Shortest Path Games:
Computational Complexity of Solution Concepts

MSc Thesis (*Afstudeerscriptie*)

written by

Frank Nebel
(born December 22nd, 1982 in Aschaffenburg, Germany)

under the supervision of **Dr Ulle Endriss**, and submitted to the Board of Examiners in partial fulfillment of the requirements for the degree of

MSc in Logic

at the *Universiteit van Amsterdam*.

Date of the public defense:	**Members of the Thesis Committee:**
February 5th, 2010	Dr Stéphane Airiau
	Prof Dr Krzysztof R. Apt
	Prof Dr Peter van Emde Boas
	Dr Ulle Endriss

Institute for Logic, Language and Computation

Contents

Abstract

Over the last few years a series of papers has been published that analyse the computational complexity of solution concepts applied to different types of coalitional games, which are expressed by more or less concise representation languages. However, the coalitional games that have been analysed in a computational context typically have fixed and restricted characteristics and were studied in isolation from each other. For instance, results for related classes of games, like graph-based games, have not been systematically compared with respect to computational complexity or expressive power.

If we are exclusively interested in a specific type of game, this is certainly adequate, whereas a one by one analysis of complexity theoretic problems is impractical when we want to analyse a wide range of more or less distantly related coalitional games. To tackle this issue, we want to motivate in this thesis a more abstract approach, which is based on the characteristics of related types of coalitional games. For this reason, we have chosen an interesting graph-based coalitional game, namely shortest path game, to demonstrate the proposed approach on a sample game.

In particular, we study the computational complexity of solution concepts applied to different variants of shortest path games, as well as the expressive power of those variants. Based on these results, we then analyse the influence of different characteristics of shortest path games with respect to both aspects. Furthermore, we conduct a case study, where we relate our results on shortest path games to known results on different types of graph-based coalitional games.

But apart from having an interesting sample game, we want to stress that shortest path games are worthwhile to consider for their own sake as well.

Chapter 1

Introduction

In this thesis we study computational issues of graph-based coalitional games: On the one hand we analyse the influence of different characteristics on complexity-theoretic problems and on the other hand we consider complexity-theoretic problems over different types of graph-based coalitional games. At first, we want to give an overview of the research area our thesis is part of, and then we motivate our work conducted in this area.

1.1 Area of Research

The research area is an interdisciplinary area that offers new insights in problems arising in economics, social science, etc., by transferring these problems into coalitional games and by analysing these games in a game-theoretic environment. A survey of various coalitional games, which correspond to interesting problems in operations research, is given in Borm, Hamers and Hendrickx [13].

Having transferred a problem into a coalitional game, a manifold of results and methods of cooperative game theory can be used to solve the problem. The basic idea is that by applying well-known solution concepts to coalitional games, new or more efficient means to solve the original problem can be found. Here, we are especially interested in problems regarding the distribution of costs or gains. An important question that arises in this context and that attracted quite some attention in the last 15 years is the following: Is it feasible to apply this problem solving strategy (procedure) to analyse a particular problem in a game-theoretic context? To answer this question appropriately, it is necessary to analyse the computational complexity of solution concepts with respect to the particular coalitional game that corresponds to the given problem. Thus, we are interested if the application of solution concepts is efficient for a particular type of game.

We would like to mention some publications in this area [49, 24, 43, 44, 31, 18, 40, 10, 27, 1, 4, 7, 2, 6, 28] (ordered chronologically) that analyse the computational complexity of solution concepts applied to different types of coalitional games, which are expressed by different representation languages. The nature of our work shows similarities in results and organization to the work of Bachrach and Rosenschein [6, 7], Aziz [1] and Aziz, Lachish, Paterson and Savani [3]. In Figure 1.1, we sketched the different fields of research that are involved, like economics, social choice, etc., and their interrelations to give an overview of the area.

On the right-hand side of this figure, we can see different kinds of coalitional games, which originate from problems in economics and social choice. Looking at the bottom we have the machinery of

