the same volume and they are connected with a capillary.
- The capillaries have negligible volume.
- The cylinders have a circular base with area of 1.
- The volume of vessel $i$ is equal to the size of agent $i$’s claim.
- Finally, each vessel has the same height denoted by $h$. We may assume $h = d_{\text{max}}$ where $d_{\text{max}}$ denotes the largest claim.

Note that the last condition implies that the vessel with the largest volume has no capillary part. We say that a hydraulic is talmudic if it incorporates the above characteristics.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{connected_talmudic_hydraulic.png}
\caption{A connected talmudic hydraulic}
\end{figure}

It is not included in the above list but for a proper representation every vessel has to have also a capillary on the top tank where the air can leave the vessel. During the proof, we will need two types of vessel systems (or hydraulics). In a connected hydraulic, vessels are connected with capillaries at the bottom. This way, if we pour water into any of the vessels each vessel starts to fill up. In case of a disconnected hydraulic, vessels are not connected by capillaries. Thus different vessels can have different ’water levels’ depending on how much water we pour into them.

We say that a hydraulic $\mathcal{H}$ corresponds to a bankruptcy problem $(d, E)$ if the following conditions hold

- $\mathcal{H}$ has $n$ vessels
• the volume of the \( i \)th vessel is equal to the size of the claim of agent \( i \)

• there is \( E \) amount of water distributed among the vessels.

A hydraulic that corresponds to a bankruptcy problem always implicitly defines an allocation rule. The nature of the rule depends on the shape of the vessels. Here we show the representation of the rules we have already mentioned.

\[\text{Observation 1. (Kaminski, 2000)}\]

Let \((d, E)\) be a two-person bankruptcy problem and let \(\mathcal{H}\) be the corresponding connected talmudic hydraulic. The solution \(x\) defined by the common water level \(z\) in \(\mathcal{H}\) is CG-consistent.

Indeed, it is not hard to check that the common water level \(z\) induces the same allocations as the CG-principle. For a detailed explanation the reader is referred to (Aumann, 2003) or (Kaminski, 2000). To further illustrate the robustness of Kaminski’s method, we observe an interesting property of consistent solutions (cf. Definition 4.1 and the remark that followed it).

\[\text{Observation 2. A rule is consistent if and only if it corresponds to a connected hydraulic in which the shape of a vessel depends only on the respective claim size.}\]

Consistency of a rule means that a connected hydraulic corresponds to the same rule after removing some of its vessels. Note that the random arrival rule is an example of a rule that is not consistent. Since if a claim disappears then the shape of the remaining vessels has to change. Another remarkable feature of the hydraulic approach is that it characterizes self-duality in a succinct way.

\[\text{Observation 3. The Talmud rule is a self-dual one.}\]

Let \(\mathcal{H}\) be a connected talmudic hydraulic with \(E\) amount of water in it. We have to show that the corresponding rule \(T\) is a self-dual rule, or formally, that
4.4 Hydraulic rationing

\[ T^*(d, E) = d - T(d, d(N) - E). \]

We can translate this into the language of hydraulics. Let \( x = (x_1, x_2, \ldots, x_n) \) be the solution induced by the common water level. Now consider a copy of \( \mathcal{H} \) which is fully filled with water. Let \( y = (y_1, y_2, \ldots, y_n) \) be the distribution of air when we let out \( E \) amount of water from the fully filled hydraulic. As the figure shows the distributions are the same.

![Figure 4.4: The hydraulic \( \mathcal{H} \) with \( E \) and \( d(N) - E \) amount of water in it.](image)

Actually more is true. Easy to conclude the following fact.

**Observation 4.** A rule is self-dual if and only if it corresponds to some horizontally symmetric connected hydraulic.

Recall that the satisfaction of coalition \( S \) is \( \text{sat}(S, x) = x(S) - v(d,E)(S) \). If a coalition \( S \) has nothing after all other claimants outside the coalition have been paid off then its satisfaction will be \( x(S) \). Otherwise the gain of \( S \) should be decreased by \( v(d,E)(S) \) since \( S \) would get \( v(d,E)(S) \) anyway.

We need two small observations. Let \( \mathcal{H} \) be a disconnected hydraulic. Now \( x(S) \) is the amount of water that is distributed among the vessels that belong to coalition \( S \). The next lemma shows that the satisfaction of coalition \( S \) is the minimum of the following two amounts: the water contained in \( S \) and the air contained in \( N \setminus S \).

**Lemma 4.4.** Let \( v(d,E) \) be a bankruptcy game on player set \( N \) and \( x \) an imputation. The satisfaction of \( S \subseteq N \) can be written as

\[ \text{sat}(S, x) = \min \left( x(S), d(N \setminus S) - x(N \setminus S) \right). \] (4.1)
Proof. We apply the definition:

\[
\text{sat}(S, x) = x(S) - v_{(d,E)}(S) = x(S) - \max(0, E - d(N \setminus S))
\]

\[
= x(S) + \min(0, -E + d(N \setminus S)) = \min(x(S), x(S) - E + d(N \setminus S))
\]

\[
= \min(x(S), x(S) - x(N) + d(N \setminus S)) = \min(x(S), d(N \setminus S) - x(N \setminus S))
\]

Our second observation is that the CG-consistent solution coincides with the nucleolus for two-person bankruptcy games. We prove this with the following lemma.

**Lemma 4.5.** If \((d, E)\) is a two-person bankruptcy game with \(N = \{1, 2\}\) and \(x\) is a solution of it then \(\text{sat}(1, x) + \text{sat}(2, x) = E - (v_{(d,E)}(1) + v_{(d,E)}(2))\). This means that the sum of the satisfaction values of the two players is exactly the doubly claimed amount (i.e. the part of estate that is not concealed by any of the agents).

**Proof.** By definition, \(\text{sat}(1, x) + \text{sat}(2, x) = x(1) - v_{(d,E)}(1) + x(2) - v_{(d,E)}(2) = x(1) + x(2) - (v_{(d,E)}(1) + v_{(d,E)}(2)) = E - (v_{(d,E)}(1) + v_{(d,E)}(2))\). As we have seen, \(v_{(d,E)}(i)\) is that part of the estate that the other player does not claim, the right hand side is the exactly the doubly claimed part.

**Lemma 4.6.** If \((d, E)\) is a two-person bankruptcy game then its CG-consistent solution is the nucleolus of the corresponding coalitional game. Moreover, if we change a non-CG-consistent solution \(x\) in such a way that it becomes closer to the CG-consistent one then the satisfaction vector \(\theta(x)\) is lexicographically increasing.

**Proof.** As \(\text{sat}(\emptyset, x) = \text{sat}(N, x) = 0\) for any allocation \(x\) of the game, the nonzero coordinates of \(\theta(x)\) are the satisfaction values of the single players. As the sum of these satisfaction values are constant by Lemma 4.5, the satisfaction vector is lexicographically increasing if we decrease the share of the player with the greater satisfaction and increase the share of the player with the smaller satisfaction. This
means that the nucleolus is that solution where the satisfaction values of the two players are equal, that is, where the doubly claimed part is halved. This is exactly the CG-consistent solution.

Now we are ready to formulate the elementary proof we promised (as appeared in Fleiner and Sziklai (2012)).

**Theorem 4.7.** (Aumann and Maschler, 1985) The CG-consistent solution of a bankruptcy problem is the nucleolus of the corresponding coalitional game.

**Proof.** Let \( t \) denote the CG-consistent solution of the bankruptcy game corresponding to \( (d, E) \) and assume \( x \neq t \) for some solution \( x \) of the game. Note that \( t \) also corresponds to a common water level in the connected talmudic hydraulic. We shall show that there is another solution \( x' \) lexicographically greater than \( x \) such that \( x' \) and \( t \) agrees in more coordinates than \( x \) and \( t \). Clearly, this immediately implies Theorem 4.7.

As \( x \) and \( t \) are different solutions, there are players \( i \) and \( j \) such that \( x_i > t_i \) and \( x_j < t_j \). Construct solution \( x' \) from \( x \) such that we start to decrease \( i \)'s share and increase \( j \)'s share until we reach \( x'_i = t_i \) or \( x'_j = t_j \). (In other words, if we start from disconnected hydraulic representing \( x \) then we let some water from vessel \( i \) to vessel \( j \) until one of the water levels reaches the water level of \( t \) that is strictly between the two original levels.) Clearly, \( x' \) agrees with \( t \) on more coordinates than \( x \) did.

Observe that if \( sat(S, x) \neq sat(x', S) \) then either \( i \in S \not\ni j \) and \( sat(x', S) < sat(S, x) \) or \( i \not\in S \ni j \) and \( sat(x', S) > sat(S, x) \). Let \( S_{i,j} \) denote the set of coalitions that contain \( j \) but not \( i \). Therefore, to show that \( \theta(x') > \theta(x) \), it is enough to prove that any decreased satisfaction is at least as great as the smallest increased satisfaction, that is

\[
\min_{S \in S_{i,j}} sat(x', S) \geq \min_{S \in S_{j,i}} sat(x', S). \tag{4.2}
\]

If we think about the disconnected hydraulic that corresponds to \( x' \) then the satisfaction is either the amount of water in \( S \), namely \( x(S) \) or the amount of air in the complement coalition \( N \setminus S \), namely \( d(N \setminus S) - x(N \setminus S) \). By Lemma 4.4, this is the minimum of these two amounts. Observe that the smaller the coalition
Bankruptcy games

$S$ the smaller the value $x(S)$. Moreover, the bigger the coalition $S$ the smaller the value $d(N \setminus S) - x(N \setminus S)$. Therefore looking at the coalitions in $S_{i,j}$, either $i$ or $N \setminus j$ has the minimum satisfaction. Similarly, looking at coalitions $S_{j,i}$, either $j$ or $N \setminus i$ has the minimum satisfaction. Formally, it is an exchange of minima and an application of Lemma 4.4, as follows

$$
\min_{S \in S_{i,j}} \text{sat}(x', S) = \min_{S \in S_{i,j}} \{ \min\{x(S), x(N \setminus S) - d(N \setminus S)\} \}
$$

$$
= \min\{ \min_{S \in S_{i,j}} \{x(S)\}, \min_{S \in S_{i,j}} \{x(N \setminus S) - d(N \setminus S)\}\} = \min\{x'(i), d(j) - x'(j)\}
$$

A similar calculation shows that

$$
\min_{S \in S_{j,i}} \text{sat}(x', S) = \min\{x'(j), d(i) - x'(i)\}
$$

To finish the proof, we have to show that

$$
\min\{x'(i), d(j) - x'(j)\} \geq \min\{x'(j), d(i) - x'(i)\}. \tag{4.3}
$$

These minima depend only on the payoff of $i$ and $j$, no other coalitions are involved. Since we have a disconnected hydraulic we may forget about the other vessels for the moment and concentrate on the vessel system of $i \cup j$. In this context (4.3) is equivalent to saying that $\text{sat}(x', i) \geq \text{sat}(x', j)$. We selected $x'$ in such a way that $x'(i)$ is not less than the CG-consistent share of $i$ and $x'(j)$ is not more than the CG-consistent share of $j$. We have seen that the Talmud rule is consistent and for the restricted game this implies that $i$ gets at least as much as her CG-consistent share. By Lemma 4.6, the CG-consistent solution of the restricted game gives the same satisfaction to both players, hence the satisfaction of $i$ is not less in this game than that of $j$, and this is exactly what we had to prove.

To illustrate the power of the hydraulic approach, we repeat the somewhat tricky argument of the last paragraph in the language of the vessels. Consider disconnected hydraulic corresponding to $x$. As $i$ has more and $j$ has less than their respective CG-consistent shares, the water level according to $x$ is higher in the $i$th vessel than in the $j$th one.

To obtain solution $x'$ from $x$, we let some water from vessel $i$ into vessel $j$. We stop as soon as any of the levels reach the common water level of the CG-consistent
solution. Now $x'$ still has the property that the water is higher in vessel $i$ than in vessel $j$. Lemma 4.6 shows that equal water level means equal satisfaction values in two vessel hydraulics. Therefore the $x'$-satisfaction of $i$ is more than the $x'$-satisfaction of $j$ for the two vessel hydraulic, just as we claimed.

Though the above proof works without introducing the hydraulic framework, some statements, like observation 2 and 4 are more transparent this way. The main advantage of hydraulics is that it makes easier to interpret notions like self-duality and consistency and helps to find and prove the right statements. Moreover we can easily design an algorithm by simulating the talmudic hydraulic system.

INPUT: a set of agents $N = \{1, 2, \ldots, n\}$ a claims vector $d$ and estate size $E$.
(1) If $E \leq \frac{d(N)}{2}$ distribute awards otherwise distribute losses.
(2) Sort the claims in non-decreasing order, i.e. $(d_1 \leq d_2 \leq \cdots \leq d_n)$
(3) For $i = 1, 2, \ldots, n$
   - If $|N| \cdot \frac{d_i}{2} \geq E$ then $x_i := \frac{E}{|N|}$ for $i \in N$. (go to $\rightarrow$ (4))
   - Otherwise $x_i := \frac{d_i}{2}$; $N := N \setminus \{i\}$; $E := E - x_i$
(4) If losses were distributed then $x_i := d_i - x_i$ for $i = 1, 2, \ldots, n$.
OUTPUT: an imputation $x$ that is the nucleolus of the corresponding game.

Sorting the claims clearly is the most time consuming operation. All the other steps can be executed in linear time. Hence we obtained an algorithm of time complexity $O(n \log n)$.

We conclude this section with a somewhat surprising consequence of Theorem 4.7 that falls out from the hydraulic representation. The talmudic vessels have two cylinders of equal size. A straightforward question is what kind of allocation rule
do we obtain if we increase the number of (equally sized) cylinders while we leave
the size of the estate fixed?

\[ \text{Figure 4.7: Hydraulics with } k \text{ equally sized cylinders for } k = 2, 3, 4 \]

Let \((d, E)\) be a bankruptcy problem and \(i, j \in N\) two arbitrary players. Let \(d_i = m\)
and \(d_j = \alpha m\) for some \(\alpha > 0\) and let the size of the biggest claim be \(d_n = M\).
Consider a hydraulic where each vessel have the same height \(h = M\) and have
\(k\) cylinders of equal size. The cylinders are connected by capillaries of the same
length (see Figure 4.7). That is, for \(k = 2\) we have the Talmudic-hydraulic. The
vessel of player \(i\) starts with a cylinder of size \(\frac{m}{k}\) and followed by a capillary of
length \(\frac{M - m}{k - 1}\) then cylinders and capillary parts alternate in the same way. Similarly
the second player’s vessel consist of cylinders of \(\frac{\alpha m}{k}\) size and capillaries of \(\frac{M - \alpha m}{k - 1}\)
length. Notice that if \(k\) is big then

\[
\frac{m}{k} + \frac{M - m}{k - 1} \approx \frac{\alpha m}{k} + \frac{M - \alpha m}{k - 1}.
\]  

(4.4)

In other words a cylinder together with a capillary part is approximately of the
same height for both players. To prove this let us fix \(M = tk\) and let \(m = \beta M =
\beta tk\) for some \(\beta \in [0, 1]\). Then

\[
\lim_{k \to \infty} \frac{m}{k} + \frac{M - m}{k - 1} - \left( \frac{\alpha m}{k} + \frac{M - \alpha m}{k - 1} \right) \\
= \lim_{k \to \infty} \beta tk + \frac{tk - \beta tk}{k - 1} - \frac{\alpha \beta tk}{k} - \frac{tk - \alpha \beta tk}{k - 1} \\
= \lim_{k \to \infty} \beta t - \alpha \beta t + (\alpha \beta t - \beta t) \frac{k}{k - 1} = 0.
\]

As a direct consequence of Eq. 4.4 if we let some water into a \(k\)-cylinder hydraulic
and \(q\) number of cylinders are full in the vessel of player \(i\) then approximately
The nucleolus with characterization sets

$q$ number of cylinders are full in the vessel of player $j$. Thus, whenever $k$ is big $x(i) = r \Rightarrow x(j) \approx \alpha r$. In other words for big $k$ values the allocation generated by the hydraulic will be close to the proportional distribution. Let us demonstrate how the payoff vector converges for a 3-player game, where $E = 300$ and $c = (120, 240, 360)$.

<table>
<thead>
<tr>
<th>$E=300$</th>
<th>$d_1 = 120$</th>
<th>$d_2 = 240$</th>
<th>$d_3 = 360$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$x(1)$</td>
<td>$x(2)$</td>
<td>$x(3)$</td>
</tr>
<tr>
<td>$k=2$</td>
<td>60</td>
<td>120</td>
<td>120</td>
</tr>
<tr>
<td>$k=3$</td>
<td>40</td>
<td>90</td>
<td>170</td>
</tr>
<tr>
<td>$k=4$</td>
<td>60</td>
<td>100</td>
<td>140</td>
</tr>
<tr>
<td>$k=5$</td>
<td>48</td>
<td>96</td>
<td>156</td>
</tr>
<tr>
<td>$k=6$</td>
<td>52</td>
<td>100</td>
<td>148</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Proportional</td>
<td>50</td>
<td>100</td>
<td>150</td>
</tr>
</tbody>
</table>

Which brings us to our final note.

Observation 5. The nucleolus and the proportional rule are the two extremes of a family of solutions that correspond to hydraulics where the vessels have $k$ cylinders of equal size.

4.5 The nucleolus with characterization sets

With the help of the hydraulic framework we were not just able to prove Aumann and Maschler’s result but we found an efficient way to compute the nucleolus. To show the power of characterization sets we present here another way to derive an efficient algorithm. Equation 4.2 hints that only the singleton coalitions and the $n - 1$ person coalitions are relevant in the computation of the nucleolus. Let us investigate whether any of the characterization sets coincide with this collection of coalitions.

Lemma 4.8. Let $(d, E)$ be a bankruptcy problem and $\Gamma = (N, v_{(d,E)})$ be the corresponding bankruptcy game. Then $\mathcal{E}(\Gamma)$ consist of the singleton coalitions and coalitions with non-zero characteristic function value.

Proof. Let $S \subset N$ be such that $|S| > 1$ and $v_{(d,E)}(S) = 0$. It follows from monotonicity of $v_{(d,E)}$ that $v_{(d,E)}(T) = 0$ for any $T \subset S$. Thus for any partition $S_1, \ldots, S_k$ of $S$,

$$v_{(d,E)}(S) \leq v_{(d,E)}(S_1) + \cdots + v_{(d,E)}(S_k).$$
That is, $S$ is inessential. Now we prove that coalitions with non-zero characteristic function value are essential. By contradiction suppose that $v_{(d,E)}(S) > 0$ and $S$ is inessential. Then there exists a partition $T_1, \ldots, T_{k+1}$ of $S$ such that $v_{(d,E)}(S) \leq \sum_{i=1}^{k+1} v_{(d,E)}(T_i)$. Some of the $v_{(d,E)}(T_i)$ values may be zeros. By uniting these coalitions the characteristic function may only increase. Thus we may assume that $v(T_i) > 0$ for $i = 1, \ldots, k$ and $v_{(d,E)}(T_{k+1}) = 0$ where we allow $T_{k+1}$ to be the empty set. Notice that $k \geq 2$ in this setting.

$$v_{(d,E)}(S) \leq \sum_{i=1}^{k+1} v_{(d,E)}(T_i) = \sum_{i=1}^{k} v_{(d,E)}(T_i),$$

$$E - d(N \setminus S) \leq \sum_{i=1}^{k} (E - d(N \setminus T_i)) = k(E - d(N)) + \sum_{i=1}^{k} d(T_i).$$

Then by subtracting $\sum_{i=1}^{k} d(T_i)$ from both sides and estimating the sum from below we get

$$E - d(N) \leq E - d(N \setminus T_{k+1}) = E - d(N \setminus S) - \sum_{i=1}^{k} d(T_i) \leq k(E - d(N)),
1 \geq k.$$ 

which contradicts that $k \geq 2$.

Lemma 4.9. Let $(d, E)$ be a bankruptcy problem and $\Gamma = (N, v_{(d,E)})$ be the corresponding bankruptcy game. Then $DE(\Gamma)$ consist of the $n - 1$ player coalitions and coalitions with characteristic function value of zero.