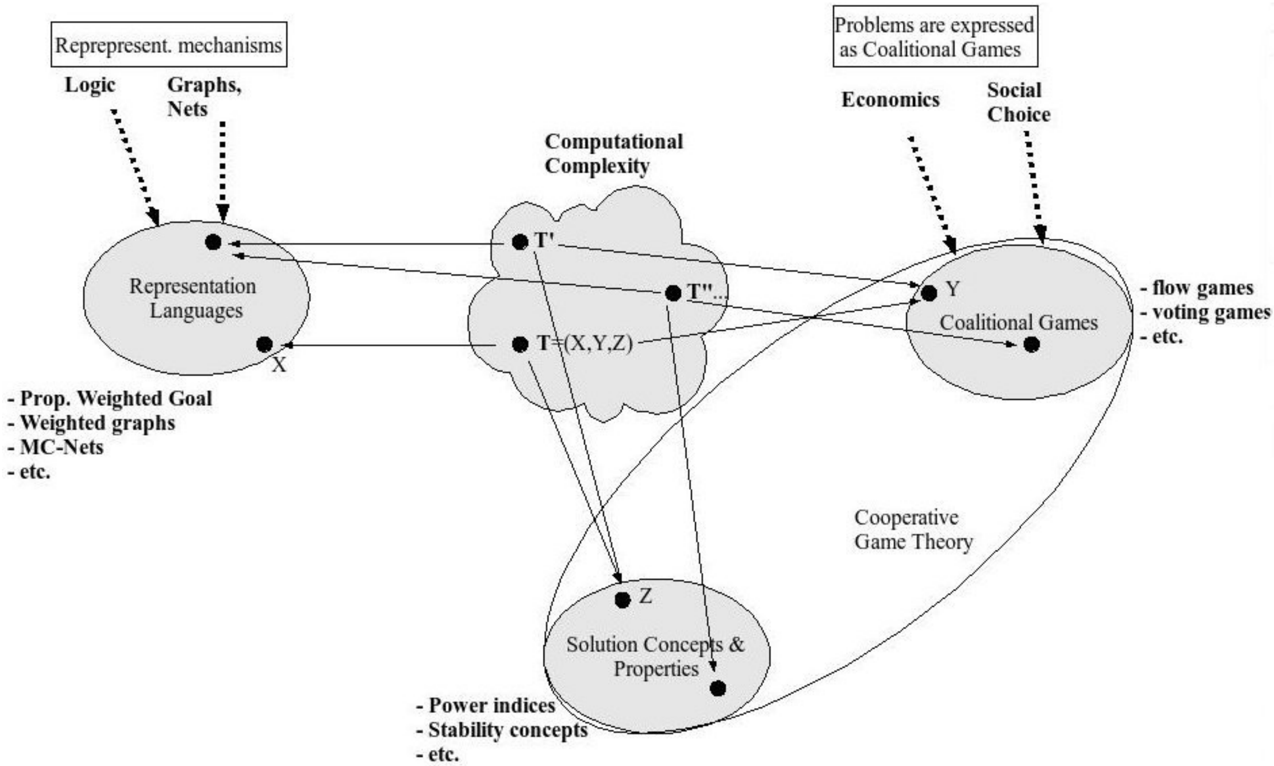

Figure 1.1: Overview of Research Area

cooperative game theory, namely solution concepts and properties that are used to analyse given coalitional games. Another important part involved in this area of research are the representation languages, which are used to represent coalitional games (left-hand side).

The central elements, represented by points in the stylized cloud, are the complexity results, which depend basically on three aspects, namely which class of coalitional games is involved, which kind of representation language is chosen and which solution concept is applied. Given their nature, they might be referred to as “triplets” (G, S, R), where G is a particular class of coalitional games, S is a solution concept and R is the representation language used. Note that we do not actually work with fully-expressive representation languages, but with particular game representations of coalitional games of a specific type. Nevertheless, it can be interesting in some situation to consider different types of game representations for particular classes of games, especially when these game representations express precisely the same class of coalitional games.

For our work, we will rely on different types of coalitional games, which have already been introduced in the literature to solve particular types of problems in operations research. We analyse these games with respect to expressive power and computational complexity of solution concepts, and interpret the results.

1.2 Motivation and Goals

Translating economical problems into coalitional games and analysing them with the machinery of cooperative game theory is a promising problem solving strategy, but we have to consider, as mentioned above, issues of complexity to actually decide if it is feasible. For many problems, like flow problems or weighted voting systems the corresponding coalitional games are such that important solution concepts are often intractable (see for example [6, 1]). This suggests the following questions:

- Is it the case that most solution concepts for coalitional games are actually intractable?
- What influence do characteristics of coalitional games have on the expressive power of games and on the complexity-theoretic results of solution concepts?
- Are there any heuristics or indicators how characteristics influence the computational complexity?
- Is there a tendency towards tractable or intractable results for interesting sub-classes of games, like graph-based or voting-based coalitional games?

These questions are quite general and much more research has to be done before these questions can be answered in a satisfactory way. So, as a first step, we decided to concentrate our research on well-defined and restricted types of coalitional games to show that these questions are interesting and worthwhile to consider, at least for restricted types of coalitional games. Here, in this thesis we are for example interested in what influence different characteristics of coalitional games might have on the expressive power of games and computational complexity of solution concepts applied to these games. Due to the fact that different coalitional games have often quite different representations and therefore different characteristics, we focused our research on a specific type of coalitional games, namely graph-based games. The reasons why we selected this type of coalitional games are the following:

- Graph-based games are particularly interesting to solve network-based problems of all sorts, which often occur in operations research.
- They offer a manifold of characteristics that can be analysed.
- Several graph-based games have already been treated in the literature over the last two years [24, 3, 6, 7]. This allows us to compare complexity-theoretic results over different graph-based games.

In our studies we observed that most graph-based representations of coalitional games, which can be found in the literature, are quite restricted games. So, we were curious to analyse which effect the consideration of different characteristics of graph-based games would have. Given that there are a lot of characteristics for graph-based games in general, we had to focus on a particular type of graph-based game, which has to be analysed exhaustively by considering all variations of interesting characteristics for this type of game. At this point, it is therefore necessary to be systematic, because only by an extensive analysis of all involved factors of interest, we are able to give a meaningful interpretation of the final results. For this reason we looked for a promising type of graph-based game to show how different characteristics influence the computational complexity of solution concepts.

Our final choice were shortest path games. We had several reasons to select this particular type of coalitional game:

- In the literature [33, 58], there are two different definitions of shortest path games, which have quite different properties (e.g. expressive power). We think that this is a good indication that shortest path games are a promising candidate to analyse the effect of different characteristics of graph-based games on complexity-theoretic results. Based on this analysis, we hope to find some hints about the influence of characteristics on graph-based games in general.
- The second reason why we have chosen shortest path games is that they are similar to various graph-based games that have already been treated in the literature [6, 3, 45, 36, 11, 29]. Hence, our intention was to extend the corpus of complexity-theoretic results for graph-based games, which allows us to discuss if there is a difference between graph-based games or if they are similar when it comes to the complexity of solution concepts.

Having selected shortest path games to be analysed, our agenda and motivation in this thesis is twofold:

- We want to determine the computational complexity of various solution concepts with respect to different variants of shortest path games. Based on these results, we want to analyse how different characteristics of shortest path games influence the computational complexity of solution concepts. For this reason, we fix a particular solution concept and vary over systematically defined variants of shortest path games, which have different configurations of characteristics. We are especially interested in characteristics of shortest path games, which reduce intractable problems to tractable ones. Furthermore, we want to determine the expressive power of various variants of shortest path games and compare them with other coalitional games.
- Given the results for shortest path games and results from the literature, we want to see how flow games, shortest path games, minimum cost spanning tree games, etc., graph-based games that share basic characteristics, are related to each other with respect to the computational complexity of the corresponding triplets. We hope that this might give us some insight about complexity problems for graph-based coalitional games in general, and maybe even heuristics to estimate the computational complexity of solutions concepts applied to graph-based coalitional games.

1.3 Thesis Overview

Due to the interdisciplinary character of our work, we intended this thesis to be self-contained to keep it accessible for researchers from different areas, who are interested in our work. The rest of this thesis is organized as follows:

In Chapter 2 we introduce the concept of a coalitional game, as well as basic definitions, properties and theorems in cooperative game theory that will be used in later chapters. Then we give an overview of solution concepts for coalitional games, which are divide in two groups: the power indices and the stability concepts. Due to the central role of computational complexity in our work, we give a short account of computational complexity in general and more specifically with respect to coalitional games. The last issue of this chapter is the representation of coalitional games and the importance of game representations for complexity-theoretic considerations.

In the next chapter we present several coalitional games, namely shortest path games, flow games, minimum cost spanning tree games, vertex connectivity games, spanning connectivity games, linear production games and market games. In later chapters, when we prove properties and complexity results for different coalitional games, we will often refer to models defined in this chapter. Having introduced the models, we determine characteristics common to various types of graph-based coalitional games, but especially shortest path games and discuss their influence with respect to expressive power and computational complexity. We preselect a set of characteristics, which we would like to consider for shortest path games and introduce systematically several variants of shortest path games, which are based on the set of preselected characteristics.

In the following chapter we determine some basic properties of coalitional games, which we defined in Chapter 3. Then we analyse how expressive the different types of coalitional games are and how they relate to each other with respect to expressive power.

In Chapter 5 we analyse the computational complexity of solution concepts applied to different variants of shortest path games, as well as the computational complexity to determine player-based properties for these games. Our focus is on power indices, but we also consider an important stability concept in our analysis. Then we present some complexity results for various graph-based coalitional games from the literature.

In the next chapter we discuss the results of the previous chapter: We summarize and interpret on one hand the results for the different variants of shortest path games and discuss the complexity-theoretic influence of different characteristics, and on the other hand, we conduct a case study of various graph-based coalitional games, which all share basic characteristics. In this case study we compare the various results and discuss possible complexity-theoretic implications for graph-based coalitional games in general.

The last chapter concludes by summarising our main results and discussing possible directions for future work.

Chapter 2

Preliminaries

In this chapter we give an overview of various concepts and notions of different areas of mathematics, computer science and social choice. We start by introducing the predominant notion in this thesis, namely the notion of a coalitional games and we furthermore present some basic notions of cooperative game theory. Then we introduce some definitions, properties and theorems in cooperative game theory that will be important for our line of work. This is followed by a detailed account of an essential part of cooperative game theory, namely the study of solution concepts. Here, we will distinguish between two types of solution concepts: power indices and stability concepts. Due to the fact that a main part of our work is about the computational complexity of solution concepts applied to coalitional games, we give a condensed introduction of the area of computational complexity. Finally, we conclude this chapter by giving a short account of different representations for games and the connection between game representations and computational complexity.

2.1 Coalitional Games

We now introduce basic notions of cooperative game theory (also sometimes referred to as coalitional game theory). As the term "coalitional" already indicates, the main modeling unit of a coalitional game is the group of players involved. So, given a set of players N, a coalitional game defines, expressed informally, how well each coalition $S \subseteq N$ (or group) of players can do for itself. Contrary to traditional game theory, it is not of concern how players make individual choices within a coalition, how they coordinate, and what are their individual outcomes.

Stated quite general, cooperative game theory is used basically to answer two questions, namely which coalitions will form and how should such a coalition divide its payoff among its players. The answer to the first question is often "the grand coalition" (including all players of a game), whereas this also depends on the right choice for the second question.

We have two main classes of coalition games, with and without transferable utility. Our focus will be on coalitional games with transferable utility, where the payoffs to a coalition may be freely redistributed among its members. This is, of course, only possible if there is a universal currency that is used for exchange in the system. So, as a result of this setting each coalition can be assigned a single value as its payoff. Another important notion, apart from transferable utility, is the notion of side payments. In this thesis we concentrate on games with *side payments (SP)*. We will give a short introduction how both concepts are related and what role do they play. To see the effect of the "side payment" property we give a short overview of the distinct behaviours. In the case of a coalitional game with side payments we can express a possible outcome of a coalition by one real number, the total payoff achievable by the coalition. So, utilities of players in the outcome of a game can be summarized. In contrast, a game without side payments states the payoff to each player in the coalition separately in the outcome, without summarizing it. Games with side payments require special consideration of the underlying utility theory, namely the assumption of "transferable utility". Having both, SP and TU, this implies that utility appears to be transferred at a one to one rate between players. More information about utility theories in coalitional games can be found in [9, 42].