Proof. By default $DE(\Gamma)$ contains all the $n - 1$ player coalitions. Let $S \subset N$ be such that $|S| < n - 1$ and $v_{(d,E)}(S) > 0$. It follows from monotonicity of $v_{(d,E)}$ that $v_{(d,E)}(T) > 0$ for any $S \subset T$. Thus for any overlapping decomposition $T_1, \ldots, T_k$ of $S$, 

54
\[ E - d(N \setminus S) \leq E - d(N \setminus S), \]
\[ E - d(N \setminus S) \leq kE - d(N \setminus S) - (k - 1)E, \]
\[ E - d(N \setminus S) \leq E - d(N \setminus T_1) + \cdots + E - d(N \setminus T_k) - (k - 1)v_{(d,E)}(N), \]
\[ v_{(d,E)}(S) \leq v_{(d,E)}(T_1) + \cdots + v_{(d,E)}(T_k) - (k - 1)v_{(d,E)}(N), \]

where we used that \( d(N \setminus S) = d(N \setminus T_1) + \cdots + d(N \setminus T_k). \) Thus \( S \) is dually inessential. Now we prove that coalitions with characteristic function value of zero are dually essential. By contradiction suppose that \( v_{(d,E)}(S) = \max\{E - d(N \setminus S), 0\} = 0 \) and \( S \) is dually inessential. By perturbing the claims with a small positive number we can always achieve that no collection of claims sum up to the estate, therefore we may also suppose that \( E - d(N \setminus S) < 0 \). Then there exists an overlapping decomposition \( T_1, \ldots, T_k, T_{k+1}, \ldots, T_\ell \) of \( S \) such that

\[ v_{(d,E)}(S) \leq \sum_{i=1}^{\ell} v_{(d,E)}(T_i) - (\ell - 1)v_{(d,E)}(N). \]

Some of the \( v_{(d,E)}(T_i) \) values may be zeros. We may assume that \( v(T_i) > 0 \) for \( i = 1, \ldots, k \) and \( v_{(d,E)}(T_i) = 0 \) for \( i = k + 1, \ldots, \ell \) where we allow \( k = \ell \).

\[
0 = v_{(d,E)}(S) \leq \sum_{i=1}^{\ell} v_{(d,E)}(T_i) - (\ell - 1)v_{(d,E)}(N)
= \sum_{i=1}^{k} v_{(d,E)}(T_i) - (\ell - 1)v_{(d,E)}(N)
= k(E - d(N)) + \sum_{i=1}^{k} d(T_i) - (\ell - 1)E
\leq k(E - d(N)) + (k - 1)d(N) + d(S) - (\ell - 1)E
\leq kE - d(N) + d(S) - (k - 1)E = E - d(N \setminus S) < 0.
\]

which is clearly a contradiction. Note that we used that

\[ \sum_{i=1}^{k} d(T_i) \leq (k - 1)d(N) + d(S), \]
which follows from the fact that $T_1, \ldots, T_k, T_{k+1}, \ldots, T_\ell$ compose an overlapping decomposition of $S$.

It seems that neither of these two characterization sets coincide with the desired collection, both of them is somewhat larger. In fact we can easily construct a bankruptcy game where every coalition is dually essential or a game where every coalition is essential. What is more we can define a game with $n$ players where both the size of $\mathcal{E}(\Gamma)$ and $\mathcal{DE}(\Gamma)$ is $O(2^n)$. A natural idea is to examine the intersection of these two sets.

Observation 6. Let $(d, E)$ be a bankruptcy problem and $\Gamma = (N, v_{(d,E)})$ be the corresponding bankruptcy game. Then $\mathcal{E}(\Gamma) \cap \mathcal{DE}(\Gamma)$ contains the $n-1$ person coalitions with non-zero characteristic function value and singleton coalitions with characteristic function value of zero.

Although, Observation 6 gives us the collection of coalitions that we have looked for, we still need to prove that $\mathcal{E}(\Gamma) \cap \mathcal{DE}(\Gamma)$ is actually a characterization set itself. For this we will show that the grand coalition is vital.

**Theorem 4.10.** Let $(d, E)$ be a bankruptcy problem and $\Gamma = (N, v_{(d,E)})$ be the corresponding bankruptcy game. Then

$$\mathcal{E}(\Gamma) \cap \mathcal{DE}(\Gamma) = \{i \in N \mid v_{(d,E)}(i) = 0\} \cup \{N \setminus i \mid i \in N, \ v_{(d,E)}(N \setminus i) > 0\}$$

is a characterization set for $\mathcal{N}(\Gamma)$.

**Proof.** Due to Theorem 3.20 we only need to prove that the grand coalition is vital. Suppose by contradiction that the grand coalition is not vital, that is, there exists a collection $B \subset 2^N$ and positive balancing weights $\{\lambda_T > 0 \mid T \in B\}$ such that $\sum_{T \in B} \lambda_T e_T = e_N$ and

$$v_{(d,E)}(N) \leq \sum_{T \in B} \lambda_T v_{(d,E)}(T). \quad (4.5)$$

If $B_+ \subseteq B$ denotes those coalitions $T$ for which $v_{(d,E)}(T)$ is not zero, then Eq. (4.5) can be written as

$$v_{(d,E)}(N) \leq \sum_{T \in B_+} \lambda_T v_{(d,E)}(T) = \sum_{T \in B_+} \lambda_T (E - d(N \setminus T)).$$

56
4.5 The nucleolus with characterization sets

Note that \( \sum_{T \in B^+} \lambda_T > 1 \) otherwise \( v_{(d,E)}(N) = E > \sum_{T \in B^+} \lambda_T (E - d(N \setminus T)) \).

Then

\[
v_{(d,E)}(N) \leq \sum_{T \in B^+} \lambda_T (E - d(N)) + \sum_{T \in B^+} \lambda_T d(T) \leq \sum_{T \in B^+} \lambda_T (E - d(N)) + \sum_{T \in B} \lambda_T d(T).
\]

Using that \( \sum_{T \in B} \lambda_T d(T) = d(N) \) we obtain

\[
E - d(N) \leq \sum_{T \in B^+} \lambda_T (E - d(N)),
\]

\[
1 \geq \sum_{T \in B^+} \lambda_T,
\]

which contradicts that \( \sum_{T \in B^+} \lambda_T > 1 \).

In light of Theorem 4.10, Lemma 4.8 and 4.9 can be further simplified. Actually it is enough to prove that \( \mathcal{E}(\Gamma) \) is a subset of the singleton coalitions and coalitions with non-zero characteristic function value. Similarly it is enough to show that \( \mathcal{DE}(\Gamma) \) is a subset of the \( n - 1 \) player coalitions and coalitions with characteristic function value of zero. From these it follows that

\[
\mathcal{E}(\Gamma) \cap \mathcal{DE}(\Gamma) \subseteq \{i \in N \mid v_{(d,E)}(i) = 0\} \cup \{N \setminus i \mid i \in N, \; v_{(d,E)}(N \setminus i) > 0\}.
\]

Since the grand coalition is vital \( \mathcal{E}(\Gamma) \cap \mathcal{DE}(\Gamma) \) is a characterization set. Any enlargement of a characterization set is a characterization set by Corollary 3.7, thus

\[
\{i \in N \mid v_{(d,E)}(i) = 0\} \cup \{N \setminus i \mid i \in N, \; v_{(d,E)}(N \setminus i) > 0\}
\]
is a characterization set.

Instead of proving that the grand coalition is vital we could simply deduct the above result from Theorem 3.6. This method is also instructive since it sheds some light on the structure of satisfaction values, thus we present it as well.

Second proof of Theorem 4.10. We need to prove that when we exclude coalition $S$ because it is inessential (dually inessential), then there is a partition (overlapping decomposition) of $S$, containing coalitions that belong to $\mathcal{E}(\Gamma) \cap \mathcal{DE}(\Gamma)$ and have smaller satisfaction values than $S$. There are two cases. If $v_{(d,E)}(S) = 0$, then by monotonicity $v_{(d,E)}(i) = 0$ for all $i \in S$. Thus $sat(S, x) \geq sat(i, x)$ for all $i \in S$ and for any core allocation $x$. Furthermore, $i \in \mathcal{E}(\Gamma) \cap \mathcal{DE}(\Gamma)$ for all $i \in S$ and naturally $e_S$ can be expressed as a linear combination of $\{e_i : i \in S\}$. If $v_{(d,E)}(S) > 0$, then by monotonicity $v_{(d,E)}(N \setminus i) > 0$ for all $i \in N \setminus S$. The collection $\{N \setminus i : i \in N \setminus S\}$ is an overlapping decomposition of $S$, furthermore for any core allocation $x$ (cf. Lemma 4.9)

$$v_{(d,E)}(S) \leq \sum_{i \in N \setminus S} v_{(d,E)}(N \setminus i) - (|N \setminus S| - 1)v_{(d,E)}(N)$$

$$x(S) - (v_{(d,E)}(S)) \geq -\left(\sum_{i \in N \setminus S} v_{(d,E)}(N \setminus i)\right) + (|N \setminus S| - 1)x(N) + x(S)$$

$$sat(S, x) \geq \sum_{i \in S} sat(N \setminus i, x)$$

Thus $sat(S, x) \geq sat(N \setminus i, x)$ for all $i \in N \setminus S$. Again $N \setminus i \in \mathcal{E}(\Gamma) \cap \mathcal{DE}(\Gamma)$ for all $i \in N \setminus S$ and $e_S$ can be expressed as a linear combination of $\{e_{N \setminus i} : i \in N \setminus S\}$ and $e_N$. By Theorem 3.6 we conclude that $\mathcal{E}(\Gamma) \cap \mathcal{DE}(\Gamma)$ is a characterization set for the nucleolus of $\Gamma$. \qed
Chapter 5

Directed acyclic graph games

The chapter is largely based on (Sziklai, Fleiner, and Solymosi, 2014). Some results, like the road construction algorithm, are published here for the first time.

5.1 Cost sharing on rooted graphs

Network-based cost games are among the most studied classes of cost allocation games. Standard tree and minimum cost spanning tree games as well as various other tree enterprises were proposed to analyze the same economic situation: How to share the incurring costs among costumers who would like to gain access to some public service or public facility and who are located in a predetermined spatial structure. Such situations arise frequently in the real world: Farmers share the maintenance costs of an irrigation system, workers share the cost of carpooling, and basically the cost sharing of any infrastructural projects that connects the users to a common source (electricity, sewage system, etc.) belongs to this family of problems. Respectively the game theoretical literature is vast.

What is common in the above listed examples, that there is a center that every player would like to reach. This center can be a service provider (power plant), a simple destination (workplace) or a source of supplies (water reserve). It is natural to model the players together with the center as a network. More formally we have a rooted graph, where nodes represent players, edges represent connection possibilities between the nodes, and a non-negative connection cost is assigned to each edge. There is a special node, the so called root of the graph, which represents
the service provider. The aim of every player is to get connected to the root, which they can achieve by constructing the edges. This cost however is incurred only once no matter how many players use the same route.

The first such game was introduced by Littlechild and Owen (1973). In the airport game aviation companies have to share the cost of a landing strip. The cost of building or maintaining a runway essentially depends upon the largest type of aircraft to land there. In this case the underlying graph is a path and the source corresponds to the beginning of the landing strip.

Let us formally define the game. Again \( N = \{1, 2, \ldots, n\} \) denotes the set of players or – in this case – aviation companies. Each aviation company has a respective cost factor \( l_i \), that is proportional to the length of landing strip the particular company needs for its aircrafts. We may assume that \( l_1 \leq l_2 \leq \cdots \leq l_n \). The characteristic function associated with the cost allocation problem is defined as follows \( c(S) = \max_{i \in S} l_i \). Let \( d_i = l_i - l_{i-1} \), where \( l_0 = 0 \). Then \( d_i \) describes how much more player \( i \) has to pay compared to player \( i - 1 \) to extend the landing strip and make it suitable for its aircrafts. In this way we can represent the airport problem as a rooted graph (see Figure 5.1), where the \( d_i \) values are the edge weights.

![Figure 5.1: The landing strip and the corresponding costs.](image)

Airport games were followed with other graph related cost games, such as standard tree games and minimum cost spanning tree games. Although, these games came with a different economic 'story' due to the similarity of the underlying network structure, many methods that were developed for one of the games work for the other ones too. The painting algorithm, which we will discuss shortly, is one such method.

In the upcoming sections we will introduce a natural generalization of standard tree games where the underlying network is a directed acyclic graph (DAG). We will analyze the properties of the game and show various results related to its core. For a large class of directed acyclic graph games we will provide a sufficient condition for balancedness and we also present methods that allows us to compute the nucleolus for some of these games.
5.2 Literature overview

An algorithm for the nucleolus of airport games was discovered by Littlechild (1974), later Sönmez (1993) derived a formula for it. Brânzei, Iñarra, Tijs, and Zarzuelo (2006) provided an algorithm for the nucleolus of airport profit games. In this variant each company has an expected revenue and a coalition is only feasible if this value exceeds the cost of the coalition. Otherwise the characteristic function value of that coalition is set to zero. For further references we direct the reader to the survey of Thomson (2007).

The next class of rooted graphs that was analyzed from cost sharing perspective was the tree. Different models were considered depending on how many players resided in a node and whether the nodes carried cost or not. In general these cost sharing games are called tree enterprises. In its most basic form tree enterprises contain no public nodes (i.e. nodes which are unoccupied by players), there is a non-negative cost assigned to the edges and zero cost to the nodes. Megiddo (1978) computed the nucleolus for tree enterprises where there is a one-to-one correspondence between players and the node set. His result was further strengthened by Galil (1980) and Granot, Maschler, Owen, and Zhu (1996). Other weighted allocations for this family of games were considered by Bjørndal, Koster, and Tijs (2004).

In case of standard or fixed tree games public nodes are possible and nodes can accommodate more than one player, but each player is still assigned to one node and no cost is attached to the nodes. Irrigation games are one example of such games (Márkus, Pintér, and Radványi, 2011). The most influential paper in the subject is (Maschler, Potters, and Reijnierse, 2010). This paper introduced the so called painting algorithm that computed the nucleolus of standard tree games. The algorithm was not only fast but managed to give an economic interpretation of the calculations. The algorithm roughly goes like this (some details are omitted): Each player start to paint the road that leads from its residence to the root. Players paint with unit speed and skip over road segments that are already painted. When a path from a player’s residence to the root is fully painted that player stops the painting. At the end of the algorithm each player is assigned the time he or she spent with

\footnote{Note that some models allow the root to be occupied. However, this distinction is insignificant as any meaningful cost sharing rule allocates zero cost to the residents of the root.}

\footnote{Although, this article was published in 2010 in IJGT, a working paper version existed and circulated since 1995.}
The painting story initiated research like (Bjørndal, Koster, and Tijs, 2004; Tijs, Koster, Molina, and Sprumont, 2002) and (Bergantinos, Gomez-Rúa, Llorca, Pulido, and Sánchez-Soriano, 2014) and the road construction algorithm that we will present here for directed acyclic graph games also resembles to this algorithm.

It is worth to note that the theory behind all these algorithms originates from (Maschler, Peleg, and Shapley, 1979). In this seminal paper the lexicographic center of a game — which turns out to be the nucleolus — is reached by ‘pushing hyperplanes’ with unit speed (cf. Section 3.1). Among the above cited papers (Granot, Maschler, Owen, and Zhu, 1996) used the most general approach: There are not only public nodes, but besides the edges there is also a cost function defined on the node set of the graph.

Minimum cost spanning tree (MCST) games are far the most popular among the cost sharing games based on rooted graphs. In this model the common assumption is that there is exactly one player assigned to each node. Hence the meaning of ‘public nodes’ is different. A node is considered to be public if the other players can use it without the authorization of the residing player, in other words they can use it even when the player is not in the coalition. A MCST game is called monotonic if every node in the network is public. We abbreviate this class by mMCST. The name comes from the fact that the characteristic function of such games is monotonic, since if a player joins to a coalition the cost can only increase. For an example see (Granot and Maschler, 1998) or (van den Nouweland, Tijs, and Maschler, 1993). In case of non-monotonic MCST games the underlying graph is usually considered to be a complete graph. Coalitions can only use the nodes that correspond to the residence of its players and edges that span between such nodes, or connect to the root. Monotonic MCST games are much harder as in that case finding the characteristic function value of a coalition — which is the cost of the cheapest subnetwork that connects all the players of that coalition to the root — is equivalent to the Steiner Tree problem, therefore it is NP-hard.

MCST problems were introduced by Claus and Kleitman (1973), but Bird (1976) was the first to associate a cooperative game to such games. Bird proposed the following rule: Find a minimum cost spanning tree $T$ in the graph, and to each player allocate the cost of the first edge that player encounters on the unique path in $T$ that leads from its residence to the root. Although, the Bird-rule always produces a core stable allocation, it yields a favorable bias towards players who are
located relatively far from the root (Faigle, Kern, Hochstättler, and Fekete, 1997). Dutta and Kar (2004) recommended another solution which was axiomatically better supported. However, Trudeau (2012) pointed out that both the Bird-rule and the Dutta-Kar solution is based on Prim’s algorithm, thus a little change in the edge weights can seriously alter the proposed allocation. This is of course an undesirable property considering real life applications. There is an extensive literature devoted to fair allocation rules on MCST games. Without attempting to be comprehensive we refer the reader to (Bergantiños and Vidal-Puga, 2007) and (Bogomolnaia and Moulin, 2010) and the references therein.

Classical game theoretical solutions for MCST games were also analyzed by Granot and Huberman (1981) and Granot and Huberman (1984), although later Faigle, Kern, and Kuipers (1998) proved that finding the nucleolus of a MCST game is \( \text{NP}\)-hard. They also proved that determining whether an allocation is a core member or not is \( \text{NP}\)-complete Faigle, Kern, Hochstättler, and Fekete (1997). MCST games with multiple source are called minimum cost spanning forest games and were studied by Rosenthal (2013); Granot and Granot (1992); Kuipers (1997).

There are many applications connected to games on rooted graphs. Just to list a few, Young, Okada, and Hashimoto (1982) provided a case study of the cost allocation of a water supply project in Sweden, Aadland and Koplin (1998) analyzed the cost sharing arrangements on a sample of twenty-five irrigation ditches located in a south-central Montana community, Dong, Ni, and Wang (2012) proposed methods to distribute the cost of cleaning up a polluted river that crosses more than one countries. Leoneti, do Prado, and de Oliveira (2011) and Leoneti, de Oliveira, and Pires (2013) reports on a possible application in Brazil where water and sanitation sector has received greater government attention recently and there is a significant amount of resources to be invested.

Let us also mention that there are many (not necessarily rooted) graph based cost games that are similar in their concept (e.g. shortest path games, peer group games, highway games), without discussing them in depth we refer the reader to (Rosenthal, 2013; Brânzei, Fragnelli, and Tijs, 2002; Çiftçi, Borm, and Hamers, 2010).

Directed acyclic graph (DAG) networks are a generalization of standard tree games. This structure has not been previously analyzed from a cooperative game
theoretic perspective. In the next section we introduce and discuss the basic properties of this game. Our aim is to describe the core of the game, then to establish an efficient algorithm for the nucleolus of large classes of DAG-games.

In order to give more insight into our model let us compare airport games, standard tree games, mM CST games, and DAG-network games. These games have the same setup, namely they are based on a rooted graph, where players, who are located on the nodes, would like to share the construction cost of the edges. Table 5.1 summarizes the differences of these games, while Figure 5.2 shows how they are related to each other.

<table>
<thead>
<tr>
<th>Game</th>
<th>Graph</th>
<th>Edges</th>
<th>Residents/node</th>
<th>Convexity</th>
<th>Core</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airport</td>
<td>path</td>
<td>(un)directed</td>
<td>1 – n</td>
<td>concave</td>
<td>non-empty</td>
</tr>
<tr>
<td>Standard Tree</td>
<td>tree</td>
<td>(un)directed</td>
<td>0 – n</td>
<td>concave</td>
<td>non-empty</td>
</tr>
<tr>
<td>mM CST</td>
<td>connected</td>
<td>undirected</td>
<td>1</td>
<td>not concave</td>
<td>non-empty</td>
</tr>
<tr>
<td>DAG</td>
<td>connected DAG</td>
<td>directed</td>
<td>0 – n</td>
<td>not concave</td>
<td>can be empty</td>
</tr>
</tbody>
</table>

Table 5.1: Comparison of graph related cost games

Notice that in case of airport games and standard tree games the edges can be considered both directed or undirected. In case of airport games and mM CST games each node is occupied, while in standard tree and DAG-games there can be nodes where no player resides.