When we speak of coalitional games in this thesis we always implicitly assume that they have both properties: transferable utility and side payments. A coalitional game is given by specifying a value for every coalition, or more formally:

Definition 2.1.1. *A* coalitional game *with transferable utility (TU-game) is a pair $\langle N, v \rangle$, where*

- *N is the set of players; and*
- *$v : 2^N \rightarrow \mathbb{R}_0^+$ is a function (characteristic function) that maps each group of players $S \subseteq N$ to a positive real-valued payoff.*

Due to the shape of function v, coalitional games, which are represented in this way are in *characteristic function form* and are often called *characteristic function games*. We often use the simple term "game" or "coalitional game", when we want to refer to coalitional game with transferable utility as defined above. For the sake of brevity, we will abuse the notation to sometimes refer to game $\langle N, v \rangle$ as v. An *outcome* in a coalitional game specifies the payoffs (utilities) the players receive and a *solution concept* assigns to each coalitional game a set of "reasonable" outcomes. We will give a more detailed account of solution concepts later in this section. A thorough and complete introduction to cooperative game theory can be found in [20, 26, 48].

2.2 Basic Definitions, Properties and Theorems

We start to introduce some properties of coalitional games.

Definition 2.2.1. *A game v is* monotonic *if for all $S, T \subseteq N$, $S \subseteq T$ implies $v(S) \leq v(T)$.*

A fundamental type of a coalitional game, considered in cooperative game theory, is a simple game.

Definition 2.2.2. *A coalitional game $\langle N, v \rangle$ is called a* simple game[1] *if $v(S) \in \{0, 1\}$ for all $S \subseteq N$ and $v(\emptyset) = 0$, $v(N) = 1$.*

Simple games are well suited to model situations of voting and committee control. Especially monotonic simple games, often called *(simple) voting games* are interesting in this context, because the definition already captures the basic building blocks for most voting systems.

Definition 2.2.3. *A coalition $S \subseteq N$ in a simple game is called* winning *if $v(S) = 1$, and* losing *if $v(S) = 0$.*

[1] In the literature simple games are sometimes defined as being monotonic as well.

Now we want to introduce some player-based properties of coalitional games.

A player in a coalitional game is called a null player[2], if he or she contributes nothing to any coalition.

Definition 2.2.4. *A* null player *is a player $i \in N$ such that for all coalitions $S \subseteq N \setminus \{i\}$ we have $v(S \cup \{i\}) = v(S)$.*

A dummy player is defined as follows:

Definition 2.2.5. *A* dummy player *is a player $i \in N$ such that for all coalitions $S \subseteq N \setminus \{i\}$ we have $v(S \cup \{i\}) = v(S) + v(\{i\})$.*

A player is a veto player in a simple coalitional game if no coalition can win without the involvement of this player.

Definition 2.2.6. *A* veto player *in a simple game is a player $i \in N$ such that for all coalitions $S \subseteq N \setminus \{i\}$ we have $v(S) = 0$.*

A similar, but more general definition of a veto player can also be given for non-simple coalitional games. In a non-simple game, a player $i \in N$ is a veto player if no coalition can achieve any gains without the involvement of player i.

Definition 2.2.7. *A player $i \in N$ in a simple game $\langle N, v\rangle$ is* critical *in a winning coalition $i \in S \subseteq N$ if the player's removal from that coalition would make the coalition lose: $v(S) = 1$ but $v(S \setminus \{i\}) = 0$. A pair of coalitions $(S \cup \{i\}, S)$ is called a* swing *for player $i \in N$ in a simple game $\langle N, v\rangle$, if $S \cup \{i\}$ is winning and $S \subseteq N \setminus \{i\}$ is losing*

Definition 2.2.8. *A coalition S in a simple game is called a* minimal winning coalition *if $v(S) = 1$ and for every $T \subset S$, $v(T) = 0$.*

The concept of a minimal winning coalition is interesting, because it represents a very important kind of coalition, a coalition with maximal power, but minimal effort. Hence, a limit in some sense.

We can generalize this concept to minimal profitable coalitions:

Definition 2.2.9. *A coalition S in a coalitional game is called a* minimal profitable coalition *if $v(S) > 0$ and for every $T \subset S$ $v(T) = 0$.*

Balancedness and Totally-Balancedness

We now introduce the concept of balancedness, a basic and significant concept in the theory of coalitional games and directly connected to an important solution concept, namely the core.

Given a coalition S, we define the vector $1_S \in \mathbb{R}^{|N|}$ by

$$(1_S)_i = \begin{cases} 1 & \text{if } i \in S \\ 0 & \text{otherwise} \end{cases}$$

Definition 2.2.10. *A collection of numbers between zero and one (weights for each coalition), $(\lambda_S)_{S \in 2^N}$ is a* balanced collection of weights *if we have $\sum_{S \in 2^N} \lambda_S * 1_S = 1_N$*

A TU-game $\langle N, v\rangle$ is balanced *if for each balanced collection of weights, we have $\sum_{S \in 2^N} \lambda_S * v(S) \leq v(N)$.*

[2] In some publication a null player is referred to as a dummy player.

The intuition behind balancedness is the following: Players can distribute one unit of working time to any coalition. But by doing this, they cannot generate more value than the grand coalition (see [19]).

Another, but more indirect insight can be gained by considering the core of a coalitional game, which is strongly related to the balancedness property. The core of a game is a significant solution concept in cooperative game theory and will be presented later in this thesis (see Section 2.3.2). Now to the connection of both concepts: Bondareva [12] and Shapley [52] proved independently the following result, which is now called the Bondareva-Shapley theorem.

Theorem 2.2.11. *A game has a non-empty core iff it is balanced .*

For any $S \subseteq N$ let $v|S$ denote the induced subgame v restricted to S. Based on this notion of a subgame, we can define the property of totally balancedness:

Definition 2.2.12. *A coalitional game is* totally balanced *if for all $S \subseteq N$ $v|S$ is balanced, i.e. all its subgames have non-empty cores.*

Proposition 2.2.13. *There are monotonic games that are not balanced.*

An example is given in the proof of Proposition 4.2.2, where we show that a particular monotonic game is not balanced.

Then we have the following well known result, which relates the emptiness of the core with the concept of a veto player for simple games.

Theorem 2.2.14. *A simple game v is balanced (has a non-empty core) iff v has veto players. ([20], Theorem 1.10.6)*

2.3 Solution Concepts for Coalitional Games

In this section we introduce notions and concepts to talk about "reasonable" outcomes in coalitional games. In general, each solution concept defines for each coalitional game $\langle N, v\rangle$ a set $\mathcal{F}$ of allocations. Expressed more formally, we have the following definition for solution concepts:

Definition 2.3.1. *A function $\Psi : 2^{\mathbb{N}} \times \mathbb{R}^{2^{|N|}} \rightarrow \mathcal{P}(\mathbb{R}^{|N|})$ which assigns to every $|N|$-person coalitional game $\langle N, v\rangle$ a possibly empty subset $\Psi(N, v)$ of $\mathbb{R}^{|N|}$ is called a* solution concept.

Each element of the set $\Psi(N, v)$ is a vector of payoffs, which are assigned to each player. So, each vector of $\Psi(N, v)$ specifies a different way how to divide the payoff (of the grand coalition) between the players. We call a function x from N to $\mathbb{R}$ (written as $x \in \mathbb{R}^{|N|}$) an *allocation* and denote its i-th component by $x(i)$ or x_i. Given a coalition S, we abbreviate $\sum_{i \in S} x(i)$ by $x(S)$. A solution concept Ψ is called a *one-point solution concept*[3] if $|\Psi(N, v)| = 1$ for every game $\langle N, v\rangle$. We introduce the following notion for a one-point solution concept: For $i \in N$, $\Psi_i(N, v)$ is i's payoff.

In cooperative game theory there are different solution concepts, which capture outcomes that are in a way stable or fair. A more complete account of solution concepts in coalitional games can be found in [48].

Before we start to introduce examples of popular solution concepts we would like to give an overview of properties, which are applicable to solution concepts and present different types of payoff allocations. Some common properties are:

[3]A one-point solution concept is often referred to as a solution function (or value function).

- **Feasibility:** $x(N) \leq v(N)$,
- **Efficiency:** $x(N) = v(N)$,
- **Individual rationality:** for all $i \in N$ $x_i \geq v(i)$,
- **Group rationality:** for all $S \subseteq N$ $x(S) \geq v(S)$,
- **Existence:** for any game, the solution concept exists.
- **Uniqueness:** When we define a solution concept on a set of games, we might want to know if this solution is unique under special conditions. The Shapley-Shubik index is an example for a unique solution concept.
- **Computational ease:** The solution concept can be calculated efficiently.

We continue by presenting important types of payoff allocations:

We know that an allocation assigns to each player some payoff. In what follows next, we discuss the meaningful types of allocations in the context of coalitional games $\langle N, v \rangle$. To do this on an objective ground, we focus on several properties of an allocation x, which have been mentioned above, and which link the allocations with the characteristic function v.

We call an allocation a *pre-imputation* if it is efficient (thus feasible as well). A pre-imputation is an *imputation* if it also satisfies the individual rationality property. So, an imputation is a distribution of the payoff which the grand coalition can achieve (efficiently) such that no player has an individual interest in rejecting it (individual rationality). Imputations play an important role, because most solution concepts are actually imputations or at least pre-imputations.

If we consider how to distribute the payoff of the grand coalition, it is clearly the case that only imputations should be considered as meaningful. As the reader may have noticed, we focus heavily on the notion of a grand coalition for our definitions. This is the reason, because many coalitional games are superadditive, additive or monotonic and therefore the grand coalition achieves the highest payoff for these games. In this thesis for example, we will focus exclusively on monotonic games.

Given a coalitional game, the goal is typically to find a "fair" imputation, i.e., the share each player receives is proportional to his or her contribution, or "stable", in the sense that it provides as little or no incentive for a group of players to abandon the grand coalition and form a coalition of their own. There is a manifold of different solution concepts, which formalize notions of fairness and stability. Now we present several of these solution concepts.

Solution concepts are divided in two main classes, namely the power indices and the stability concepts. We will start with power indices, which play a major role in this thesis.

2.3.1 Power Indices

Power indices originally arose in the context of voting and as the name already indicates, they were used to determine the power of a party in a parliament given the distribution of votes over the parliament. In general, each player in a voting game will have a certain number of votes. So, approaching this problem naively, the power of players might seem predetermined from the start. But this is not necessarily true, why different power indices has been introduced.

In principal, a power index allows a more detailed view on power distributions in voting situations and give insights that are not obvious on the surface. Despite their origin, power indices are not restricted to voting games, but can also be used to determine the power of players for many other types of monotonic coalitional games.

We introduce the following power indices: Shapley-Shubik index, Banzhaf index, Deegan and Packel index and the Public Good index. Note that the Shapley-Shubik and Banzhaf power indices are defined on monotonic games, whereas the Deegan-Packel index and the Public Good index are defined for monotonic simple games only.