![Venn-diagram of graph related cost games](image)

**Figure 5.2**: Venn-diagram of graph related cost games

### 5.3 Formal definition of directed acyclic graph games

A directed acyclic graph network $\mathcal{D}$ or shortly a $DAG$-network is given by the following:

- $G(V, A)$ is a directed acyclic graph, with a special node - the so called root of $G$, denoted by $r$ - such that from each other node of $G$ there leads at least one directed path to the root. $G$ is considered to be a simple graph, i.e. it has no loops or parallel arcs.
• There is a cost function $\delta : A \to \mathbb{R}_0^+$ that assigns a non-negative real number to each arc. This value is regarded as the construction cost of the arc.

For a subgraph $T$, $V(T)$ denotes the node set of $T$. Similarly $A(T)$ denotes its arc set, while $A_p$ is used for the set of arcs that leave node $p$. We call nodes that have one leaving arc passages, while nodes that have more than one leaving arcs are called junctions. Junctions that have more than one leaving zero cost arcs (or simply zero arcs) are called gates (see Figure 5.3).

![Figure 5.3: A typical passage, junction and gate. Note that gates are special junctions.](image)

Let $N$ be a set of players and let $\mathcal{R} : N \to V$ be the residency function that maps $N$ to the node set of $G$. If player $i$ is assigned to node $p$ we say that player $i$ resides at $p$. A node is occupied if at least one player resides in it. Note that unoccupied leaves are redundant and can be omitted from the network. The residency function is not assumed to be injective and/or surjective, but it is a proper function. It means that any one player resides at exactly one node, but there can be unoccupied nodes or nodes having more than one residents. The set of residents of a subgraph $T$ is denoted by $N(T)$, formally, $N(T) = \mathcal{R}^{-1}(V(T))$. A network $D$ together with a residency mapping $\mathcal{R}$ defined on $D$ is called a player network.

For a subgraph $T$, we define its construction cost $C(T)$ as the total cost of the arcs in $T$, i.e. $C(T) = \sum_{a \in A(T)} \delta(a)$. A path whose end point is the root is called a rooted path. A connected subgraph of $G$ that is a union of rooted paths is called a trunk. For each coalition $S$, let $T_S$ denote the set of trunks that have maximum number of arcs among the cheapest trunks that connect all players in $S$ to the root. The maximality requirement may seem odd at first, but it is needed to ensure the uniqueness of $T_N$. We will provide an example in Section 5.5.

We say that a trunk $T$ corresponds to a node set $B$ if $V(T) = B$. Similarly we say that a coalition $S$ corresponds to the trunk $T$ if $T \in T_S$. Note that more than one coalition can correspond to the same trunk.
The characteristic function of the cost allocation game that is associated with the player network \((D, R)\) is defined as follows.

\[
    c_{(D, R)}(S) \overset{\text{def}}{=} C(T) \quad T \in T_S.
\]

The pair \((N, c_{(D, R)})\) is called a DAG-game. The definition of \(c_{(D, R)}\) is motivated by the fact that by leaving the grand coalition the players in \(S\) need not pay more than \(c_{(D, R)}(S)\) to get connected to the root. As any trunk in \(T_S\) has the same construction cost, \(c_{(D, R)}(S)\) is well-defined. Finding these trunks, however, may be computationally demanding.

It is straightforward to see that the characteristic function of any DAG-game is non-negative, monotone and subadditive (even strongly subadditive, i.e. \(c(S) + c(T) \geq c(S \cup T)\) holds for any not necessarily disjoint coalitions \(S\) and \(T\)). On the other hand, Figure 5.4A shows an example when a stronger property, submodularity is not satisfied.

Let \(S_1 = \{1, 3\}\) and \(S_2 = \{2, 3\}\), then

\[
    3 + 2 = c(S_1) + c(S_2) < c(S_1 \cup S_2) + c(S_1 \cap S_2) = 4 + 2,
\]

thus we conclude that DAG-games need not be concave.

The following example demonstrates that DAG-games need not even be balanced. Consider the player network \((D, R)\) depicted in Figure 5.4B. The cost of connecting any two-player coalition is 3, however \(c_{(D, R)}(N) = 5\) which leaves the core empty.

![Example A](image1)

![Example B](image2)

**Figure 5.4:** The first example shows that the characteristic function need not be submodular. Example B displays a DAG-network that induces a cost game with an empty core. The residents of the nodes are given in braces in both cases.
Later we will show that the condition

\[(*) \text{ there must be a resident at each node with more than one entering arc and with leaving arc(s) all of positive cost}\]

is sufficient for a DAG-game to have a non-empty core. Although, this property seems unrestrictive it is crucial in the applicability of our results. Notice that property \((*)\) can be checked efficiently. In the following we will assume that \((*)\) holds for any \((D, R)\) network.

Finally, we note that in general it is computationally hard to calculate the characteristic function value of a given coalition. Finding an element of \(T_S\) for an arbitrary \(S \subseteq N\) is equivalent to the acyclic directed Steiner tree\(^3\) problem, which is \(NP\)-hard (Winter, 1987).

5.4 Significance of DAG-games

Before we prove our results related to DAG-games, let us elaborate on the practical and theoretical significance of this game class. A typical economic situation that can be modeled in this way is the cost allocation of infrastructural projects. Consider for instance a group of towns that would like to connect themselves to a water reserve. Clearly not every town has to build a direct pipeline to the source. A possible solution is to connect the nearest towns with each other and then one of the towns with the reserve. The towns that are already connected to the water system can force the rest to pay some of their construction cost, otherwise they can close down the outgoing water flow. On the other hand, no town can be forced to pay more than the cost of directly connecting itself to the water reserve.

This example can be retold in a minimum cost spanning tree setting as well since the water can flow both ways in the pipe. However, if the reserve is on a hill or mountain (which is quite possible) and the towns that have to be serviced are at different elevations then the DAG-setting can be more convincing. Especially if we assume that pumping the water uphill is too costly for the towns.

Even when the links are undirected some mM CST games can be converted into DAGs by choosing directions for the links appropriately. Figure 5.5 depicts a

\(^3\) Also known as the Steiner arborescence problem.
mMCST and a DAG-network that induce the same characteristic cost function. However, as the next example shows – not every mMCST-game can be converted into a DAG in such a straightforward fashion.

**Example 5.1.** Consider the 3-player mMCST-game depicted in Figure 5.6. The characteristic function is given by: \( c(\{1\}) = 5, c(\{2\}) = 10, c(\{3\}) = 9, c(\{1, 2\}) = 10, c(\{1, 3\}) = 14, c(\{2, 3\}) = 14 \) and \( c(\{1, 2, 3\}) = 15 \). Notice that coalition \( \{2, 3\} \) uses the \( b - c \) edge in different direction than the grand coalition.

Now we provide a construction that works for any monotonic, subadditive game. Let \( N \) be a player set and \( \hat{c} : 2^N \to \mathbb{R} \) be a monotonic, subadditive cost function. We assign a node \( p_S \) for each coalition \( S \subseteq N \) and introduce a new node \( r \), which will be the root of the graph. Each coalition has an arc \( a_S \) that leaves \( p_S \) and enters the root, such that \( \delta(a_S) = \hat{c}(S) \). Furthermore, for each \( i \in N \), there is a zero arc, denoted by \( a^i_S \) that leaves \( p_{\{i\}} \) and enters \( p_S \cup \{i\} \) for all \( \emptyset \neq S \subseteq N \setminus \{i\} \). Only the singleton coalitions are occupied, i.e. \( R(i) = p_{\{i\}} \). We call the obtained player-network \( (D, R) \) as the characteristic DAG-representation of \( \hat{c} \).

**Theorem 5.1.** Let \( N \) be a player set and \( \hat{c} : 2^N \to \mathbb{R} \) a monotonic, subadditive cost function. There exists a DAG-network \( D = (G(V,A), \delta) \) and a residency
5.4 Significance of DAG-games

mapping $\mathcal{R} : N \rightarrow V$ such that

$$c_{(D,\mathcal{R})}(S) = \hat{c}(S) \quad \forall S \subseteq N.$$ 

**Proof.** We will use the characteristic DAG-representation of $\hat{c}$. We have to show that the cost of the cheapest trunk of any coalition $S$ equals to $\hat{c}(S)$. This is trivially true for the singleton coalitions. Each player – as a singleton – will use its direct connection to the root as any other route could be only more expensive due to the monotonicity of $\hat{c}$. Now let $S$ be an arbitrary non-singleton coalition. We may suppose that $\delta(a_S) > 0$, otherwise both $\hat{c}(S)$ and $c_{(D,\mathcal{R})}(S)$ are trivially zero. We will prove that there exists a cheapest trunk $T' \in T_S$ such that the arc set of $T'$ consists of $a_S$ and the zero arcs $\{a^i_S | i \in S\}$. Let $T \in T_S$ and suppose that $A(T) = \{a_{S_1}, \ldots, a_{S_k}\}$. If $S = S_\ell$ for some $\ell \in \{1, \ldots, k\}$, then by deleting the $a_{S_j}$, $j \neq \ell$ arcs and the zero arcs that enter $p_{S_j}$, $j \neq \ell$ we can obtain $T'$. This would imply $T' \in T_S$ as the cost could only decrease by deleting these arcs. If $S \subset S_\ell$ for some $\ell \in \{1, \ldots, k\}$, then again we can obtain a weakly cheaper trunk by connecting the members of $S$ via $a_S$ and deleting all the other $a_{S_\ell}$ arcs.

Thus $S_\ell \subset S$ for all $i \in \{1, \ldots, k\}$. Although, a player may be connected to more than one $p_{S_\ell}$ node, in real he only needs one of the $a_{S_\ell}$ arcs to reach the root. Let us assign the players of $S$ to one of the $a_{S_\ell}$ arcs. Let us denote by $S^\ell \subset S$ those players of $S$ that were assigned to $S_\ell$. Note that the $\{S^\ell | \ell \in \{1, \ldots, k\}, S^\ell \neq \emptyset\}$ coalitions comprise a partition of $S$. If $S^\ell = \emptyset$ for some $\ell \in \{1, \ldots, k\}$ then $a_{S_\ell}$ and the entering zero arcs of $p_{S_\ell}$ can be deleted. If $\emptyset \neq S^\ell \subset S_\ell$ then the players of $S^\ell$ can be reassigned to $a_{S_\ell}$, that is, we can delete $a_{S_\ell}$ and the entering zero arcs of $p_{S_\ell}$ and construct $a_{S^\ell}$ and the the entering zero arcs of $p_{S^\ell}$ instead. Due to the monotonicity, the cost can only decrease this way, while all the players of $S$ are still able to reach the root. Let us perform this transformation for all the $S_\ell$ coalitions. We obtain trunk $\hat{T} \in T_S$ such that

$$A(\hat{T}) = \{a_{S^\ell} | \ell \in \{1, \ldots, k\}, S^\ell \neq \emptyset\} \cup \{a^i_{S^\ell} | \ell \in \{1, \ldots, k\}, S^\ell \neq \emptyset, i \in S\}.$$ 

By the subadditivity of $\hat{c}$
\[ \hat{c}(S) \leq \sum_{\ell \in \{1, \ldots, k\}, S^\ell \neq \emptyset} \hat{c}(S^\ell) \]
\[ \delta(a_S) \leq \sum_{\ell \in \{1, \ldots, k\}, S^\ell \neq \emptyset} \delta(a_{S^\ell}) \]

Thus \( T' \) is as least as cheap as \( \hat{T} \), from which \( T' \in T_S \) follows.

Figure 5.7 shows an example of a characteristic DAG-representation.

![Diagram of characteristic DAG-representation](image.png)

**Figure 5.7:** The characteristic DAG-representation of the mMCST-game of Example 5.1. In order to retain transparency the leaving zero arcs of \( p_{(2)} \) and \( p_{(3)} \) were shortened.

The characteristic DAG-representation has mostly theoretical value. In such networks there are many unoccupied passages with more than one entering arcs, hence the (*) property is not satisfied. The other drawback is that it can inflate a game with \( n \) players, which has a nice DAG-representation of \( O(n) \) nodes and arcs, into a monster network of \( O(2^n) \) nodes and arcs (e.g. the representation of an airport or a standard tree game).

As mMCST games are monotonic and subadditive, there indeed exists a DAG-representation for any of them\(^4\). In addition multiple source standard tree games can also be converted into DAG instances by contracting the sources into a single node. Such conversions can have various benefits. In DAG-games the players are hierarchically structured which in some cases allows us to compute the nucleolus of the game in polynomial time or describe the core efficiently. In comparison

\(^4\)It can happen that the only DAG-representation of an mMCST-game is the characteristic representation where the (*) property is not satisfied. This does not contradict the fact that every mMCST game is balanced, since (*) is not a necessary condition for the non-emptiness of the core.
finding the nucleolus of an MCST game is $NP$-hard in general (Faigle, Kern, and Kuipers, 1998).

5.5 The canonization process and its consequences

In this section we will introduce a network canonization process which has two main advantages. Firstly, it makes the structure plainer, allows us to define notions like 'principal ancestor' (cf. Definition 5.4) in a simpler way. Secondly, it makes the trunk of the grand coalition ($T_N$) unique. Again this will tremendously help to understand and work with DAG-networks.

We say that a player network $(\mathcal{D}, \mathcal{R})$ is in canonical form if the following properties are fulfilled:

**P1** Each junction has a leaving zero arc: $p \in V, |A_p| > 1 \Rightarrow \exists a \in A_p \text{ s.t. } \delta(a) = 0$.

**P2** For each passage the cost of the leaving arc is positive: $p \in V, A_p = \{a\} \Rightarrow \delta(a) > 0$.

**P3** There resides a player in each passage: $p \in V, A_p = \{a\} \Rightarrow \exists i \in N \text{ s.t. } R(i) = p$.

To transform a player-network into a form where property **P1** is fulfilled we have to perform the following procedure for each node $p \in V$ such that $|A_p| \geq 2$ and $\min_{a \in A_p} \delta(a) = \alpha_p > 0$.

1. Introduce an unoccupied new node $p'$ with the same set of leaving arcs as $p$ has, but reduce the cost of the arcs by $\alpha_p$.
2. Erase all the arcs that leave $p$.
3. Finally, introduce a new arc from $p$ to $p'$ with cost $\alpha_p$.

Property **P2** can be achieved by contracting each passage that has a leaving zero arc with the endnode of that arc, by uniting the resident sets of the contracted nodes, and by eliminating that zero arc. Obtaining both **P1** and **P2** require
equivalent transformations in the sense that the construction cost of the trunks in $T_S$ is unchanged for any coalition $S$.

Finally, if $p$ is an unoccupied passage and $p$ has only one entering arc then it can be omitted from the network. The entering and leaving arc of $p$ can be replaced by a single arc with the aggregated construction cost. Needless to say that this procedure does not change the costs of the $T_S$ trunks either. Note that if a passage has more than one entering arc then by property (*) it is occupied.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{network.png}
\caption{A DAG-network with player set $N = \{1, 2, 3, 4, 5, 6\}$ before and after canonization. Notice that after canonization $g'$ has a shortcut to the root.}
\end{figure}

**Example 5.2.** Let us demonstrate the canonization process on an example. Consider the player-network depicted in Figure 5.8. There are two junctions $g$ and $i$. The former does not have a leaving zero cost arc. Hence we introduce a new node $g'$, connect $g'$ with the neighbours of $g$, delete the leaving arcs of $g$ and finally connect $g$ and $g'$. The new arcs are given appropriate weights. As $h$ is a passage with a zero cost arc, we contract it with node $g$. The player sets of the two nodes are united. Node $b$ is an unoccupied passage, thus we delete this node. We connect $d$ with $a$ directly. The new arc has the aggregated cost of the deleted entering and leaving arcs of $b$. This concludes the three steps of canonization.

Our first observation summarizes the above findings.

**Observation 7.**

- All networks that satisfy (*) can be canonized.
5.5 The canonization process and its consequences

- The characteristic function is unaffected by the canonization process.

Although, canonization ensures that $T_N$ contains only a single element, this cannot be said in general about the trunks of other coalitions. In the following we will assume that $T_S$ contains only a single trunk for any coalition $S$. This can always be achieved by perturbing the positive arc costs. The most important solution concepts, such as the core, the nucleolus and the Shapley-value are all continuous functions of the characteristic function, which in turn is a continuous function of the arc costs. Thus, such perturbation does not affect the outcome of the game (Figure 5.9 demonstrates the process). Henceforward we will refer to $T_S$ as this unique trunk that has maximum number of arcs among the cheapest trunks that connect all members of $S$ to the root.

**Figure 5.9:** There are two equally cheap ways to connect coalition $\{3\}$ to the root. By perturbing the positive arc costs with a small $\varepsilon > 0$ the trunk of coalition $\{3\}$ becomes unique (the red path in the network on the right).

From now on we will drop the residency function $\mathcal{R}$ from the notation and simply write $c_D$. We will denote by $\Gamma_D$ the cost game induced by $c_D$, i.e. $\Gamma_D = (N, c_D)$. For sake of simplicity we will also call a DAG-game on a canonized player network as a *canonized DAG-game*. Let us now see some consequences of canonization. We also need to introduce further notions and notation.

For each node $p$, the cheapest arcs in $A_p$ are called $T_N$-arcs. The name comes from the fact that an arc is a $T_N$-arc if and only if it is an element of $A(T_N)$. If $a, a' \in A_p$, $a$ is a $T_N$-arc and $\delta(a') > \delta(a)$, then $a'$ is called a shortcut. Thus every arc that is not a $T_N$-arc is a shortcut. If two nodes $p$ and $q$ are connected with a shortcut then either it is the cheapest route between these two nodes (hence the name) or it is so costly that no coalition uses it at all. In this latter case the arc can be omitted from the network. If $a, a' \in A_p$ are $T_N$-arcs then the construction cost of both $a$ and $a'$ is zero (this is a consequence of $P1$).
The subgraph associated to the grand coalition \((T_N)\) holds special importance. First this is the graph that will be constructed in the end. All the other arcs are only good for improving the bargaining positions of certain players\(^5\). Note that \(T_N\) is not necessarily a tree as it may contain some additional zero arcs. Unlike other trunks, \(T_N\) can be constructed efficiently in linear time. The connection cost of any occupied node is at least as much as the cost of the cheapest arc that leaves that node. Furthermore, every unoccupied node has a leaving zero arc, therefore connecting an unoccupied node does not impose extra cost. Thus including the cheapest arcs from every node connects all nodes to the root. It follows that 
\(V(T_N) = V\) and \(A(T_N)\) contains every arc that is not a shortcut.

Another feature of \(T_N\) is that it induces a partial order \(\prec\) on the nodes. We say that \(p\) is an ancestor of \(q \neq p\) if \(p\) can be reached from \(q\) via a path in \(T_N\), we denote this by \(p \prec q\). In such cases we also say that \(q\) is an descendant of \(p\). Node \(p\) is a direct ancestor or parent of \(q\) if \(p\) is an ancestor of \(q\) and they are connected with a \(T_N\)-arc. This relation is denoted by \(\pi(q)\) whenever the direct ancestor is unique (gates have more than one parent). If \(p\) is a parent of \(q\) then \(q\) is referred as a direct descendant or child of \(p\). The node set that contains \(p\) together with its descendants is called a full branch and denoted by \(B_p\).

Sometimes we are interested only in some of the descendants of \(p\), therefore we cut off some segments of \(B_p\). Removing a node from \(B_p\) other nodes can become unreachable too. A specific branch, denoted by \(B^Q_p\) is a subset of \(B_p\) that collects nodes that still can reach \(p\) using only \(T_N\)-arcs after removing the node set \(Q\) from \(B_p\). Formally

\[
B^Q_p \overset{def}{=} \{ q \in B_p \mid \exists P_{q-p} \text{ such that } V(P_{q-p}) \subseteq B_p \setminus Q \},
\]

where \(P_{q-p}\) denotes a path in \(T_N\) that leads from \(q\) to \(p\). In other words a branch is the node set of a union of paths in \(T_N\) which have a common origin\(^6\). To emphasize its origin a \(B^Q_p\) branch is also called a \(p\)-branch. Note that if \(B^Q_p = B^Q_p\) then \(B^Q_p \cap Q\)

---

\(^5\)Since we analyze the game from a cooperative perspective, we do not formalize the bargaining processes that take place between the players. However, such processes are certainly present.

Consider the case of Example 5.2. Without the shortcut of \(g'\) player 6 may be forced to pay as much as 10 in the core, while with the shortcut present his core payoff is bounded from above by 7.

\(^6\)Since the word 'root' is taken, we use 'origin' instead. Do not be alarmed by the fact that the 'origin' is in real the common end node of the union of paths. By reversing the direction of the graph the word makes more sense.
define the same node set as well. We say that the \( B_p^Q \) branch is in \textit{standard form} if the cardinality of \( Q \) is minimal, in other words if there exists no \( Q' \) such that \( B_p^Q = B_p^{Q'} \) and \(|Q'| < |Q|\).