Shapley-Shubik Power Index

Lloyd Shapley and Martin Shubik introduced the Shapley-Shubik index [53] in 1954 to measure the powers of players (parties) in a voting game (voting system). The original Shapley-Shubik index has been defined for monotonic simple games (voting games), but can also be generalized for monotonic coalitional games $\langle N, v\rangle$.

Definition 2.3.2. *Given a monotonic coalitional game $\langle N, v\rangle$, the* Shapley-Shubik-Index *of player i is given by $\varphi_i(N, v) = \frac{\kappa_i}{|N|!}$, where the* Shapley-Shubik value *κ_i is equal to* $\sum_{S\subseteq N\setminus\{i\}} |S|!(|N| - |S| - 1)![v(S \cup \{i\}) - v(S)]$.

We now give an interpretation of this formula: A coalition is assembled by starting with an empty coalition and adding one player at a time. The players to add are chosen uniformly at random.

For any such sequence of inclusions of players, we consider player i's marginal contribution at the time the player is added to the coalition. If the player $i \in N$ is added to the coalition S, his or her contribution is $v(S \cup \{i\}) - v(S)$. Now we have to multiply this quantity by the $|S|!$ different ways the set S could have been formed prior to player i's addition and by the $(|N| - |S| - 1)!$ different ways the remaining players could be chosen and added afterwards. Then we finally sum over all possible coalitions S, thus the corresponding marginal contributions, and divide it by $|N|!$. Note that by dividing by $|N|!$ we basically average over all the sequences by which the grand coalition can be build up from the empty coalition. This process basically captures the *average marginal contribution* of player i.

The Shapley-Shubik index is strongly based on marginal contributions of coalitions, a notion that is applicable and meaningful for any kind of monotonic coalitional games and therefore the generalization of the Shapley-Shubik index introduced above is indeed reasonable.

Banzhaf Power Index

The Banzhaf power index was originally defined by John Banzhaf [8] in the context of weighted voting games. The intention was to measure the political power of each member of a voting system. The Banzhaf index can be defined, similarly to the Shapley-Shubik index, for monotonic coalitional games. Expressed informally, the index is easily derived by counting for each player the number of winning coalitions it participates in, but which are not winning if the player is not participating anymore.

Definition 2.3.3. *Given a monotonic coalitional game $\langle N, v\rangle$, the* Raw-Banzhaf-Index *(also called* Banzhaf value*) $\eta_i(v)$ of player i is the number of swings of player i ($\eta_i(v) = \#\{S \subseteq N \setminus \{i\} \mid v(S \cup \{i\}) - v(S) = 1\} = \frac{1}{2^{n-1}} \sum_{S\subseteq N\setminus\{i\}}[v(S \cup \{i\}) - v(S)]$). If we assign probability $\frac{1}{2^{n-1}}$ to each coalition $S \subseteq N \setminus \{i\}$, the swing probability*

$\beta_i(N, v) = \frac{\eta_i(v)}{2^{n-1}}$

is called the Banzhaf index *of player i and denoted by β_i.*

Comparing both indices, Shapley-Shubik and Banzhaf, we can immediately see that both indices reflect the player‘s expected marginal contribution to the value of a coalition. However, the underlying probabilistic model of coalition formations is different. Instead of assuming that players join the coalition at random order and thus all permutations of the agents are equally likely, the Banzhaf index assigns equal probability to all 2^n possible coalitions.

Note that the Shapley-Shubik index and the Banzhaf index are normalized versions of the Shapley-Shubik value and Banzhaf value respectively.

Deegan-Packel Power Index

A refinement of the Banzhaf index was introduced by Deegan and Packel [22]. The reason for this refinement is the following: The authors claimed that only minimal winning coalitions should be considered to establish the power of a voter, and they based this claim on the assumption that rational players want to maximize power and therefore only minimal winning coalitions are reasonable. To incorporate these ideas into the power index, they assumed that all minimal winning coalitions are equiprobable and all the voters belonging to the same minimal winning coalition should basically obtain the same power. Hence, the Deegan-Packel power index gives a measure of power that satisfies certain conditions:

- Minimality: Only minimal winning coalitions will be victorious,
- Equiprobability: Each minimal winning coalition has an equal probability of forming, and
- Solidarity: Players in a minimal winning coalition divide the power equally

In many cases these conditions are reasonable. Expressed formally, the index is defined as follows:

Definition 2.3.4. *The* Deegan-Packel power index *of a player i in the simple game $\langle N, v\rangle$ is give by:*
$p_i(N, v) = \frac{1}{|M(v)|}\sum_{S\in M_i(v)}\frac{1}{|S|}$
where we denote by $M(v)$ the set of minimal winning coalitions of v and by $M_i(v)$ the subset of $M(v)$ formed by coalitions $S \subseteq N$ such that $i \in S$.

Note that contrary to the Shapley-Shubik and Banzhaf index, the Deegan-Packel index cannot be extended to arbitrary coalitional games (see [38]).

With the Deegan-Packel index, players should look for minimal winning coalitions that are minimal in the cardinality of the coalition. A different index based on minimal winning coalitions is the Public Good index. For this power index, the size of the minimal winning coalition does not matter to measure power.

Public Good Power Index (PGI)

Manfred Holler [37] proposed the Public Good Power index (PGI). The PGI for player $i \in N$ is basically determined by the number of minimal winning coalitions (MWC) containing player i divided by the sum of such numbers across all the players. It is assumed that coalitions that do not have the minimal winning property do not matter and thus should not be taken into consideration when it comes to measuring power.