We say that the node set \( B \) is \textit{proper} if deleting \( B \) from \( G \) along with all of its entering and leaving arcs the root can still be reached on a directed path from any of the remaining nodes (i.e. the remaining graph is a trunk).

Let us illustrate the above introduced notions with some examples. Consider again the canonized DAG-network \( D_c \) depicted in Figure 5.2. The only shortcut in \( D_c \) is the one that connects node \( g' \) with the root. All the other arcs are \( T_N \)-arcs. Nodes \( a \) and \( d \) form a branch in \( D_c \), but this branch is not proper as without \( a \), node \( c \) gets disconnected. On the other hand node \( f \) composes a proper branch in itself. Finally, the node set that corresponds to the trunk \( T_{\{3,6,8\}} \) is \( V \backslash (B^1_c \cup B^g_c) = \{a, d, g', g, i\} \) and \( c_D(\{3,6,8\}) = 12 \).

This last example raises a question. Can we obtain every trunk by removing some branches from the graph? By Lemma 5.2 the answer is positive.

**Lemma 5.2.** The node set of every trunk that corresponds to a coalition \( S \subset N \) can be obtained by deleting some branches from \( V \). The removal branches can be chosen in such way that each of them originates from a passage. Formally for any \( S \subset N \) there exists \( Q_1, \ldots, Q_k \subset V \) and \( p_1, \ldots, p_k \in V \) such that

\[
V(T_S) = V \backslash \bigcup_{j=1}^k B_{p_j}^{Q_j},
\]

where \( p_j \) is a passage for all \( j \in \{1, 2, \ldots, k\} \).

**Proof.** Any trunk \( T \) has a representation where \( V(T) \) is obtained by removing branches from \( V \). This is trivial as any single node is a branch in itself if we trim all its children. The only thing we need to prove is that these branches can be picked in such way that each of them originates from a passage. Let \( \{p_1, \ldots, p_k\} \subset V \backslash V(T_S) \) denote those passages that connect to \( V(T_S) \) from the outside, i.e. for which \( \pi(p_j) \in V(T_S) \) for all \( j = 1, \ldots, k \). Due to the definition of \( T_S \) there exists at least one such passage. Let us remind the reader that \( T_S \) is the trunk that has maximum number of arcs among the cheapest subgraphs that connect \( S \) to the root. Therefore any junction that connects to a such trunk with a zero arc by definition is included in \( T_S \) even if no player of \( S \) resides there. If we remove all the \( B_{p_1}, \ldots, B_{p_k} \) branches from \( V \) it can happen that
we removed some nodes in $V(T_S)$ as well, i.e. $V \setminus (\bigcup_{j=1}^k B_{p_j}) \subset V(T_S)$. In order to retain all the nodes of $V(T_S)$ we trim the $B_{p_j}$ branches where they intersect with $V(T_S)$. Let $Q_j = V(T_S) \cap B_{p_j}$ then $B_{p_j}^{Q_j}$ is a proper branch for any $j$ and $V(T_S) = V \setminus (\bigcup_{j=1}^k B_{p_j}^{Q_j})$.

The obtained $V \setminus \bigcup_{j=1}^k B_{p_j}^{Q_j}$ expression is called a standard representation of $V(T_S)$, if the redundant nodes have been removed from the $Q_j$ sets, i.e. each $B_{p_j}^{Q_j}$ branch is in standard form. A trunk can have more than one standard representation. Also notice that the $B_{p_j}^{Q_j}$ branches may not be disjoint. For instance in Example 5.2 the trunk of coalition $\{1, 2, 3, 5\}$ corresponds to $V \setminus (B_e \cup B_g)$.

5.6 The core of the canonized DAG-game

In this section we show that canonized DAG-games are balanced (i.e. they have a non-empty core). We identify nodes that behave like the source. Their residents do not invest in the construction of the network. Moreover if a set of players reach such a secondary source – which we will call a free node – they do not participate in the construction anymore. That is, the residents of a branch that originates from a free node need not pay more in the core than the construction cost of the branch itself. Uncovering the free nodes have many advantages. One of them is – as we will see in the next section – that the free nodes can be contracted with the root. Although, this transformation changes the characteristic function of the game, it does not affect the core or the nucleolus.

The following extension of the cost function will be needed. We define $\tau(Q, S)$ as the cost of the arcs in $T_S$ that leave from node set $Q$, i.e.

$$\tau(Q, S) \overset{def}{=} \sum_{a \in (\bigcup_{q \in Q} A_q) \cap A(T_S)} \delta(a).$$

We define the standard allocation\footnote{The standard allocation is just a weighted version of the Bird-rule which was proposed for MCST games (Bird, 1976).} $\hat{x}$ of $\Gamma_D$ as follows. For each player $i \in N$ let $\hat{x}(i) = \frac{\delta(a_p)}{|N(p)|}$ where $i \in N(p)$ and $a_p$ is one of the leaving $T_N$-arcs of $p$. For instance in the canonized network of Example 5.2 $\hat{x}(6) = \hat{x}(7) = 2$ and $\hat{x}(8) = 0$.

In our first lemma we show that the standard allocation is a core selector.
Lemma 5.3. $C(\Gamma_D) \neq \emptyset$ for any DAG-network $D$ in canonical form.

Proof. We will use the standard allocation. The residents of a junction do not have to pay according to the standard allocation, while the residents of a passage pay for the construction cost of the $T_N$-arc that leaves the passage. Since unoccupied nodes can only be junctions, it follows that for any node set $B$,

$$\hat{x}(N(B)) = \tau(B, N).$$ (5.1)

That is, the players of $B$ pay for all the arcs that leave $B$ in $T_N$. This means, in particular, that $\hat{x}(N) = \tau(V, N) = c_D(N)$. On the other hand, for any $S \subseteq N$

$$\hat{x}(S) = \sum_{p \in R(S)} |S \cap N(p)| \cdot \frac{\delta(a_p)}{|N(p)|} \leq \sum_{p \in R(S)} \delta(a_p) \leq \sum_{p \in V(T_S)} \delta(a_p) \leq C(T_S) = c_D(S),$$

where $R(S) = \{ R(i) : i \in S \}$. The last inequality holds, because $\sum_{p \in V(T_S)} \delta(a_p)$ collects the cost of the cheapest arcs of each node in $T_S$, but $A(T_S)$ may contain shortcuts as well.

The following definitions will be useful. We say that node $q$ is a key ancestor of node $p$, if there are two paths in $T_N$ from $p$ to $q$ such that these paths are arc-disjoint except maybe for some zero arcs (semi-arc-disjoint from now on). The degenerate cases when these two paths partially or (completely) coincide are also included in this definition. Thus if there exists a zero cost path from $p$ to $q$ then $q$ is a key ancestor of $p$. Clearly, each junction has at least one key ancestor. On the other hand, by property $P2$, a passage could not have a key ancestor, so we define the only key ancestor of a passage to be itself. For similar reasons we define the root to be the key ancestor of itself.

Definition 5.4. The principal ancestor of node $p$ is a unique node $q \in V$, denoted by $\Pi(p)$ that is a key ancestor of $p$ and $q \prec q'$ for every other key ancestor $q'$ of $p$ (i.e. the key ancestor closest to the root).

Notice that a junction cannot be a principal ancestor of any of its descendants. The only principal ancestor that is not a passage is the root. In Section 5.10, we
provide an efficient algorithm that finds the principal ancestor of each node in a DAG-network.

**Definition 5.5.** We say that an occupied node $p$ is free if $x(N(p)) = 0$ for any core element $x$, i.e. the residents of $p$ do not have to pay to get connected to the root. An unoccupied node $p$ is called free if $\Pi(p) = r$. The set of free nodes is denoted by $F$.

Note that if $p$ is a passage then the standard allocation would assign positive value to $N(p)$. In other words every free node is a junction.

**Example 5.3.** Consider the DAG-network depicted in Figure 5.10. The nodes $a, b, d, e, f$ and $h$ are passages. Thus the key and principal ancestors of these nodes are themselves. Node $c$ has two (semi-)arc-disjoint paths in $T_N$ that enter the root, hence the principal ancestor of $c$ is the root. Although, there exists two paths from $g$ to $a$, the latter node is not a key ancestor of $g$, since one of the paths does not belong to $T_N$. The key ancestors of $g$ are $g$ and $e$. Among these two node $e$ is closer to the root, hence $e$ is the principal ancestor of $g$. Finally, the principal ancestor of $i$ is $d$. Note, that if $d$ was a junction, the principal ancestor of $i$ would be the root.

![Figure 5.10: A canonized network with nine players. Note that aside for the root node $c$ is free as well.](image)

In our next theorem we will characterize the set of free nodes. Before we proceed let us state a simple lemma that will play a crucial role in the proof.

**Lemma 5.6.** Let $B^Q_p$ be any branch originating from node $p$. If $\text{sat}(N(V \setminus B_p), y) = 0$ for every core allocation $y$, then $y(N(B^Q_p)) \leq \tau(B^Q_p, N)$. In other words the residents of $B^Q_p$ do not pay more than the costs of their $T_N$-arcs.
5.6 The core of the canonized DAG-game

Proof. We proceed by contradiction. Suppose \( y(N(B_p^Q)) > \tau(B_p^Q, N) \) for some \( y \in C(\Gamma) \), then

\[
    c_D(N((V \setminus B_p) \cup B_p^Q))) = c_D(N(V \setminus B_p)) + \tau(B_p^Q, N)
\]

\[
    sat(N((V \setminus B_p) \cup B_p^Q)), y) = 0 + \tau(B_p^Q, N) - y(N(B_p^Q)) < 0
\]

would contradict the non-negativity of satisfaction values.

The residents of the root or players that can reach the root via a zero cost path do not pay anything in the core. It is natural to assume that any other player must contribute to the construction costs, or at least there exists a core allocation when the payoff of such player is positive. Surprisingly this assertion is wrong. In Example 5.3 players 1 and 2 must bear huge costs since connecting nodes a and b is an expensive task. However, neither of them can force player 3 to pay some of their construction costs, at least not in the core. As it will follow from our next theorem, node c is free, hence the payoff of player 3 is zero in any core allocation.

**Theorem 5.7.** Node \( p \) belongs to \( F \) if and only if \( \Pi(p) = r \).

Proof. If \( p \) is unoccupied we have nothing to prove, therefore we may assume that \( |N(p)| > 0 \).

First we prove the only if part. Suppose \( p \) is a free node but its principal ancestor \( q \) is a passage. We modify the standard allocation in the following way. Let \( i_p \) be a resident of \( p \) and \( i_q \) be a resident of \( q \) and let

\[
    y(i_p) = \epsilon,
\]

\[
    y(i_q) = \hat{x}(i_q) - \epsilon,
\]

\[
    y(j) = \hat{x}(j) \text{ for any other player } j \in N,
\]

where \( \epsilon > 0 \) is a sufficiently small real number (e.g. \( \epsilon = \frac{\min_{a \in A} \delta(a)}{2|N|+1} \)). Note that \( \hat{x}(i_q) > 0 \) due to \( P2 \). We prove that \( y \in C(\Gamma_D) \). If \( S \) is such that \( i_p, i_q \in S \) then \( y(S) = \hat{x}(S) \). If \( i_p \not\in S \supseteq i_q \) then \( y(S) < \hat{x}(S) \). The only interesting case is when \( i_p \in S \not\supseteq i_q \). If \( a_q \in A(T_S) \) then
\[ y(S) = y(S \setminus i_p) + \varepsilon \leq \hat{x}(S \setminus i_p) + \hat{x}(i_q) \leq \hat{x}(S \cup N(q)) \leq c_D(S \cup N(q)) = c_D(S), \]

where we used that \( \varepsilon < \hat{x}(i_q) \). The last equality comes from the fact that \( N(q) \) can join \( S \) for free as \( S \) builds \( a_q \) anyway. If \( a_q \not\in A(T_S) \) then there is at least one shortcut in \( T_S \). Let this shortcut be \( a' \). Then

\[ y(S) = y(S \setminus i_p) + \varepsilon = \hat{x}(S \setminus i_p) + \varepsilon \leq \tau(R(S), N) + \delta(a') \leq c_D(S), \]

where we used that \( \hat{x}(S \setminus i_p) \leq \tau(R(S), N) \) by (5.1). The last inequality is obviously true since apart from the cheapest arcs that leave \( R(S) \), the members of \( S \) need to build at least one shortcut, namely \( a' \). We cannot overestimate the costs as the cheapest arc that leave the origin of \( a' \) – the cost of which is included in \( \tau(R(S), N) \) – is a zero arc due to \( P1 \).

To justify the other direction we prove that if the principal ancestor of \( p \) is the root then \( p \) is free and \( sat(N(V \setminus B_p), x) = 0 \) for any \( x \in C(\Gamma_D) \). This is a slightly stronger statement from which the \( if \) part of the proof clearly follows.

Let \( d(q) \) denote the length of the shortest path in \( T_N \) leading from \( q \) to \( r \). We proceed by induction on \( d(p) \). If \( d(p) = 0 \) then \( p \) is the root for which \( sat(N(V \setminus B_r), x) = sat(\{\emptyset\}, x) = 0 \) is satisfied. Let us assume that \( d(p) = l \) and the lemma is true for any node \( p' \) with \( d(p') < l \) where \( l > 0 \) integer. Two cases are possible. The first is when \( p \) has a free parent \( f \) (see Figure 5.11, Example I.). Let \( y \) be an arbitrary core element. Applying the induction step we obtain \( sat(N(V \setminus B_r), y) = 0 \). Both \( p \) and \( f \) are junctions therefore \( c_D(N(V \setminus B_f), y) = c_D(N((V \setminus B_f) \cup \{p\})) \). Hence \( y(N(p)) > 0 \) would imply \( sat(N((V \setminus B_r) \cup \{p\}), y) < 0 \) – a contradiction.
The second case is when \( p \) does not have a free parent. As the principal ancestor of \( p \) is the root \( p \) must be a gate. There exists paths from \( p \) to \( r \) in \( T_N \) which are semi-arc-disjoint. There may be some intermediary nodes that coincide on these paths. Let the first such node denoted by \( f \) (see Figure 5.11, Example II.). Note that the principal ancestor of \( f \) is the root (\( f \) may be the root itself) therefore we can apply the induction step. That means that \( f \) is free and \( \text{sat}(N(V \setminus B_f), y) = 0 \) for any core allocation \( y \). This also implies that \( \tau(B_f, N) = y(N(B_f)) \).

There exists two arc-disjoint path from \( p \) to \( f \) in \( T_N \). Let \( q_1 \) and \( q_2 \) be the direct ancestors of \( p \) that lie on these paths. We can separate the node set \( B_f \setminus B_p \) into two \( f \)-branch \( B_1 \) and \( B_2 \) such that \( q_1 \in B_1 \), \( q_2 \in B_2 \) and \( B_1 \cap B_2 = \{ f \} \). For instance such a partition can be obtained by coloring the path from \( q_1 \) to \( f \) red and the path from \( q_2 \) to \( f \) blue (as \( f \) is contained in both paths we can pick either one of the colors, say red). Then we color each node one-by-one in \( B_f \setminus B_p \) in the following way. Take a direct descendant of a colored node. If it has a red parent we paint it red, if it has a blue one we paint it blue. If it has both a red and a blue parent paint it arbitrarily with one color. Let \( B_1 \) contain the red nodes, while \( B_2 \) the blue ones in addition with \( f \). Indeed the node sets defined in this way are \( f \)-branches which satisfy \( B_1 \cup B_2 = B_f \setminus B_p \) and \( B_1 \cap B_2 = \{ f \} \). This leads us to

\[
y(N(B_f)) = \tau(B_f, N) = \tau(B_1 \cup B_p, N) + \tau(B_2, N),
\]

\[
[\tau(B_1 \cup B_p, N) - y(N(B_1 \cup B_p))] + [\tau(B_2, N) - y(N(B_2))] = 0.
\]

We implicitly used that \( f \) is a junction, therefore its cheapest arc is a zero arc. Furthermore, \( y(N(f)) = 0 \) since \( f \) is free. Therefore it implies no additional cost that both \( B_1 \) and \( B_2 \) contain \( f \). The node set \( B_1 \cup B_p \) is an \( f \)-branch and so is \( B_2 \), therefore the sums in the square brackets are non-negative by Lemma 5.6. It follows that \( \tau(B_2, N) = y(N(B_2)) \). Now let us move the \( B_p \) branch from \( B_1 \) to \( B_2 \). With exactly the same argument we can show that \( \tau(B_1, N) = y(N(B_1)) \).

The coalitions \( N(B_1) \) and \( N(B_2) \) pay only for their own branch’s construction cost, i.e. the cost of the cheapest arcs that leave the \( B_1 \) and \( B_2 \) branch. From \( N(B_p) = N(B_f) \setminus N(B_1 \cup B_2) \) it follows that \( \text{sat}(N(V \setminus B_p), y) = 0 \). By Lemma 5.6 the \( B^Q_p \) branch pays at most \( \tau(B^Q_p, N) \) for any \( Q \subset B_p \). In particular \( y(N(p)) = 0 \) for any core element \( y \), i.e. \( p \) is free. This concludes the proof of Theorem 5.7. □
Directed acyclic graph games

There are many coalitions whose satisfaction is zero in any core allocation. For instance it is easy to prove that if \( p \) is a passage that is a direct descendant of the root, then \( N(B_p) \) is such a coalition. In the following we characterize the set of saturated coalitions that bear this property. Let \( S_0(\Gamma) \) denote the set of saturated coalitions (cf. Definition 3.14, page 33) whose satisfaction is zero for any core allocation, formally

\[
S_0(\Gamma) \overset{\text{def}}{=} \{ S \in S(\Gamma) \mid c_D(S) = x(S) \text{ for any } x \in C(\Gamma) \}.
\]

In our next lemma we identify certain branches that pay only for their own construction cost, i.e. the cost of the cheapest arcs that leave the branch. A branch \( B_p^Q \) is called a building block if it has the following properties:

- \( p \) is a passage whose parent is free,
- all the nodes in \( Q \) are free,
- \( B_p^Q \) does not contain a free node.

**Lemma 5.8.** If \( B_p^Q \) is a building block, then \( x(N(B_p^Q)) = \tau(B_p^Q, N) \) for any core allocation \( x \).

**Proof.** Since \( \pi(p) \) is free, it is a junction and \( x(N(\pi(p))) = 0 \). We know from Theorem 5.7 that \( \text{sat}(N(V \setminus B_{\pi(p)}), x) = 0 \) for any core allocation \( x \). It follows that

\[
\text{sat}(N((V \setminus B_{\pi(p)}) \cup \{\pi(p)\}), x) = 0
\]

is also true. With a similar argument as in Lemma 5.6 it can be shown that \( x(N(B_p^Q)) \leq \tau(B_p^Q, N) \).

Each node in \( Q \) has (at least) two semi-arc-disjoint paths that lead to the root. As \( B_p^Q \) does not contain a free node one of these paths for each node avoids \( B_p^Q \). We prove this by contradiction. Let \( q \in Q \) an arbitrary free node. Suppose there exists two semi-arc-disjoint paths in \( T_N, P_1 \) and \( P_2 \) that leads from \( q \) to the root and crosses \( B_p^Q \). Let \( q_1 \in B_p^Q \cap V(P_1) \) be such that there exist no other \( q' \in B_p^Q \cap V(P_1) \) such that \( q' \prec q_1 \). Similarly let \( q_2 \) be the node closest to the root that is an element of both \( B_p^Q \) and \( P_2 \). As \( q_1 \) and \( q_2 \) lie on semi-arc-disjoint paths, one of them — say \( q_1 \) — is not \( p \). Thus the \( P_1 \) path leaves the \( B_p^Q \) node.
set at \( q_1 \) on a zero arc. There leads a path in \( T_N \) from \( q_1 \) to \( \pi(p) \) through \( B_p^Q \) that is arc-disjoint of \( P_1 \). As \( \pi(p) \) is free there leads two semi-arc-disjoint paths \( P_3 \) and \( P_4 \) from \( \pi(p) \) to the root. Without loss of generality we may assume that \( P_1 \) intersects with \( P_3 \) first (or at the same time as it intersects with \( P_4 \)). Let us denote this node by \( q^* \). Note that if \( q^* \) is a common node of \( P_3 \) and \( P_4 \) then it is a junction, otherwise the two paths would not be semi-arc-disjoint. Let \( P_A \) be the path that starts from \( q_1 \), follows \( P_1 \) till \( q^* \), then reaches the root following \( P_3 \). Let \( P_B \) be the path that originates at \( q_1 \), reaches \( \pi(p) \) using only \( T_N \)-arcs and nodes from \( B_Q^p \), and goes to the root following \( P_4 \). By construction \( P_A \) and \( P_B \) are semi-arc-disjoint, thus \( q_1 \) is free, which contradicts the assumption that \( B_Q^p \) is a building block.