Definition 2.3.5. *Given a simple game $\langle N, v\rangle$, the PGI assigns to each player $i \in N$ the real number: $\delta_i(N, v) = \frac{|M_i(v)|}{\sum_{j\in N}|M_j(v)|}$, where we denote by $M_i(v)$ the set of minimal winning coalitions that include player $i \in N$ of game $\langle N, v\rangle$.*

As stated by Holler in [39], the PGI index is only defined for monotonic simple games. Note that although only MWC are taken into account for calculating the PGI, it is not claimed that no other coalitions will be formed.

General Properties of Power Indices

We prove now a direct correspondence between the null-player property of coalitional games and the different power indices. For the Shapley-Shubik and Banzhaf index this result follows immediately, but we would like to check, if it holds for the other two indices as well. So, for completeness we will give a proof for all four indices.

Proposition 2.3.6. *For the Shapley-Shubik-index, the Banzhaf index, the Deegan-Packel index and the Public Good Index: For all monotonic coalitional games $\langle N, v\rangle$ (simple monotonic games in the case of the Deegan-Packel and Public Good Index) player i is a null-player in v iff the corresponding index of player i for game v is equal to* 0.

Proof. We prove both directions separately:

$\Rightarrow$: Let i be a null-player in game v , i.e. $v(S \cup \{i\}) = v(S)$ Shapley-Shubik and Banzhaf index, it follows immediately that $\varphi_i(v) = \beta_i(v) = 0$. This is, because there are no marginal contributions for player i. We can also verify immediately that $M_i(v) = \emptyset$ and therefore $p_i(v) = 0$ and $\delta_i(v) = 0$.

$\Leftarrow$: Let the index for game v and player i be 0. We assume for the sake of a contradiction that player $i \in N$ is not a null-player. Then, there is a coalition $S \subseteq N$ such that $v(S \cup \{i\}) \neq v(S)$. Given that v is a monotonic game, we have that $v(S \cup \{i\}) > 0$ and $v(S) = 0$. Hence, $v(S \cup \{i\}) - v(S) \geq 1$ and therefore $\varphi_i(v) \neq 0$ and $\beta_i(v) \neq 0$. This means we reached a contradiction for the case of the Shapley-Shubik and Banzhaf index.

For the other two indices, we know that $T = S\cup\{i\}$ is a winning coalition and S is a losing coalition, but we do not know if it is a minimal winning coalition. But it can easily be shown that there is a minimal winning coalition $T' \subseteq T$ such that $i \in T'$:

We just iterate through all players j in S and check if $v(T) = 1$ and $v(T \setminus \{j\}) = 0$. If this is the case, we continue with the next player in the iteration. Otherwise, If this is not the case, then $v(T) = 1$ still holds, but $v(T \setminus \{j\})$ is equal to 1 instead of 0. So, we remove j from T and continue with the next player in the iteration.

After this process is terminated, we have a new coalition $T' \subseteq T$ such that for all players $a \in T'$, $v(T') = 1$ and $v(T' \setminus \{a\}) = 0$. So, given that v is monotonic, it is clearly the case that T' is a minimal winning coalition. So, due to the fact that $i \in T'$ we know that both $M(v)$ and $M_i(v)$ are non-empty. Thus, $p_i(v) \neq 0$ and $\delta_i(v) \neq 0$ and we reached a contradiction for the Deegan-Packel and PGI index as well. □

2.3.2 Stability Solution Concepts

The different power indices that we introduced above, define a “fair” way of how to divide the grand coalition’s payoff among the players. However, the power indices often ignore questions of stability, and therefore "oversee" the possibility that sometimes smaller coalitions can be more attractive for subsets of players, even if they lead to a lower overall value.

In the context of stability, we want to know if the players would be willing to form the grand coalition given the way the payments are divided, or would some players prefer smaller coalitions instead. An example of a situation where smaller coalitions can be more attractive for subsets of players is the following: Let us consider a voting game where an amount of money should be divided between four parties A, B, C and D. To implement a distribution 51% of the votes are necessary. So, if A and C have 40 and 20, then they could split the amount of money $66,7 - 33,3$. This is clearly better than a grand coalition where B and D would also receive their parts.

The concept of a *stable set* of a game (also know as the von Neumann-Morgenstern solution) was the first stability concept proposed for games with more than 2 players (see [57]). To give the reader a flavour of what stability concepts are, we present some more details. Before we continue, we first want to introduce the notion of domination between two imputations of a game. We say that an imputation x is *dominated* by an imputation y, written $y \succ x$, if for some coalition S.

- $v(S) \geq y(S)$,
- $y(i) > x(i)$ for all $i \in S$

Intuitively, the first condition states that the imputation y offers to the coalition S a payoff that can actually be achieved, while the second one states that y offers to each member of S strictly more than the imputation x. So, for coalition S the imputation y is then more attractive than the imputation x.

Definition 2.3.7. *A subset Y of the set X of imputations of a coalitional game with transferable payoff $\langle N, v\rangle$ is a* stable set *if it satisfies the following two conditions:*

- Internal stability: *No payoff vector in Y is dominated by another vector in Y.*
- External stability: *All payoff vectors outside of Y are dominated by at least one vector in the set.*

The advantage of this general concept is that it can be used in a wide variety of coalitional games. An important property of stable sets is: A stable set may not exists, and if it exists it is typically not unique.

The problem of stable sets is that they are difficult to find. This and other difficulties have led to the development of many new solution concepts:

- The core
- The least-core
- The strong epsilon-core
- The kernel
- The nucleolus

In this thesis, we concentrate on one of the most important stability concept, namely the core. The others concept are important stability concepts as well, but will not be used in what follows.