It follows that there exists a path in \( T_N \) for every \( q \in Q \) that leads to the root, that does not pass through any node of \( B_Q^p \). A straightforward consequence is that \( B_Q^p \) is a proper branch and every node in \( V \setminus B_Q^p \) can reach the root by using only \( T_N \)-arcs. Note that there is no zero arc that leaves \( B_Q^p \) and enters in \( V \setminus B_Q^p \), otherwise \( B_Q^p \) would contain a free node. Thus the node set \( V \setminus B_Q^p \) corresponds to a trunk, namely to \( T_N(V \setminus B_Q^p) \). Finally, for any core allocation \( x \)

\[
c_D(N) = c_D(N(V \setminus B_Q^p)) + \tau(B_Q^p, N),
0 = [c_D(N(V \setminus B_Q^p)) - x(N(V \setminus B_Q^p))] + [\tau(B_Q^p, N) - x(N(B_Q^p))],
0 = [\text{sat}(N(V \setminus B_Q^p), x)] + [\tau(B_Q^p, N) - x(N(B_Q^p))].
\]

Both expressions in the square brackets are non-negative, thus \( x(N(B_Q^p)) = \tau(B_Q^p, N) \).

For instance in Example 5.3 the branch \( B_4 \) composes a building block. The next lemma tells us how to decompose certain branches into building blocks and free nodes.

**Lemma 5.9.** Let \( \bigcup_{j=1}^k B_{p_j}^{F_j} \) be a union of branches such that \( p_j \) is a passage, \( \pi(p_j) \in F \) and \( F_j \subset F \) for \( j = 1, \ldots, k \). Then \( \bigcup_{j=1}^k B_{p_j}^{F_j} \) can be decomposed into a disjoint union of building blocks and free nodes.
Proof. The proof proceeds by induction on the number of nodes. If $\cup_{j=1}^{k} B_{p_j}$ consist of a single node, then $k = 1$ and $B_{p_1}$ must be a building block. Now suppose the lemma is true for node sets with less than $l$ nodes and let $|\cup_{j=1}^{k} B_{p_j}| = l$. Let $B_{p_1}^Q$ be a branch where $Q = B_{p_1} \cap F$ and let $B_{p_1}^Q$ be the standard form of this branch. Note that $B_{p_1}^Q$ is a building block and it is a subset of $B_{p_1}$. Let us delete $B_{p_1}^Q$ from $B_{p_1}$. If $Q \cap B_{p_1}$ is not empty we delete those nodes too (these are free as all the nodes of $Q'$ are free). If some descendant of a node in $Q'$ has a zero cost path that leads to a node in $Q'$ then it is free therefore it can be deleted too. If we deleted all the free nodes in this way and there are still some nodes in $B_{p_1}$ then those must be passages. Let us denote these by $p'_1, \ldots, p'_K$. Note that $\pi(p'_1), \ldots, \pi(p'_K) \in F$. Hence the remaining nodes can be written as $\cup_{i=1}^{K} B_{p'_i} \cup_{j=2}^{k} B_{p_j}$. By reindexing $p'_i$ we are done as $|\cup_{i=1}^{K} B_{p'_i} \cup_{j=2}^{k} B_{p_j}| < l$. 

In Example 5.3 the branch $B_a$ can be decomposed, since $a$ is a passage and it’s parent is free. The decomposition involves two building blocks $B_a^c, B_d$ and a free node $c$.

These technical results are not just for their own sake. First with the help of these two lemmata we can characterize $S_0(\Gamma)$. Secondly we will see that the residents of $B_a^c = \{a\}$ cannot expect any help from those players who reside in a descendant of $a$ (at least not in the core). Thus these building blocks can be 'separated' from the rest of the graph without altering the core or the nucleolus of the game. We will come back to this question in the next section.

Theorem 5.10. $S \in S_0(\Gamma)$ if and only if $V(T_S)$ can be written as

$$V(T_S) = V \setminus \cup_{j=1}^{k} B_{p_j}$$

where $p_j$ is a passage $\pi(p_j) \in F$ and $F_j \subset F$ for all $j \in \{1, 2, \ldots, k\}$.

Proof. Let $|S| = n - 1$, then $c_D(S) = x(S)$ for all $x \in C(\Gamma)$ if and only if the missing player resides in a free node. Since free nodes are junctions this implies that $T_S = T_N$ and $V(T_S) = V$. This is a special case of $V(T_S) = V \setminus \cup_{j=1}^{k} B_{p_j}$ when all the missing branches are empty branches (i.e. $k = 0$). In the following we will assume that $|S| \neq n - 1$.

In the light of Lemma 5.8 and Lemma 5.9 the only if part can be verified easily. If the trunk of coalition $S$ can be represented as $V(T_S) = V \setminus \cup_{j=1}^{k} B_{p_j}$, then $V(T_S)$
is the complement of a disjoint union of building blocks and free nodes. As the residents of building blocks and the free nodes pay only for their own construction cost, the rest of the players have to pay for their own part of $T_N$. Thus from the $c_D(N) = x(N)$ equality it follows that $c_D(S) = x(S)$ for any core allocation $x$. Note that we implicitly used that every resident of $V(T_S)$ is involved in building $T_S$, that is $S$ is saturated.

Now we prove the other direction, i.e. $S \in \mathcal{S}_0(\Gamma) \Rightarrow V(T_S) = V \setminus \bigcup_{j=1}^k B_{\pi(p_j)}^{E_j}$. From Lemma 5.2 we know that we can choose a representation of $V(T_S)$ where $p_j$ - the origin of the removed $B_{\pi(p_j)}^{E_j}$ branch - is a passage for all $j \in \{1, 2, \ldots, k\}$. Furthermore, $\pi(p_j) \in V(T_S)$ and $F_j \subset V(T_S)$ for all $j \in \{1, 2, \ldots, k\}$.

If $T_S$ has a shortcut then the standard allocation induces a non-zero satisfaction value for $S$. It follows that $T_S$ is a connected subgraph of $T_N$. First let us consider a simple graph structure when only one branch is missing, that is $V(T_S) = V \setminus B_p^{Q_1}$. If $\pi(p)$ is not free then there exist a core allocation $y$ where $y(N(p)) > \tau(p, N)$. The argument is similar to the reasoning used in the first part of Theorem 5.7. As $\pi(p)$ is not free, $\Pi(\pi(p))$ is a passage. A coalition that contains a player from $N(p)$ has to use this passage or avoid it with a shortcut. In either case the standard allocation can be modified: a little amount can be transferred from $N(\Pi(\pi(p)))$ to $N(p)$ without leaving the core. Thus if the satisfaction of $N(V \setminus B_p^{Q_1})$ was zero under the standard allocation it is not zero under $y$.

Now let $V(T_S) = V \setminus \bigcup_{j=1}^k B_{\pi(p_j)}^{E_j}$ and let us use the standard representation of $V(T_S)$. Take an arbitrary $\pi(p_j)$. Basically the same argument works as above, we only need to show that $\Pi(\pi(p_j))$ is in $V(T_S)$. Suppose on the contrary that $\Pi(\pi(p_j)) \notin V(T_S)$. We know that every path from $\pi(p_j)$ to the root that lies in $T_N$ crosses $\Pi(\pi(p_j))$. Since $T_S$ is a subgraph of $T_N$ it follows that $\pi(p_j) \notin V(T_S)$. However, in the standard representation $p_j$ was chosen such way that $\pi(p_j) \in V(T_S)$ - a contradiction.

Finally, we need to prove that if $F_j \not\subset F$ then $S \not\in \mathcal{S}_0(\Gamma)$. Let $f$ be an arbitrary non-free element of a given $F_j$. There leads a path in $T_N$ from $f$ to $\pi(p_j)$ through $B_{\pi(p_j)}^{E_j}$. There leads another path in $T_S$, arc-disjoint from the previous one to the root. By our previous observation if this path contains a shortcut, then $S \not\in \mathcal{S}_0(\Gamma)$. Thus this path lies entirely in $T_N$. Since $\pi(p_j)$ is free there leads two semi-arc-disjoint paths from $\pi(p_j)$ to the root. It is impossible that the path from $f$ to the root intersects both of these paths at a passage, since then they would not be
semi-arc-disjoint. Thus there exist two semi-arc-disjoint paths from \( f \) to the root, i.e. \( f \) is free.

Notice that this direction did not require for coalition \( S \) to be saturated. Non-saturated coalitions can have zero satisfaction in the core, in particular when there are occupied free nodes in the trunk of \( S \).

A typical example of \( S_0(\Gamma) \) is a full branch that originates from the root. But not only full branches may have constant zero satisfaction in the core. For instance in Example 5.3, \( V(T_{\{1\}}) = V \setminus B_h \) satisfies the conditions of Theorem 5.10, therefore it is constructed only by player \( \{1\} \).

The interpretation of Theorem 5.10 becomes simpler when we consider the free nodes as some kind of secondary roots. The residents of a free node do not have to pay (Theorem 5.7), and the residents of a full branch that originates from a free node pay only for their own branch’s construction cost (a consequence of Theorem 5.7 and Lemma 5.6). A natural simplification would be to contract the free nodes with the root. The above results already suggest that this transformation does not alter the structure of the core. In the next section we will introduce further graph transformations that simplify the network (cf. Lemma 5.15, page 92). In addition we offer an approach to efficiently describe the core of a large family of DAG-games.

We conclude this section with a lemma that gives an upper bound on how much certain branches are willing to pay in the core. Let \( a_s \) be an arc that originates from a non-free node \( p \). We say that \( a_s \) is critical if replacing \( a_s \) with a zero arc would set \( p \) free.

**Lemma 5.11.** Let \( p \) be an arbitrary node with a critical arc \( a_s \in A_p \) and let \( B_Q^p \) be a \( p \)-branch. If \( a_s \) is not a shortcut then \( x(N(B_Q^p)) \leq \tau(B_Q^p, N) \) for any core allocation \( x \). If \( a_s \) is a shortcut then \( x(N(B_Q^p)) \leq \tau(B_Q^p, N) + \delta(a_s) \) for any core allocation \( x \).

**Proof.** If \( a_s \) is not a shortcut, then \( p \) is a passage. That is \( a_s \) is the only arc of \( p \) and \( \delta(a_s) \) is contained in \( \tau(B_Q^p, N) \). If we set \( \delta(a_s) \) to zero, then \( p \) becomes free only if \( \pi(p) \) was already free. By Theorem 5.7 \( \text{sat}(N(V \setminus B_{\pi(p)}), x) = 0 \), thus if \( x \) is a core allocation then
Figure 5.12: Schematic picture of $D$ when $a_s$ is a shortcut. Dashed lines indicate branches.

$$c_D(N((V \setminus B_{\pi(p)}) \cup B^Q_p)) = c_D(N(V \setminus B_{\pi(p)})) + \tau(B^Q_p, N),$$

$$sat(N((V \setminus B_{\pi(p)}) \cup B^Q_p), x) = 0 + \tau(B^Q_p, N) - x(N(B^Q_p)),$$

where we used that $\pi(p)$ is a junction, hence its $T_N$ arc is a zero arc. The proposition follows from the non-negativity of satisfaction values.

Let $a_s$ be a critical shortcut then. If we replaced $a_s$ with a zero arc, there would exist two semi-arc-disjoint paths from $p$ to the root. One that leads through an original zero arc of $p$, and one through $a_s$. We will use a similar argument as in the second part of Theorem 5.7. We color the nodes of the former path red while the nodes of the latter path blue. The nodes contained in both paths (e.g. the root) are assigned both colors, except for $p$ that is painted only red. Then we color each node in $V$ one-by-one in the following way. Take a direct descendant of a colored node. If it has a red parent we paint it red, if it has a blue one we paint it blue. If it has both a red and a blue parent we paint it red. Among the possible colorings we chose one where every node in $B^Q_p$ was painted red. Let $B_1$ contain the red nodes, while $B_2$ the blue ones. Every node has been assigned at least one color, i.e. $B_1 \cup B_2 = V$. The intersection of $B_1$ and $B_2$ contains nodes that coincide on the red and the blue paths. These nodes are free by construction. In $T_{N(B_1)}$ and $T_{N(B_2)}$ every player can reach the root by using only arcs of $T_N$. Thus if $x$ is an arbitrary core allocation, then
\[ c_D(N) = c_D(N(B_1)) + c_D(N(B_2)), \]
\[ c_D(N) - x(N) - x(N(B_1 \cap B_2)) = c_D(N(B_1)) - x(N(B_1)) + c_D(N(B_2)) - x(N(B_2)), \]
\[ 0 = \text{sat}(N(B_1), x) + \text{sat}(N(B_2), x), \]

where the last equality comes from the fact that \( x(N(B_1 \cap B_2)) = 0 \), as \((B_1 \cap B_2) \subseteq F\). From the non-negativity of the satisfaction values we obtain that \( \text{sat}(N(B_2), x) = 0 \). Finally,

\[ 0 \leq c_D(N(B_2 \cup B^Q_p)) \leq c_D(N(B_2)) + \delta(a_s) + \tau(B^Q_p, N), \]
\[ 0 \leq \text{sat}(N(B_2 \cup B^Q_p), x) \leq c_D(N(B_2)) - x(N(B_2)) + \delta(a_s) + \tau(B^Q_p, N) - x(N(B^Q_p)), \]
\[ 0 \leq 0 + \delta(a_s) + \tau(B^Q_p, N) - x(N(B^Q_p)). \]

5.7 Dually essential coalitions of a DAG-game

In this section we uncover the graph structure of dually essential coalitions. As it will turn out it is simple and easy to deal with. Using the dually essential coalitions we prove that contracting the free nodes with the root or rerouting critical shortcuts to the root does not alter the core or the nucleolus of the game. The main result is that whenever the size of the dually essential coalitions is polynomially bounded in the number of players and we can efficiently enumerate them, then the core and the nucleolus can be computed in polynomial time.

**Theorem 5.12.** The dually essential coalitions of the cost game \( \Gamma_D \) are the coalitions with \( n - 1 \) player and saturated coalitions whose trunks correspond to node sets of the form \( V \setminus B_q^U \), where \( B_q^U \) is a proper branch and \( q \) is a passage.

**Proof.** We have already seen in Lemma 3.15 that only saturated and \( n - 1 \) player coalitions are dually essential. By Lemma 5.2 we know that trunks of (saturated) coalitions can be generated by removing branches from \( G \). The one thing we
have to prove is that coalitions that correspond to trunks that have more missing branches are dually inessential. Let $S$ be a saturated coalition for which $V(T_S) = V \setminus \bigcup_{j=1}^{k} B_{p_j}^{Q_j}$ where $k \geq 2$. As $D$ is in canonical form there resides at least one player in each of the branches. These branches may intersect with each other in general, however – by choosing the $Q_j$ set wisely – we can always arrange them in such way that they are mutually disjoint. Among such representations we choose one where $Q_k$ is either empty or a subset of $V(T_S)$.

For the ease of presentation let us introduce the following notation $B_1 = \bigcup_{j=1}^{k-1} B_{p_j}^{Q_j}$ and $B_2 = B_{p_k}^{Q_k}$. Then let $S_1 = N \setminus N(B_1)$ and $S_2 = N \setminus N(B_2)$. In this way $S_1 \cup S_2 = N$ and $S_1 \cap S_2 = S$. To prove that $c_D(S) \geq c_D(S_1) + c_D(S_2) - c_D(N)$ holds as well it is enough to show that the following two inequalities are true.

\[
c_D(S_1) \leq c_D(S) + \tau(B_2, N) - \tau(Q_k, S) \tag{5.2}
\]
\[
c_D(S_2) \leq c_D(N) - \tau(B_2, N) + \tau(Q_k, S) \tag{5.3}
\]

Note that it takes at most $\tau(B_2, N)$ to connect the players residing at $B_2$ to $T_S$. As $B_{p_k}^{Q_k}$ is a proper branch it follows that the nodes in $Q_k$ are junctions. Since the nodes in $Q_k$ are direct ancestors of some nodes in $B_2$ they are connected with zero arcs. Therefore we can save at least $\tau(Q_k, S)$ amount of cost by connecting $Q_k$ through the branch $B_2$ and not through the arcs in $(\cup_{q \in Q_k} A_q) \cap A(T_S)$. It is possible that aside from $Q_k$ there are other nodes that can reach the root in a cheaper way using the arcs of $B_2$, but no nodes of $V(T_S)$ is forced to take a more expensive path. Summarizing the above findings we gather that

\[
c_D(S_1) \leq c_D(S) + \tau(B_2, N) - \tau(Q_k, S).
\]

We can estimate $c_D(S_2)$ by keeping track how the cost changes as we swift from $T_N$ to $T_{S_2}$. As $N(B_2)$ are not in $S_2$ we can delete $B_2$ and subtract $\tau(B_2, N)$ amount of cost from $c_D(N)$. Deleting $B_2$ from $T_N$ only the direct descendents of $B_2$ can get disconnected. Therefore the only nodes that may not be connected to the root are $Q_k$ and their descendents. By building $(\cup_{q \in Q_k} A_q) \cap A(T_S)$ – the exact same arcs that we deleted in case of $S_1$ – we can ensure that every node in $V \setminus (B_2 \cup \{r\})$ has a leaving arc. None of these arcs enter to $B_2$, thus we obtained a trunk. Therefore
the cost of reconnecting \( Q_k \) is at most \( \tau(Q_k, S) \). Altogether we can estimate the cost of \( S_2 \) by

\[
c_D(S_2) \leq c_D(N) - \tau(B_2, N) + \tau(Q_k, S).
\]

Now adding (5.2) and (5.3) together, then subtracting \( c_D(N) \) from both sides yield us the desired result. If \( k \geq 3 \) then \( T_{S_1} \) has more missing subbranches. Thus to prove dual inessentiality of \( S \) we have to refine \( c_D(S) \geq c_D(S_1) + c_D(S_2) - c_D(N) \). Also \( S_1 \) and \( S_2 \) may not be even saturated. However, by repeatedly using Lemma 3.18 and the above argument we can obtain a weakly minorizing overlapping decomposition of \( S \). A cycle in the decomposition is not possible as each refinement uses larger coalitions.

Let us illustrate how Theorem 3.12 works in practice. Consider again Example 5.3 but with the color configuration of Figure 5.13. The trunk of coalition \( S = \{1, 3, 4, 6\} \) is colored red (thickest line). This trunk has three missing subbranches indicated by green (thinnest line) and yellow (medium line) colors. We define node \( i \) to be in the green branch but not in the yellow one\(^8\). Thus the coloring serves as a partition of the node set: \( \{\{r, a, c, d, f\}, \{b\}, \{e, g, i\}, \{h\}\} \). Coalition \( S \) is dually inessential. For instance we can decompose \( S \) as the overlap of \( S_1 = \{\text{red and green players}\} = \{1, 3, 4, 5, 6, 7, 9\} \) and \( S_2 = \{\text{red and yellow players}\} = \{1, 2, 3, 4, 6, 8\} \). Indeed \( N \setminus S = (N \setminus S_1) \cup (N \setminus S_2) \) and

\[
105 = c(S) \geq c(S_1) + c(S_2) - c(N) = 109 + 307 - 311.
\]

Neither coalition \( S_1 \) nor \( S_2 \) is dually essential, so we need further refinements. The trunk of \( S_1 \) has two missing subbranches (the yellow ones). Coalitions \( S_1^1 = \{1, 3, 4, 5, 6, 7, 8, 9\} \) and \( S_1^2 = \{1, 2, 3, 4, 5, 6, 7, 9\} \) compose an overlapping decomposition of \( S_1 \). These are saturated coalitions that have one missing branch, thus they are dually essential. The trunk of \( S_2 \) has just one missing branch but \( S_2 \) is not saturated. The resident of \( i \), player 9, can join \( S_2 \) for free as \( i \in V(T_{S_2}) \). Using the decomposition argument of Lemma 3.18 we obtain the dually essential coalitions \( S_2^1 = S_2 \cup \{9\} \) and \( S_2^2 = (N \setminus \{9\}) \). Notice that \( S \) is a subset of \( S_2^1, S_2^1, S_2^2 \) and \( S_2^2 \), what is more \( N \setminus S = (N \setminus S_1^1) \cup (N \setminus S_1^2) \cup (N \setminus S_2^1) \cup (N \setminus S_2^2) \) and

\(^8\)Naturally branches do not contain arcs. For visibility reasons we colored the arcs rather than the nodes.
5.7 Dually essential coalitions of a DAG-game

![Diagram of a network with nodes and arcs labeled with numbers and a decomposition of a coalition]

Figure 5.13: The decomposition of the red coalition $S = \{1, 3, 4, 6\}$ using the green and the yellow branches. Note that the shortcut (dashed line) does not play any role in the decomposition.

$$c(S) \geq c(S_1^1) + c(S_2^2) + c(S_3^1) + c(S_2^1) - 3c(N).$$

Theorem 3.12 is surprisingly analogous to the one derived by Maschler, Potters, and Reijnierse (2010) for standard tree games (see Lemma 2.3 in the cited paper). Although, they do not speak of characterization sets the relationship between the two result is unquestionable. The next lemma is a basic corollary of the definition of trunks.

**Lemma 5.13.** Let $S \in \mathcal{DE}(\Gamma_D)$ and $V(T_S) = V \setminus B^U_p$ its standard representation. Then $q \in B^U_p \Rightarrow \Pi(q) \in B^U_p$.

**Proof.** Suppose $\Pi(q) \notin B^U_p$. There exists two semi-arc-disjoint paths in $T_N$ from $q$ to $\Pi(q)$. Only one of these paths may use $p$. Thus there is zero arc that leaves $B^U_p$, which contradicts that $T_S$ has maximum number of arcs among the cheapest trunks that connect all the players in $S$ to the root.

A network can be simplified if it has critical shortcuts. Lemma 5.11 suggests that it does not matter where a critical shortcut enters. Thus critical shortcuts can be treated as if they were pointing to the root.

**Lemma 5.14.** Let $\mathcal{D}$ be a DAG-network and $\Gamma = (N,c_\mathcal{D})$ be the corresponding game. Let $s$ be a junction with a critical shortcut $a_s$. Finally, let $\mathcal{D}'$ be a network that is obtained from $\mathcal{D}$ by rerouting $a_s$ to the root and let $\Gamma' = (N,c_{\mathcal{D}'})$. The core and the nucleolus is unchanged by this transformation, formally $C(\Gamma) = C(\Gamma')$ and $N(\Gamma) = N(\Gamma')$. 

91
Proof. It is enough to prove that $DE(\Gamma) = DE(\Gamma')$ and for any $S \in DE(\Gamma)$, $c_D(S) = c_{D'}(S)$. We will use the graph representation of dually essential coalitions that we uncovered in Theorem 3.12. It is easy to check that the $n - 1$ player coalitions have the same characteristic function value in both games. Let $S = N \setminus i$ and let $t$ denote the node where $a_s$ enters. If player $i$ resides in a junction then $T_S = T_N$, that is the trunk of $S$ does not contain any shortcuts, hence it is unimportant where $a_s$ points to. If player $i$ resides in a passage $v$ and $T_S$ does not contain $v$, then the descendants of $v$ must use an alternative path to reach the root. If a direct descendant of $v$ is a passage then $v \in T_S$, hence all its children must be junctions. Due to the canonization every such junction must have an arc that points either to an ancestor of $v$ or to a node that is unrelated to $v$. That means that if $a_s \in A(T_S)$ then $t \in T_S$ both in $D$ and in $D'$. Thus it does not matter if $a_s$ points to $t$ or to the root, the cost of $T_S$ does not change.

Now let $|S| < n - 1$. Due to Theorem 3.12 we may assume $V(T_S) = V \setminus B_p^U$. We need to prove that for any such $S \in DE(\Gamma)$ it is true that $c_D(S) = c_{D'}(S)$. Let $q_1, \ldots, q_k$ be the direct ancestors of $s$, and let $t$ be the node where $a_s$ enters. If $s \in B_p^U$ then $T_S$ is the same in both games. If $q_i \not\in B_p^U$ for some $i = 1, \ldots, k$ then $s$ can be connected to the root without using $a_s$ in both games. Hence it is indifferent whether the shortcut of $s$ enters to $t$ or to $r$. If all the $q_1, \ldots, q_k$ nodes are in the removed $B_p^U$ branch and $s$ connects to the root via $t$, then the construction cost of $S$ does not change by rerouting $a_s$ from $t$ to $r$.

Now we prove that $q_1, \ldots, q_k, t \in B_p^U$ is impossible. By contradiction suppose that all the direct ancestors of $s$ and the end node of the critical shortcut belongs to the missing branch. By definition if $\delta(a_s)$ is set to zero $s$ becomes free. That is there leads two semi-arc-disjoint paths from $s$ to the root. The paths in $T_N$ between $s$ and $r$ use a node from $q_1, \ldots, q_k, t$, hence all the paths go through $B_p^U$. If there are two semi-arc-disjoint paths that go through $B_p^U$, then only one of them may use $p$. Thus there is zero arc that leaves $B_p^U$, which contradicts that $T_S$ has maximum number of arcs among the cheapest trunks that connect all the players in $S$ to the root.

The next lemma states that free nodes can be contracted with the root.

Lemma 5.15. Let $D$ be a DAG-network and $\Gamma = (N, c_D)$ be the corresponding game. Let $p$ be any free node. Finally, let $D'$ be a network that is obtained from
The nucleolus with characterization sets

Let \( \mathcal{D} \) by contracting \( p \) with the root and let \( \Gamma' = (N, c_{D'}) \). The leaving arcs of \( p \) are deleted, while the entering arcs now point to \( r \). The core and the nucleolus is unchanged by this transformation, formally \( C(\Gamma) = C(\Gamma') \) and \( N(\Gamma) = N(\Gamma') \).

Proof. Again we could proceed by checking whether \( D\mathcal{E}(\Gamma) = D\mathcal{E}(\Gamma') \) and whether for any \( S \in D\mathcal{E}(\Gamma) \), \( c_D(S) = c_{D'}(S) \). However, by using Lemma 5.14 we can give a much simpler proof.

If \( f \) is a free node and there is a zero arc that leaves \( f \) and enters the root, then \( f \) can be contracted with the root. Similarly we can contract every node from where the root can be reached on a zero cost path. Obviously the characteristic function is unaffected by this transformation.

Now take a free node \( p \) and transform the network in the following way. We assign additional \( \varepsilon > 0 \) costs to the zero arcs of \( p \) one after another until one of them becomes a critical arc. Note that if all the zero arcs are replaced in this way, we may have to canonize the graph again. This however does not affect the game as \( P1 \) leaves the characteristic function untouched and \( p \) will still have a critical arc.

The core inequalities are continuous in the arc costs. Since each coalition uses at most one non-zero arc that leaves \( p \), each core inequality is shifted by at most \( \varepsilon \). By Lemma 5.14 we can reroute the critical arc to the root. By taking \( \varepsilon \to 0 \) we obtain a new network with the same core, but where \( p \) has a zero arc that enters the root. Thus \( p \) can be contracted with the root. \( \square \)

Note that rerouting the critical shortcuts and contracting the free nodes with the root do change the characteristic function of the game. Thus they are not equivalent transformations, as opposed to the canonization which leaves the characteristic function untouched.

5.8 The nucleolus with characterization sets

Let us examine Example 5.3 one last time. In the light of Lemmata 5.14 and 5.15 the network can be simplified (see Figure 5.14). Although, the DAG-games induced by the player-networks of Figure 5.10 and 5.14 generate different characteristic functions, they are equivalent from the point of view of the core or the nucleolus.\(^9\)

\(^9\)Note that the simplifications of Lemmata 5.14 and 5.15 alter the Shapley-value (and many other solution concept) of the game. This also sheds a light why the Shapley-value is not a good
Many technical results (e.g. Theorem 5.7 and Lemma 5.11) become apparent due to these transformations. By looking at Figure 5.14 it is clear that player 3 will not pay anything in the core. It is also evident that players 1 and 2 cannot expect help from other players. They have to construct their rather expensive links alone. Moreover, player 7 will not play more than two units in the core due to its shortcut.

Whether the core can be described efficiently with dually essential coalitions, depends on how many distinct proper branches of standard form exist in the network. Unfortunately as the next example shows there can be exponentially many dually essential coalitions in a DAG-game.

Example 5.4. Consider the DAG-network depicted in Figure 5.15. The root has only one direct descendant, namely p, while the nodes q_1, ..., q_n are the children of p. Each of the q_j nodes have one additional arc – a shortcut – that enters the root. The cost of the shortcuts are chosen in such way that their total cost is less than the cost of the T_N-arc of p. For instance let \( \delta(a_p) = 1 \), and \( \delta(a_q) = \varepsilon = \frac{1}{n+1} \) for solution concept in case of DAG-games. It assigns positive payoff to the residents of any free node that does not have a zero cost path to the root, i.e. it produces a solution outside the core.
each shortcut $a_s \in A$. Let us assume that one player resides in each node, the $j$th player at $q_j$ and the $(n+1)$st player at $p$. Let $N'$ denote the set of the first $n$ players.

For an arbitrary $S \subset N'$, $T_S$ correspond to $V \setminus B^Q_p$, where $Q_S \stackrel{\text{def}}{=} \{q_j | j \in S\}$. Thus any subset of $N'$ is dually essential. As there are $n$ player in $N'$ there are at least $2^n$ dually essential coalitions in this game.

The good news is that whenever we can efficiently enumerate all proper branches of the network we can obtain a description of the core. If $B^Q_p$ is a proper branch, then $V \setminus B^Q_p$ corresponds to a trunk $T$. However, this trunk might be not the cheapest trunk for coalition $S = N(V \setminus B^Q_p)$. It might happen that the residents of $Q$ can save some cost by constructing $B^Q_p$ or a part of it. To check whether $T$ is the cheapest trunk, involves the $NP$-hard Steiner arborescence problem. Luckily we do not need to check whether $T_S = T$ or not. If it is then $S$ is dually essential. If it isn’t, then $S$ is dually inessential and $c_D(S) = C(T_S) < C(T)$. The dually inessential coalitions are redundant in the computation of the core or the nucleolus of the game. By weakening the $c_D(S) \geq x(S)$ inequality that corresponds to a dually inessential coalition we cannot cut into the core, i.e. the coalition remains redundant.

Now we provide a large family of DAG-games where there are polynomial many proper branches. We define the width of a DAG-network as the maximum number of nodes that can be chosen from the node set, such that any two chosen node is incomparable, i.e. neither of them is an ancestor or descendant of the other.\footnote{If we think about the DAG as a partially ordered set then width is equivalent to the cardinality of the maximum antichain the poset has.}

**Lemma 5.16.** Let $D$ be a canonized DAG-network of width $k$ and let $m$ denote the number of nodes in the graph. Let $k$ be fixed, independent of $m$. There are at most $O(m^{k+1})$ number of proper branches in $D$.

**Proof.** We argue that every branch in $D$ can be characterized by choosing at most $k+1$ nodes appropriately. Then the lemma follows from the fact that $\sum_{i=1}^{k+1} \binom{m}{i} \leq k \cdot m^{k+1}$. From Lemma 5.13 we know that if $q \in B^Q_p, q \neq p$ then every key ancestor of $q$ lies inside $B^Q_p$. This implies that each $q'$ such that $p \preceq q' \preceq q$ is contained in $B^Q_p$. Let $q_1, \ldots, q_\ell$ the maximum number of incomparable nodes that we can choose from $B^Q_p$ such that no descendant of $q_j$ is an element of $B^Q_p$ for $1 \leq j \leq \ell$. We claim that the nodes $p, q_1, \ldots, q_\ell$ characterize $B^Q_p$. We use a coloring argument. First we color the nodes $q_1, \ldots, q_\ell$ then we color all their
descendants till we reach \( p \), finally we color \( p \). Notice that a node is colored if and only if it is an element of \( B^Q_p \). No descendant of the \( q_j \) nodes is an element of \( B^Q_p \) and we included all the descendant of the \( q_j \) nodes. An uncolored node \( p' \in B^Q_p \) would contradict that we choose the maximum number of incomparable nodes. Since \( \ell \leq k \) each branch can be described by at most \( k + 1 \) node.

The next theorem conveys the main result of this chapter.

**Theorem 5.17.** There exists a polynomial time algorithm in the number of players to compute the core and the nucleolus of any DAG-game that is induced by a fixed width canonized DAG-network.

The cost of the \( n - 1 \) player coalitions can be calculated in polynomial time, thus Theorem 5.17 simply follows from Lemma 5.16. Note that efficiency in Lemma 5.16 is measured in the number of nodes while the time complexity of the algorithm in Theorem 5.17 is measured in the number of players. This does not create any inconsistency as we are only interested in saturated coalitions. Hence there is a one-to-one correspondence between the \( B^Q_p \) branches and the interesting coalitions.

### 5.9 The road construction algorithm

Theorem 5.17 allows us to design a sequential LP that computes the nucleolus of large classes of canonized DAG-games in polynomial time. Given that the problem has a nice graph structure together with the fact that there exists an efficient LP for finding the nucleolus suggests that there is also a graph algorithm for it. One would wish for something like the painting algorithm of Maschler, Potters, and Reijnierse (2010) which was developed for standard tree games. In this section we will formulate an algorithm that works for DAG-networks that do not have shortcuts. Then we will present an idea how to extend this algorithm for larger game classes.

#### 5.9.1 The simplified version

In the (simplified) road construction algorithm nodes function as work stations, from where players send out workers to help build some road segments. One player
may send out many workers as there can be more than one possible path for him to reach the root. In the simplified version of the algorithm each player sends out only one worker. Let us denote the number of workers assigned to an arc $a$ by $w(a)$. The algorithm is divided into cycles. Not every player participates in every cycle. Some players finish sooner than others.

From the structure of $F$ it is clear that players that reside in a gate $p$ will note help to construct the $T_N$-arcs of $p$’s parents. If they manage to connect the principal ancestor of $p$ to the root or to another free node then $p$ itself becomes free.

Bearing this in mind we are ready to present the simplified road construction algorithm. Each cycle begins with the canonization of the network, then we repeat the following steps.

1. Set the number of workers to zero for each arc

2. For each node $p \in V \setminus F$
   
   (a) If $p$ is a passage, increase the number of workers of $a_p$ by $|N(p)| - 1$.

   (b) If $p$ is a passage, increase the number of workers by 1 for the arc that leaves the node $\Pi(\pi(p))$, whenever such arc exists.

   (c) If $p$ is a junction increase the number of workers by $|N(p)|$ for the arc that leaves the node $\Pi(p)$.

3. For each critical arc $a$ increase the number of workers by 1. These workers are associated to the residents of the node from where $a$ leaves.

4. Decrease the cost of each non-zero arc $a$ by $\min_{a' \in A} \frac{d(a')}{w(a')} \cdot w(a)$ and update $F$.

   Note that $\arg\min_{a' \in A} \frac{d(a')}{w(a')}$ shows which arcs became zero arc in the current cycle.

At time $t_i$ the $i^{th}$ cycle finishes. We distribute $t_i - t_{i-1}$ payoff among the players that participated in the construction of this phase (where $t_0 = 0$). The algorithm stops when every node becomes free. Let us denote the $i^{th}$ cycle by $P_i$. Furthermore, let $\hat{\pi}$ be the allocation that is produced by the algorithm. Note that the re-canonization is only necessary because of $P2$ the other properties ($P1$ and $P3$) are automatically fulfilled.
Although, the steps may seem a little obscure at first glance it becomes clearer when we take into consideration the intentions of the players. Due to the underlying bargaining mechanism that comes with the core, some players - the residents of the free nodes - are in special position. They do not have to pay for the service as the infrastructure would be constructed even without them. In some sense they are freeriders. The aim of the players are therefore not to construct a path to the root but rather to gain free status for their residence. They will construct those arcs that will help them achieve this goal in the fastest way.

Example 5.5. Consider the DAG-network \( D \) depicted in Figure 5.16. Clearly there are no shortcuts here, \( G \) and \( T_N \) are the same. We label the arcs after their origin. This will not lead to ambiguity since we will not need to deal with the arcs leaving \( c \) and \( g \) (i.e. the only gates). During \( \mathcal{P}_1 \) one worker builds \( a_a \) and another one \( a_b \) due to Step 3. However, \( a_d \) is constructed by four workers\(^{11}\). Player 4 and 5 send a worker due to Step 2/b, player 6 sends a worker due to Step 2/c and finally one extra worker sent by player 3 helps due to Step 3. No other arc is constructed during this phase. At \( t = 1 \) the cost of \( a_a \) and \( a_b \) is reduced by one while \( a_d \) becomes a zero arc. Now \( \Pi(g) = \Pi(d) = r \) player 3 and 6 stops participating in the construction process. In the remaining phases of the algorithm each non-zero arc is built by one worker due to Step 3. The final allocation obtained this way is \((3,5,1,2,3,1)\). It is easy to check (e.g. with the Kohlberg-criterion) that this is indeed the nucleolus of the cost allocation game corresponding to \( \Gamma_D \).

5.9.2 The proof

Let us now prove that the above algorithm indeed finds the nucleolus of the game.

\(^{11}\)In the simplified road construction algorithm workers and players are the same. Nevertheless we will continue to write 'worker' since this distinction will be meaningful later on.
Theorem 5.18. Let \( D \) be a DAG-network with no shortcuts. The simplified road construction algorithm calculates the nucleolus of the game, i.e. \( \hat{z} = N(\Gamma_D) \)

Proof. The nucleolus allocates zero cost to players that reside in a free node, but so does our algorithm. Therefore without loss of generality we can assume that in our starting network no player resides in a free node (this is not necessarily true after \( P_1 \)). The proof proceeds by induction on the number of nodes. Let us assume that the algorithm works for canonized graphs with less than \( m \) nodes. The first non-trivial case is when \( m = 2 \). Indeed if the graph consist of a single passage and the root then \( \hat{z} \) coincides with the nucleolus. Let \( \bar{D} \) denote the DAG-network that is generated after \( P_1 \). For the ease of presentation let us write simply \( \Gamma \) instead of \( \Gamma_D \) and \( \bar{\Gamma} \) instead of \( \Gamma_{\bar{D}} \) from now on.

In \( \bar{D} \) the number of nodes are strictly less as in \( D \) due to the canonization hence our assumption holds. Therefore \( \hat{z} = N(\bar{\Gamma}) + t_1 \) where \( t_1 \) is the \( |N| \) dimensional vector whose coordinates are \( t_1 \). We need to show that \( \hat{z} \) is the nucleolus of the game.

The next observation basically states that during the construction some dually essential coalitions become inessential, but this change does not occur in the opposite way. If a coalition is dually inessential it stays so even if some of the non-zero arcs in the graph are replaced with zero arcs.

Observation 8. \( DE(\bar{\Gamma}) \subseteq DE(\Gamma) \).

This observation immediately follows from the graph structure of the dually essential coalitions.

As \( DE(\bar{\Gamma}) \) is a characterization set for the nucleolus in \( \bar{\Gamma} \) by definition \( N(\bar{\Gamma}^{DE(\bar{\Gamma})}) = N(\bar{\Gamma}) \). Combining Observation 8 with Corollary 3.7 we can also conclude that \( N(\bar{\Gamma}^{DE(\bar{\Gamma})}) = N(\bar{\Gamma}^{DE(\bar{\Gamma})}) \) since enlarging a characterization set does not change the characterization property. By Theorem 3.5 the set \( \{ S \in DE(\Gamma) \mid sat_{\bar{\Gamma}}(S, x) \leq y \} \) is balanced or empty for any \( y \in \mathbb{R} \). We remind the reader that \( S_0(\Gamma) \) denotes the set of saturated coalitions whose satisfaction is zero for any core allocation.
Lemma 5.19. During $P_1$ $T_S$ is constructed by $|S|$ workers for each $S \setminus N(F) \in \mathcal{DE}(\Gamma) \cap S_0(\Gamma)$ and by $|S| + 1$ workers for each $S \setminus N(F) \in \mathcal{DE}(\Gamma) \setminus S_0(\Gamma)$, thus

$$c_D(S) = c_D(S) + |S|t_1 \quad \text{for any } S \in \mathcal{DE}(\Gamma) \cap S_0(\Gamma),$$

$$c_D(S) = c_D(S) + (|S| + 1)t_1 \quad \text{for any } S \in \mathcal{DE}(\Gamma) \setminus S_0(\Gamma).$$

Proof. If $|S| = n - 1$ and $S \in S_0(\Gamma)$ then the missing player resides in a free node\(^{12}\). The residents of free nodes do not participate in the construction process hence $T_S$ is constructed by $|S|$ workers. Otherwise the missing player helps in constructing $T_S$ due to Step 2/b or 2/c depending whether the player resides in a passage or a junction. In the following we will assume that $|S| \neq n - 1$.

First we prove that $T_S$ is constructed only by its residents for any $S \in \mathcal{DE}(\Gamma) \cap S_0(\Gamma)$. By Lemma 5.2 for any $S \in \mathcal{DE}(\Gamma) \cap S_0(\Gamma)$ the trunk $T_S$ corresponds to $V \setminus B^Q_p$. On the other hand due to Theorem 5.10, $T_S$ is a connected subgraph of $T_N$ that corresponds to $V \setminus \bigcup^k_{j=1} B^j_{p_j}$. It follows that $\pi(p)$ is free and $Q \subseteq F$. As $p$ is a direct descendant of a free node the residents of $p$ do not help to construct $V \setminus B^Q_p$. By Lemma 5.13 all the other players of $N(B^Q_p)$ build only arcs that leave from $B^Q_p$, i.e. no arcs of $T_S$. As the nodes in $Q$ are free, no descendant of $Q$ works on arcs leaving an ancestor of $Q$. In particular no outside player helps to build $B^Q_p$. As each player sends exactly one worker in the simplified road construction algorithm (by our assumption the free nodes are unoccupied), it follows that $T_S$ is constructed by exactly $|S|$ workers.

In case of $S \in \mathcal{DE}(\Gamma) \setminus S_0(\Gamma)$ the trunk $T_S$ still corresponds to $V \setminus B^Q_p$, but now $\pi(p)$ is not free and/or $Q \not\subseteq F$. Note that if $q \in B_p \setminus B^Q_p$ then $\Pi(q) \not\subseteq B^Q_p$ otherwise $q$ could not reach the root in $T_S$, since every $P_{q-r}$ path contains $\Pi(q)$. Again we conclude that no outside player helps constructing $B^Q_p$. If $\pi(p)$ is not free then the players of $N(p)$ send one worker forward to $\Pi(\pi(p))$ due to Step 2/b. Again by Lemma 5.13 all the other players of $N(B^Q_p)$ build only arcs that leave from $B^Q_p$ therefore $T_S$ is constructed by $|S| + 1$ workers.

Now we show that the second case, when $\pi(p) \in F$ but $Q \not\subseteq F$ is not possible. Let $Q \ni q \not\subseteq F$. Since $\Pi(q) \not\subseteq B^Q_p$ there leads a path from $q$ to the root which is arc-disjoint from the path between $q$ and $p$. Since $\pi(p)$ is free there leads two

\(^{12}\) Although, we assumed that no players reside in a free node prior the start of the algorithm, we have to treat this case as well in order to apply Lemma 5.19 in the induction.
semi-arc-disjoint paths from \( \pi(p) \) to the root\(^{13}\). It is impossible that the path from \( q \) to the root intersects both these paths, except if it intersects at a zero arc. Hence there exist two semi-arc-disjoint paths from \( q \) to the root, which contradicts our assumption that \( q \) is not free.

Our aim is to use the Kohlberg-criterion to verify the nucleolus. Proving balancedness of the satisfaction values \( \{ sat_\Gamma(S, z) \mid S \subseteq N \} \) is a challenging task. Thus we trace it back to the balancedness of \( \{ sat_\Gamma(S, N(\bar{\Gamma})) \mid S \subseteq N \} \). As a direct consequence of Lemma 5.19 for any \( S \in \mathcal{DE}(\Gamma) \setminus S_0(\Gamma) \)

\[
c_D(S) - \hat{z}(S) = c_D(S) + (|S| + 1)t_1 - \hat{z}(S),
\]

\[
sat_\Gamma(S, \hat{z}) = sat_\Gamma(S, N(\bar{\Gamma})) + (|S| + 1)t_1 - |S|t_1,
\]

\[
sat_\Gamma(S, \hat{z}) = sat_\Gamma(S, N(\bar{\Gamma})) + t_1.
\]

That means the non-zero satisfaction values in \( \Gamma^{\mathcal{DE}(\Gamma)} \) with respect to \( \hat{z} \) differ only by a constant from the satisfaction values of \( \bar{\Gamma}^{\mathcal{DE}(\Gamma)} \) with respect to \( N(\bar{\Gamma}^{\mathcal{DE}(\Gamma)}) \). It can be shown in a similar manner that \( sat_\Gamma(S, \hat{z}) = sat_\Gamma(S, N(\bar{\Gamma}^{\mathcal{DE}(\Gamma)})) \) for any \( S \in \mathcal{DE}(\Gamma) \cap S_0(\Gamma) \). Formally, for any \( y > t_1 \)

\[
\{ S \in \mathcal{T}^1 \mid sat_\Gamma(S, \hat{z}) = 0 \} = \{ S \in \mathcal{T}^1 \mid sat_\Gamma(S, N(\bar{\Gamma})) = 0 \},
\]

\[
\{ S \in \mathcal{T}^2 \mid 0 < sat_\Gamma(S, \hat{z}) \leq y \} = \{ S \in \mathcal{T}^2 \mid 0 \leq sat_\Gamma(S, N(\bar{\Gamma})) \leq y - t_1 \}.
\]

where \( \mathcal{T}^1 = \mathcal{DE}(\Gamma) \cap S_0(\Gamma) \) and \( \mathcal{T}^2 = \mathcal{DE}(\Gamma) \setminus S_0(\Gamma) \). Notice that it follows from inequality (5.9.2) that \( sat_\Gamma(S, \hat{z}) \geq t_1 \) for any \( S \in \mathcal{DE}(\Gamma) \setminus S_0(\Gamma) \). By Theorem 3.5 it follows that \( \hat{z} \) is the nucleolus of \( \Gamma^{\mathcal{DE}(\Gamma)} \). But \( \mathcal{DE}(\Gamma) \) is a characterization for the nucleolus in \( \Gamma \) therefore

\[
\hat{z} = N(\Gamma^{\mathcal{DE}(\Gamma)}) = N(\Gamma)
\]

\(^{13}\)Note that the same argument was used in the second part of Theorem 5.10.
5.9.3 Extending the algorithm

We will now make an attempt to extend the road construction algorithm to DAG-network with shortcuts. We demonstrate how it works on a detailed example, then we will formulate a conjecture.

Again each cycle begins with the canonization of the network, then we repeat the following steps.

1. Set the number of workers to zero for each arc
2. For each node \( p \in V \setminus F \):
   (a) If \( p \) is a passage, increase the number of workers of \( a_p \) by \( |N(p)| - 1 \).
   (b) If \( p \) is a passage, increase the number of workers by 1 for the arc that leaves the node \( \Pi(\pi(p)) \), whenever such arc exists. Furthermore, increase the number of workers by 1 for those shortcuts that originate from any path between \( p \) and \( \Pi(\pi(p)) \) in \( T_N \) such that by using these arcs one of the ancestors of \( \Pi(\pi(p)) \) can be reached.
   (c) If \( p \) is a junction increase the number of workers by \( |N(p)| \) for the arc that leaves the node \( \Pi(p) \). Furthermore, increase the number of workers by \( |N(p)| \) for those shortcuts that originate from any path between \( p \) and \( \Pi(p) \) in \( T_N \) such that by using these arcs one of the ancestors of \( \Pi(p) \) can be reached.
3. For each critical arc \( a \) increase the number of workers by 1. In case \( a \) is a \( T_N \)-arc these workers are associated to the residents of the node from where \( a \) leaves. If \( a \) is a shortcut these workers are not associated to any particular player\(^{14} \).
4. Decrease the cost of each non-zero arc \( a \) by \( \min_{a' \in A} \frac{d(a')}{w(a')} \cdot w(a) \) and update \( F \).

Note that \( \arg\min_{a' \in A} \frac{d(a')}{w(a')} \) shows which arcs became zero arc in the current cycle.

The output allocation is denoted by \( z \), all the other notations remain the same.

The only real difference comes with Step 2/b and 2/c where the shortcuts are treated. Notice that players construct only those shortcuts that help them to get

\(^{14}\text{There is a correspondence between workers and players in this case as well. However, it would only make the proof more difficult, thus we omit it.} \)
connected to a free node. A player can send out many workers and can build multiple number of shortcuts in the same time.

\[
\begin{align*}
\text{Figure 5.17: A canonized DAG-network before the start of the RC-algorithm} \\
\text{and at time } t = 2.
\end{align*}
\]

**Example 5.6.** Consider the canonized network depicted in Figure 5.17 and the corresponding DAG-game \( \Gamma \). At time \( t = 2 \) the shortcut leaving node \( b \) becomes a zero arc. This marks the end of the first cycle, i.e. \( t_1 = 2 \). Table 5.2 gathers the characteristic function values prior the start of the algorithm and at \( t_1 \). We employ the notation \( \bar{D} = D_{|t=2} \) to distinguish between these two states. The 4th and 8th column of the table demonstrates the meaning of Lemma 5.19. The trunk \( T_{\{3\}} \) contains three arcs, the \( T_N \)-arcs of \( d \) and \( c \) and the shortcut of \( b \). During \( P_1 \) it is constructed by 4 worker (by player 2, 3, 4 and an extra player due to Step 3). The cardinality of the set \( \{3\} \) is only one, hence the sum \( c_D(\{3\}) - c_{\bar{D}}(\{3\}) - |\{3\}|t_1 \) is big. In comparison for any dually essential coalition \( S \) with a non-zero satisfaction value\(^{15}\) \( c_D(S) - c_{\bar{D}}(S) - |S|t_1 = t_1 \). This verifies the second statement of Lemma 5.19. Namely, that the dually essential coalitions are all constructed by \( |S| + 1 \) workers. This observation is the basis of the induction used in Theorem 5.18. Note that Lemma 5.19 and Theorem 5.18 were designed for networks with no shortcuts, but work here perfectly well.

The nucleolus of the game is \( (3, 4, 5, 3, 3) \). The next array contains the coalitions and the corresponding satisfaction values of the game in non-decreasing order.

\(^{15}\)Note that \( S_0 \) consist only of the grand coalition here.
Directed acyclic graph games

| $S$   | $c_D(S)$ | $c_{D'}(S)$ | $c_D(S) - c_{D'}(S) - |S|\mu_1$ | $S$ | $c_D(S)$ | $c_{D'}(S)$ | $c_D(S) - c_{D'}(S) - |S|\mu_1$ |
|------|----------|-------------|-------------------------------|-----|----------|-------------|-------------------------------|
| $\{1\}$ | 5        | 1           | 2                             | $\{5\}$ | 9        | 3           | 4                             |
| $\{2\}$ | 10       | 4           | 4                             | $\{1,3\}$ | 14       | 4           | 6                             |
| $\{1,2\}$ | 11       | 5           | 2                             | $\{2,5\}$ | 17       | 7           | 6                             |
| $\{3\}$ | 14       | 6           | 6                             | $\{1,2,5\}$ | 18      | 8           | 4                             |
| $\{1,3\}$ | 15       | 7           | 4                             | $\{3,5\}$ | 17       | 7           | 6                             |
| $\{2,3\}$ | 14       | 6           | 4                             | $\{1,3,5\}$ | 18      | 8           | 4                             |
| $\{1,2,3\}$ | 15      | 7           | 2                             | $\{2,3,5\}$ | 17     | 7           | 6                             |
| $\{4\}$ | 9        | 3           | 4                             | $\{1,2,3,5\}$ | 18     | 8           | 2                             |
| $\{1,4\}$ | 14       | 4           | 6                             | $\{4,5\}$ | 9        | 3           | 2                             |
| $\{2,4\}$ | 17       | 7           | 6                             | $\{1,4,5\}$ | 14      | 4           | 4                             |
| $\{1,2,4\}$ | 18      | 8           | 4                             | $\{2,4,5\}$ | 17     | 7           | 4                             |
| $\{3,4\}$ | 17       | 7           | 6                             | $\{1,2,4,5\}$ | 18     | 8           | 2                             |
| $\{1,3,4\}$ | 18      | 8           | 4                             | $\{3,4,5\}$ | 17     | 7           | 4                             |
| $\{2,3,4\}$ | 17      | 7           | 4                             | $\{1,3,4,5\}$ | 18     | 8           | 2                             |
| $\{1,2,3,4\}$ | 18     | 8           | 2                             | $\{2,3,4,5\}$ | 17    | 7           | 2                             |
| $\{\emptyset\}$ | 0       | 0           | 0                             | $\{1,2,3,4,5\}$ | 18   | 8           | 0                             |

Table 5.2: Detailed calculation of Example 5.6. Dually essential coalitions are colored red and marked with boldface. Note that the satisfaction values of $\mathcal{DE}(\Gamma)$ in $\Gamma$ and $\bar{\Gamma}$ differ by a constant of 2.

Indeed the smallest satisfaction values belong to the dually essential coalitions. Note that the set of coalitions lying left to any separation mark $|$ is balanced, therefore the Kohlberg-criterion applies.

We now have enough munition to formulate a conjecture.

**Conjecture 5.20.** Let $\mathcal{D}$ be a canonized DAG-network, where all the shortcuts are critical. The road construction algorithm calculates the nucleolus of the game, i.e. $z = \mathcal{N}(\Gamma_\mathcal{D})$

We will not dive into the intricacies of the proof of Conjecture 5.20. It follows the scheme that we have seen in the simplified version. However, let us at least address the difficulties that arise during the proof.

- The non-zero satisfaction values of the dually essential coalitions in $\Gamma$ and $\bar{\Gamma}$ do not just differ by a constant.
5.9 The road construction algorithm

- In order to obtain a collection of coalitions whose satisfactions differ by a constant in $\Gamma$ and $\bar{\Gamma}$ and which characterizes the nucleolus we need to design an ad-hoc characterization set $K(\Gamma)$. This can be done by sorting out some elements of $\mathcal{DE}(\Gamma) \cup \mathcal{S}_0(\Gamma)$.

- For this newly designed characterization set Observation 8 will not hold anymore, that is $K(\bar{\Gamma}) \not\subseteq K(\Gamma)$. Thus, we need to prove $K(\Gamma)$ characterize the nucleolus in $\bar{\Gamma}$.

Parts of the proof are already worked out by the author, but the proof is too lengthy and technical and involves a lot of side cases, therefore we content ourselves by stating our claim as a conjecture.

5.9.4 Applicability of the algorithm

Let us further elaborate on the applicability of the algorithm. Example 5.6 suggests that the road construction algorithm works on a larger class of DAG-games. What is more, it can be verified with the same proof technique. However, the DAG-network in Figure 5.17 is somewhat special, it has only critical shortcuts. The next example shows that problems may arise when the graph has non-critical shortcuts.

**Example 5.7.** Consider the network depicted in Figure 5.18. Here we have two non-critical shortcuts, one that leaves node $d$ and one that leaves $e$. If we proceed according to the road construction algorithm (cf. $\mathcal{D}_{[t=1,3]}^{RC}$) these shortcuts become zero arcs at $t = 1.5$. The shortcut of $d$ is constructed by player 4 and 6, while the shortcut of $e$ by player 5 and 6. In the next cycle the $T_N$-arc of $a$ is constructed by 5 workers, hence the road construction algorithm allocates 2.7 cost to player 1. This, however, is not the nucleolus of the game, which is given by the vector $(3, 5, 6, 3, 3, 3)$.

According to Step 2/c player 6 sends workers to both shortcuts and its resident’s principal ancestor. The network $\mathcal{D}_{[t=2]}^{act}$ depicts a situation when the workers sent by player 6 operate at a different speed – half-speed to be exact – on the shortcuts as if player 6 could not make up his mind which shortcut to chose. The rest of the graph is constructed as normal. Proceeding like this we can derive the nucleolus.
However, the presence of non-critical shortcuts does not always cause problems. Figure 5.19 depicts a DAG-network $\mathcal{D}$ and the stages of the algorithm. There are two shortcuts $a_2$ and $a_6$ from which the latter one is not critical. The algorithm derives the allocation $z = (3, 5, 2, 3)$, which – we can easily verify with the Kohlberg-criterion – is indeed the nucleolus of the game.

This latter example suggests that the algorithm works on even larger classes of DAG-games. Although, Example 5.7 shows an exception, it also points toward the solution. By setting different paces for different workers the road construction algorithm can be extended further. The question presents itself: Which is the largest game class for which a road construction type algorithm works? Can we compute the nucleolus of any canonized network in polynomial time? We do not venture to decide this question one way or another.

### 5.10 Time complexity issues

There are three computational tasks with respect to DAG-games where time complexity is an issue: finding the principal ancestors, performing a cycle of the road construction algorithm, and checking for critical shortcuts.
construction algorithm and enumerating the proper branches. Let us briefly address these three task separately. We will denote the number of nodes and the number of arcs in the graph by \( m \) and \( l \), respectively.

Identifying the principal ancestor of each node in \( T_N \) is a time consuming task. The following method provides a simple treatment of the problem. Double every zero arc and set the capacity of every arc to 1 in \( T_N \). Now determining the key ancestors of a node \( p \) becomes easy. Let \( f_{pq} \) denote the maximum flow between \( p \) and \( q \). If \( f_{pq} \geq 2 \), then \( q \) is a key ancestor of \( p \). The key ancestor closest to the root will be the principal ancestor. This is a very costly way to map the principal ancestors of the nodes, it takes around \( O(m^5) \) time. It seems likely that by dynamic programming the running time can be reduced to \( O(m^3) \) or even lower.

If we know all the principal ancestors then performing a cycle in the road construction algorithm takes linear time in the number of nodes. With each cycle of

**Figure 5.19:** The steps of the road construction algorithm. In order to retain comparability the last two networks are depicted without canonization.
the algorithm an arc becomes zero arc. Thus the algorithm finishes $O(m^5 \cdot l)$ time.

Finally, let us mention that enumerating the proper $B^Q_p$ branches in the network does not need $O(m^{k+1})$ operations on average. Firstly, the origin of the branch $(p)$ must be a passage. In addition all the nodes in $Q$ must be junctions with an arc entering into $V \setminus B^Q_p$. Lastly whenever a gate $g$ is contained in $B^Q_p$, then all the ancestors of $g$ must be included in the branch as well. These restriction speed up the enumeration process substantially.
Chapter 6

Conclusion

This thesis focused on the computational aspects of the nucleolus. We established several methods on the computation of the nucleolus as well as the corresponding theory. Furthermore, for two important classes of games we presented efficient algorithms that calculate the nucleolus in polynomial time. Having accomplished our main objectives we now discuss the results.

6.1 How to find the nucleolus

We reviewed the axiomatization of the nucleolus and its relation to other solution concepts in Chapter 2. The applicability of the axiomatization is somewhat limited. It is usually hard to confirm whether a solution admits the reduced game property. Only when the reduced game falls into the same game class as the original game it is possible to use the axiomatization. Brânzei, Inarra, Tij, and Zarzuelo (2006) used this technique to find the nucleolus of airport profit games.

Another perhaps more famous example that uses RGP is the Theorem of Aumann and Maschler (1985) which states that the Talmud-rule yields the nucleolus in case of bankruptcy games.

In the first section of Chapter 3 we reviewed the literature on the various linear programs that were developed for computing the nucleolus. By itself the linear programming approach is not an effective tool as we either need a sequential LP with exponential many programs or a unique maximization problem with exponential many constraints. The linear programming approach is often used to calculate
the nucleolus in practice when no theory is available. There are quite a few instances when the opposite is true: an LP helps to derive a theoretical result, for an example see (Kamiyama, 2014) or (Kern and Paulusma, 2003).

Another standard technique is to verify the nucleolus through the satisfaction vector, i.e. guessing what the solution is, then checking whether the satisfaction vector of the proposed allocation is lexicographically maximal. For example such a method is used to find the nucleolus of voting games with a non-empty core (Elkind, Goldberg, Goldberg, and Wooldridge, 2009) and the nucleolus of standard tree games (Megiddo, 1978). We also demonstrated how this method works in Theorem 4.7 when we gave an elementary proof of the above mentioned result of Aumann and Maschler (1985). Although, we disguised our argument by representing the problem with a hydraulic framework, in real we just showed the lexicographical optimality of the satisfaction vector.

A more advanced technique of this kind when the balancedness of the satisfaction vector is examined. The Kohlberg-criterion is mostly used in combination of the above listed methods as proving balancedness for abstract coalition structures can be a challenging task. In comparison verifying balancedness of a given collection of coalitions can be done with a simple LP (cf. Lemma 3.2, page 26). We reviewed the theory related to this criterion in Section 3.2. The Chinese postman game that was introduced by Granot, Hamers, Kuipers, and Maschler (2011) provides an example where balancedness is crucial in the proof and the effectiveness of the painting algorithm of Maschler, Potters, and Reijnierse (2010) is also proven with the help of Kohlberg-criterion. The proof of the simplified road construction algorithm that we presented in Chapter 5 also relies on this criterion.

Surprisingly few papers use the concept of characterization-sets explicitly – the main theoretical advancement that was developed parallel by Granot, Granot, and Zhu (1998) and Reijnierse and Potters (1998). Some papers like (Kamiyama, 2014) and (Brânzei, Solymosi, and Tijs, 2005) exploit this idea but there are many others which use it unknowingly. For instance Maschler, Potters, and Reijnierse (2010) identify a collection that determines the core and nucleolus of standard tree games which is in fact a characterization set. However, they do not make the connection between their method and the above mentioned two papers\(^1\).

\(^{1}\)Although, they must have been aware of it since Hans Reijnierse coauthors both papers. Perhaps this is due to the fact that the working paper version of (Maschler, Potters, and Reijnierse,
One of the primary contributions of this thesis is the expansion of the theory of characterization sets. When the game in question is well-structured, characterization sets can simplify the proof substantially. Even when the structure is more complicated characterization sets can make the proof significantly simpler or at least possible. The main advantage of characterization sets is their algebraic formalism. The proofs need less bag-tricks and more mechanical computation which is perhaps aesthetically less pleasing but much more effective in terms of results.

6.2 Characterization sets

A characterization set is a collection of coalitions that determine the nucleolus by itself. Granot, Granot, and Zhu (1998) proved that if the size of the characterization set is polynomially bounded in the number of players, then the nucleolus of the game can be computed in strongly polynomial time. A collection that characterizes the nucleolus in one game need not characterize it in another one. Thus we are interested in properties of coalitions that characterize the nucleolus independently of the realization of the coalitional function. Huberman (1980) was the first to show that such a property exists. He introduced the concept of essential coalitions which are coalitions that have no weakly minorizing partition (cf. Definition 3.9). Granot, Granot, and Zhu (1998) provided another collection that characterize the nucleolus in cost games with non-empty cores. Saturated coalitions contain all the players that can join the coalition without imposing extra cost.

Using the concept of dual game we introduced two new characterization sets: dually essential and dually saturated coalitions. We showed that each dually inessential coalition has a weakly minorizing overlapping decomposition which consists exclusively of dually essential coalitions. Thus dually essential coalitions determine the core, and if the core is non-empty they determine the nucleolus as well. If every player contributes to the value of a coalition then such coalition is called dually saturated. We showed that dually saturated coalitions also determine the core and the nucleolus of a TU-game.

The larger a characterization set is the easier to uncover it in a particular game class. However, with smaller characterization set it comes a faster LP. Hence there

2010) dates back before the paper on the $B$-nucleolus. As a matter of fact (Reijniesse and Potter, 1998) contains a short passage on standard tree games, which describes a characterization set for this game class.
is a tradeoff between the difficulty in identifying the members of a characterization set and its efficiency. In order to exploit this technique we analyzed the relationship of the four known characterization sets. We proved that essential coalitions are a subset of dually saturated coalitions in monotonic profit games and that dually essential coalitions are a subset of saturated coalition in case of monotonic cost games. We showed that in general essential and dually essential coalitions do not contain each other. In fact for additive games their intersection is empty. We proved that if the grand coalition is vital then the intersection of essential and dually essential coalitions forms a characterization set itself. We demonstrated how these novelties can be applied in case of bankruptcy and directed acyclic graph games.

6.3 Bankruptcy games

The game theoretic analysis of bankruptcy games was initiated by O’Neill (1982), but Aumann and Maschler (1985) made the problem really popular by proving the equivalence of the Talmud-rule and the nucleolus. Although, their result was spectacular the proof used concepts like the reduced game or kernel. Many believed that an elementary proof should exists for this problem. Benoît (1997) was the first to publish a simplification, although he still needed long pages of computation to reach the desired result. Recently Fleiner and Sziklai (2012) managed to provide an elementary and instructive proof with the help of the hydraulic framework that was developed by Kaminski (2000). This proof along with the characterization set approach is presented in Chapter 4.

The differences between the approaches are remarkable. The hydraulic proof seems to be straightforward enough but the elapsed time between the original and this proof signals that a few subtle tricks were needed to overcome the difficulties. It seems that for each particular game class a different idea is needed to compute the nucleolus. Bankruptcy games make a textbook examples why characterization sets are so resourceful. The benefits of Theorem 3.6, 3.13 and 3.20 all come together. First we checked whether any of the known characterization sets are small enough to apply an LP. With basic computation we managed to unfold the structure of essential and dually essential coalitions. It turned out that the intersection of these two sets is of linear size in the number of players. Next we confirmed that the grand coalition is vital, thus this set indeed describes the nucleolus.
Naturally characterization sets do not make other approaches obsolete. With the help of the hydraulic framework we uncovered the relationship of the nucleolus and the proportional rule which in other case would have remained hidden. Also from didactical point of view the hydraulic representation is much more instructive, it is much easier to interpret notions like self-duality and consistency.

6.4 Directed acyclic graph games

Directed acyclic graph games are generalization of standard tree games. In fact every monotonic and subadditive cost game can be modelled as a DAG-game. This structure has not been previously analyzed from a cooperative game theoretic perspective. Similarly to minimum cost spanning tree games they model situations when players share the cost of constructing a network that enables them to use some kind of service. Possible applications include the cost allocation of infrastructural projects, carpooling, and sharing maintenance cost of service networks.

The characteristic function of any DAG-game is non-negative, monotone and strongly subadditive, but not necessarily submodular. Unlike to standard tree and minimum cost spanning tree games, the core of a DAG-game can be empty. We showed a sufficient condition for the balancedness of the game. We also introduced a graph canonization process and showed that all the games that satisfy this condition can be canonized. Similarly to monotonic MCST games it is computationally hard to calculate the characteristic function value of a given coalition. Finding a cheapest trunk that connects every player in the coalition to the root is equivalent to the acyclic directed Steiner tree problem, which is $NP$-hard.

We proved many structural result related to DAG-games. First we identified the so-called free nodes in the graph. Due to the underlying bargaining mechanism that comes with the core, some players - the residents of the free nodes - are in special position. They do not have to pay for the service as the infrastructure would be constructed even without them. Next we characterized saturated coalitions whose satisfaction value is zero in any core allocation - these are subnetworks that pay only for their own construction cost. Finally, we uncovered the graph structure of dually essential coalitions, hence we gained a collection that describes both the core and the nucleolus. Unfortunately there can be exponential many dually essential coalitions in a canonized DAG-game. Thus this approach is not
always applicable. However, we proved that if the graph has a fixed width, which is independent of the number of nodes then the nucleolus can be found in polynomial time.

We also gave a graph based algorithm to compute the nucleolus of DAG-networks that have no shortcuts, and suggested an extension of the algorithm where only critical shortcuts are allowed. It seems that our algorithm works on even bigger class of games, but not on all canonized DAG-games. Our aim was to demonstrate the usefulness of characterization sets rather than to find the nucleolus for every DAG-game, thus we did not pursue the question further.

Let us just say a few remarks regarding the unsolved cases. We do not venture to conjecture that a polynomial time algorithm exists for all DAG-games. The result of Faigle, Kern, and Kuipers (1998), namely, that finding the nucleolus of MCST games is $NP$-hard, should be at least a warning sign. However, their proof uses a reduction to the minimum cover problem and relies heavily on the undirectedness of the edges of the graph. Thus it does not necessarily indicate that the same difficulties exists in DAG-games\footnote{Note that DAG-games include all the mMCST games, but the result of Faigle, Kern, and Kuipers (1998) is about non-monotonic MCST games.}. In a DAG-network payments flow in one direction, toward the root which makes the players hierarchically structured. Thus it seems possible that some kind of road construction algorithm works at for any kind of DAG-networks. However, as the next example shows the underlying bargaining mechanism is more complicated.

\begin{figure}[h]
\centering
\includegraphics[width=0.2\textwidth]{network.png}
\caption{A network where there is an unoccupied passage with more entering arc.}
\end{figure}

Consider the DAG-game induced by the network in Figure 6.1. If the shortcut that originates from node $a$ was not present, then player 2 would pay 3 units according to the nucleolus. If the other shortcut was missing then player 1 would pay 2. Altogether they would pay 5 units which does not cover the cost of building $T_N$. 

\section{Conclusion}

\begin{enumerate}
\item Conclusion
\item always applicable. However, we proved that if the graph has a fixed width, which is independent of the number of nodes then the nucleolus can be found in polynomial time.
\item We also gave a graph based algorithm to compute the nucleolus of DAG-networks that have no shortcuts, and suggested an extension of the algorithm where only critical shortcuts are allowed. It seems that our algorithm works on even bigger class of games, but not on all canonized DAG-games. Our aim was to demonstrate the usefulness of characterization sets rather than to find the nucleolus for every DAG-game, thus we did not pursue the question further.
\item Let us just say a few remarks regarding the unsolved cases. We do not venture to conjecture that a polynomial time algorithm exists for all DAG-games. The result of Faigle, Kern, and Kuipers (1998), namely, that finding the nucleolus of MCST games is $NP$-hard, should be at least a warning sign. However, their proof uses a reduction to the minimum cover problem and relies heavily on the undirectedness of the edges of the graph. Thus it does not necessarily indicate that the same difficulties exists in DAG-games\footnote{Note that DAG-games include all the mMCST games, but the result of Faigle, Kern, and Kuipers (1998) is about non-monotonic MCST games.}. In a DAG-network payments flow in one direction, toward the root which makes the players hierarchically structured. Thus it seems possible that some kind of road construction algorithm works at for any kind of DAG-networks. However, as the next example shows the underlying bargaining mechanism is more complicated.
\end{enumerate}
A simple calculation shows that the nucleolus payoffs are 3 and 5 units respectively, but exactly how the players reach this settlement is still obscure.

6.5 Future research

Finding examples where the four known characterization sets are applicable have many benefits. It may deepen our understanding about the nucleolus and could possibly lead to the discovery of further characterizing properties. Thus it is worthwhile to examine games where an algorithm for the nucleolus already exists and analyze them from the perspective of characterization sets.

The research of directed acyclic graph games can be rewarding on its own. This is a fairly huge game class with many possible applications. Proving the conjecture which we formulated in the end of Chapter 5 could be the first step. Another exciting problem from this topic is the conversion of mMCST games. Although, we have proven that every mMSCT game is a DAG-game, the characteristic DAG-representation cannot be used in practice. A proper conversion would inflate neither the node nor the arc set, and would only need polynomial number of steps. It is unclear how hard this task is.

There are some variants of the nucleolus that are gaining more attention recently. The per capita nucleolus is one such rule. Groote Schaarsberg, Borm, Hamers, and Reijnierse (2013) provided a balancedness criterion for the per capita nucleolus similar to Kohlberg’s. It is not known whether the Theorem of Granot, Granot, and Zhu (1998) about characterization sets (i.e. Theorem 3.6) can be translated to the per capita version. Such a result would enhance the research of this nucleolus variant greatly. Another important theoretical question is how to extend the nucleolus to partition function form games.

There are some possible research directions on the application side as well. A particular generalization of bankruptcy games is especially promising. In a so called multilateral bankruptcy game agents have initial endowments and have claims on each other. Such games can model various economic situations including clearings on stock exchanges under central clearing parties, interbank liabilities under bank authorities or sovereign debts under international organizations (e.g. IMF). However, defining the characteristic function in such cases is a delicate issue. The concept is quite new and there are a few competing models (Bjørndal and
Jörnsten, 2010; Demange, 2012; İlkilic and Kayi, 2012). Although, Bjørndal and Jörnsten (2010) discusses the nucleolus as a possible solution, the other two papers do not mention it.

Applying game theory in finance has a long tradition. There are only but a few examples of applications related to infrastructural developments. This fact is even more surprising considering the amount of literature that was developed with connection to minimum cost spanning tree games and its variants. In recent years game theory became more recognized due to the publicity generated by the Nobel Memorial Prize in Economic Sciences and by the prominent game theorist who received the prize. We expect that the cost sharing methods of cooperative game theory will gain more popularity. Applying these methods in water and sanitation networks in Latin America looks particularly promising (Leoneti, do Prado, and de Oliveira, 2011).
Summary

This thesis focused on the computational aspect of the nucleolus. We aggregated the known results and analyzed the existing methods in practice. One of the main contributions is the expansion of the theory of characterization sets. We introduced two new characterization sets, the dually essential and dually saturated coalitions that can substantially simplify the task of finding the nucleolus in balanced games. We demonstrated how these novelties can be applied in case of bankruptcy and directed acyclic graph games.

- In Chapter 2 we defined the nucleolus and reviewed its axiomatization. We also discussed some nucleolus related solution concepts and elaborated on its possible extension on partition function form games.
- In Chapter 3 we introduced and discussed the concept of characterization sets in depth. We proposed two new characterization sets and analyzed their relation with the known ones. In particular we showed that the intersection of essential and dually essential coalitions form a characterization set, if the grand coalition is vital.
- In Chapter 4 we introduced bankruptcy games and using a hydraulic framework we proved various result in connection with them. Most importantly we gave an elementary proof of Aumann & Maschler's famous theorem, namely that the solution generated by the Talmud rule coincides with the nucleolus. We also demonstrated how characterization sets can be used to simplify the computation.
- In Chapter 5 we introduced a new game class: directed acyclic graph games. We proved that every monotonic and subadditive cost game can be modeled as a DAG-game. We proved various other structural results related to the core, uncovered the graph structure of dually essential coalitions and showed that for a large class of DAG-games the nucleolus can be computed in polynomial time.
- Finally in the last chapter we summarized the results and pointed out some open problems.
Doktori értekezésem a nukleolusz kiszámítási módszereiről szól, különös tekintettel a karakterizációs halmazok elméletére. Az ismert eredmények összefoglalása mellett a gyakorlatban is bemutatom az egyes módszerek alkalmazhatóságát. Bevezetek két új karakterizációs halmazt, a duálisan lényeges és a duálisan telített koalíciókat, és két fontos játkosztályon bemutatom hogyan egyszerűsíthető a nukleolusz meghatározását.

- A 2. fejezetben definíalom a nukleoluszt és ismertetem az axiomatizációját. Számos a nukleoluszhoz kötődő megoldáskoncepciót is bemutatok, valamint röviden értekezem a nukleolusz partíciós függvény formájú játékokra való kiterjesztéséről.
- A 3. fejezetben tárgyalom a karakterizációs halmazok elméletét. Bevezetek két új karakterizációs halmazt és elemzem a már ismert karakterizációs halmazokkal való viszonyukat. Egy fontos részeredmény, hogy bebizonyítom, hogy amennyiben a nagykOAAlíció vitális, úgy a lényeges és a duálisan lényeges koalíciók metszete maga is karakterizációs halmaz alkot.
- A 4. fejezetben bevezetem a csőlátékokat és hidraulikus rendszereket felhasználva számos ezzel kapcsolatos eredményt bizonyítok. A legfontosabb ezek közül az Aumann és Maschler híres tételére adott elemi bizonyítás. Ezenfelül a karakterizációs halmazok alkalmazhatóságát is elemzem ezen a játékosztályon.
- Az 5. fejezetben egy új, hálózatokon értelmezett költségjátékozat vezetek be: az irányított aciklikus gráf játékokat. Belátható, hogy minden monoton, szubadditív költségjátékok modellezhető DAG-játékként. Számos a maggal kapcsolatos eredményt is bemutok, karakterizálom a duálisan lényeges koalíciók gráf-struktúráját, valamint a DAG-játékok egy nagy részcsaládjára polinomiális futásidőjű algoritmust adok a nukleolusz kiszámítására.
- Az utolsó fejezetben összegzem az eredményeket és bemutatok néhány megoldatlan kérdést is.
References


References


References


References


References


Index

additive solution, 10
aggregate monotonicity, 12
airport game, 35, 60
Alabama paradox, 18
algorithm
  painting, 61
  polynomial time, 4
allocation
  efficient, 9
  standard, 76
ancestor, 74
  direct, 74
  key, 77
  principal, 77
anonymity, 11
anti-core, 19
apportionment problem, 18
assignment game, 26
average lexicographic value, 17
balanced
  collection, 25, 36
  game, 10
balancing weights, 25
bankruptcy
  game, 40
  problem, 40
  rule, 40
big O notation, 4
Bird-rule, 62, 76
branch, 74
decomposition, 83
full, 74
in standard form, 75
origin, 74
proper, 75
building block, 82
canonization, 71
capillary, 45
characteristic DAG-representation, 68
characterization set, 27
child, 74
claims problem, 40
claims vector, 40
coloring argument, 81, 87, 95
concave game, 19
consistency, 42
constrained egalitarian rule, 43
constrained equal awards, 41
constrained equal losses, 41
Contested Garment
  consistent solution, 42
  principle, 41
convex game, 8
cooperative game, 3
  with coalition formation restrictions, 27
core, 10
cost game, 19
covariance under strategic equivalence, 10
critical arc, 86
critical shortcut, 91
cycle
  of the RC-algorithm, 102
  of the simplified RC-algorithm, 97
  in the decomposition, 36
decision problem, 4
descendant, 74
direct, 74
directed acyclic graph
  game, 66
  network, 64
dual game, 30
dual rule, 41
dualization, 31
dually essential coalitions, 31
dually inessential coalitions, 31
dually saturated coalitions, 33
dummy-player, 11
dummy-player property, 11
Dutta-Kar solution, 63
ε-core, 16
dependence property, 11
esential coalitions, 30
estate, 40
excess, 9
externality, 21
fixed tree games, 61
free node, 78
gate, 65
house monotonicity, 18
hydraulic, 44
connected, 45
disconnected, 45
framework, 44
talmudic, 45
imputation saving reduced game, 13
imputation saving reduced game property, 13
imputation set, 9
individual rationality, 20
individually rationality, 9
inesential coalitions, 30
irrigation game, 15
junction, 65
k-cylinder hydraulic, 52
kernel, 17
Kohlberg-criterion, 25
least core, 16
lexicographically smaller vector, 14
leximin rule, 18
lower closure, 34
marginal contribution, 13
matching game, 26
maximum antichain, 95
minimum cost spanning forest game, 63
minimum cost spanning tree, 62
  monotonic, 62
monotonic game, 7
multiple source games, 63
museum-pass problem, 44
NP, 5
NP-complete, 5
NP-hard, 2, 5
nucleolus, 15
    modified, 18
    per capita, 17
    with respect to X, 14
nucleon, 17
null-player, 11
occupied node, 65
overlapping decomposition, 31
pairwise consistent division rule, 42
parent, 74
Pareto optimality, 10
partition function form games, 21
passage, 65
payoff vector, 3
p-branch, 74
perturbation, 73
player network, 65
preimputation set, 9
prenucleolus, 15
profit game, 19
proper coalitions, 9
proportional rule, 53
proportional solution, 39
random arrival rule, 41
rationing problem, 40
reduced game, 12
reduced game property, 12
redundant constraints, 27
refined inequality, 37
rerouting, 91
residency mapping, 65
residing player, 65
reverse Talmud rule, 43
rights-egalitarian solution, 43
road construction algorithm, 102
simplified version, 96
root, 64
satisfaction
    value, 9
    vector, 14
saturated coalitions, 33
self-consistency, 42
self-dual rule, 41
semi-arc-disjoint, 77
Shapley-value, 14
shortcut, 73
Sobolev-criterion, 26
solution, 3
standard allocation, 76
standard representation, 76
standard tree game, 61
Steiner arborescence problem, 67
Steiner tree problem, 5, 62
strategically equivalent games, 8
strong monotonicity, 11
Strongly essential coalitions, 36
strongly superadditive game, 8
subadditive game, 19
superadditive game, 3
Talmud, 39
Talmud-rule, 40
time complexity, 4
$T_N$-arc, 73
transferable utility game, 3
tree enterprise, 61
trunk, 65
unoccupied node, 65
vessel, 44
vital coalitions, 36
Index

voting game, 25
water level, 45
weakly majorizing
  overlapping decomposition, 32
  partition, 30
weakly minorizing
  overlapping decomposition, 32
  partition, 30
weakly superadditive game, 8
width of a DAG-network, 95
worker, 97
zero arc, 65
zero-normalized game, 8