We introduced above the notion of an imputation, an efficient and individual rational allocation. However, an imputation does not need to be a "stable" allocation. This means that there might be some coalition of players, which would reject it if they could achieve a larger combined payoff by cooperating. This lack of stability immediately imposes a third property of an allocation x that strengthens the property of individual rationality:

- **group rationality:** $x(S) \geq v(S)$ for all coalitions S of N.

We can now define the fundamental concept of a cor*e* for a coalitional game:

Definition 2.3.8. *Let* $\langle N, v\rangle$ *be a cooperative game. Then the* core *of* v *is defined as:* $Core(v) := \{x \in \mathbb{R}^n \mid x \textit{ is an imputation that satisfies group rationality}\}$

Intuitively, an imputation lies in the core if it is a distribution of the payoff of the grand coalition such that no group of players is interested in rejecting it. Some properties of the core are:

- The core of a game may be empty.
- If the core is non-empty, it does not necessarily contain a unique vector.
- The core is contained in any stable set and if the the core is stable it is the unique stable set (see [26])

2.4 Computational Complexity

In Chapter 5 we are going to analyse the computational complexity to apply solution concepts on different types of coalitional games. Remember, a solution concept defines for each coalition game $\langle N, v\rangle$ a class $\mathcal{F}$ of imputations. When we now analyze solution concepts for particular coalitional games from a computational perspective, there are at least three natural computational problems that can be taken into account.

1. Decide whether a proposed allocation is "fair" according to $\mathcal{F}$: "Given an imputation x, does it belong to $\mathcal{F}$?"
2. Given the present coalition situation, are there any fair allocations at all? "Is $\mathcal{F}$ nonempty?".
3. Is it possible to find an element of the solution concept in polynomial time.

The consideration of computational complexity in the context of coalitional games is especially interesting, because the definition of a coalitional game involves an exponential number (with respect to the number of players of a game) of values, namely one for each coalition $S \subseteq N$. Furthermore, most of the solution concepts have a definition, which involves an exponential number of constraints or steps to compute them in a direct manner. This is the point, where the representation of coalitional games comes into play. We give a more detailed account of this topic in the next section, where we also answer the question why it makes sense to analyse the computational complexity for coalitional games nevertheless. Because of the importance of computational issues for solution concepts, Deng and Papadimitriou [24] suggested that computational complexity should be taken into account as another measure to compare different solution concepts.

An important aspect to consider is that coalitional games with transferable utility, as defined above, are characteristic functions v with codomain $\mathbb{R}_0^+$. So, when we analyse complexity-theoretic problems involving coalitional games, we assume for technical reasons that the codomain is $\mathbb{Q}_0^+$ instead.

We now present different types of problem statements, which we are going to encounter later in this work, namely decision, function and counting problems.

CORE-MEMBERSHIP (DECISION)

Instance: A game v and an imputation x

Question: Is $x \in Core(v)$?

SHAPLEY-SHUBIK INDEX (FUNCTION)

Instance: A game v, player $i \in N$

Solution: The Shapley-Shubik index for player i ($\varphi_i(v)$)

#BANZHAF VALUE (COUNTING)

Instance: A game v, player $i \in N$

Solution: Number of coalitions in the set $\{S \subseteq N \setminus \{i\} \mid v(S \cup \{i\}) - v(S) = 1\}$.

After presenting the different types of problem statements, we are going to introduce the basic concepts and notions of computational complexity. We concentrate on three classes of computational complexity, namely $\mathcal{P}$, $\mathcal{NP}$ and $\#\mathcal{P}$.

The two most fundamental complexity classes are:

- $\mathcal{P} = \bigcup_{k>1} \mathbf{TIME}(n^k)$: $\mathcal{P}$ is the class of problems that can be solved in polynomial time by a deterministic algorithm; and
- $\mathcal{NP} = \bigcup_{k>1} \mathbf{NTIME}(n^k)$: $\mathcal{NP}$ is the class of problems for which a proposed solution can be verified in polynomial time.

Apart from these two classes there is another powerful class of functions, $\#\mathcal{P}$ (pronounced "number P" or "sharp P"), which was introduced by Valiant [56]. This complexity class is defined in the following way:

Definition 2.4.1. *Let Q be a polynomially balanced, polynomial-time decidable binary relation. The* counting problem *associated with Q is the following: Given x, how many y are there such that $(x, y) \in Q$? The required output is an integer. $\#\mathcal{P}$ is the class of all counting problems associated with polynomial balanced polynomial decidable relations.*

Phrased slightly differently, the class can be defined as follows: $\#\mathcal{P}$ is the complexity class consisting of all the functions $f : \Sigma^* \rightarrow \mathbb{N}$ such that there exists a non-deterministic polynomial time Turing machine M and for all inputs $x \in \Sigma^*$, $f(x)$ is the number of accepting paths of M.

The main reason why we are not only interested in $\mathcal{NP}$, but also $\#\mathcal{P}$ is the following: To determine the number of solutions of a problem is at least as hard as determining if there is at least one solution. So, *$\#\mathcal{P}$-complete* problems are at least as hard (but possibly harder) than *$\mathcal{NP}$-complete* problems. This gives us a more fine grained distinction of the computational complexity of different solution concepts and therefore motivates the application of this rather unknown complexity class. Furthermore, it is quite common in this research area to use the complexity class $\#P$.

Finally, we would like to remind the reader of notions like hardness and completeness. We start by refreshing the reduction concept:

The notion of polynomial time reduction is defined in the following way: A problem P reduces to a problem P' if we can translate any instance of P into an instance of P'. This allows us to apply the problem solver for P' to obtain an answer to the original problem (question) of type P. If the translation process is polynomial, then we can state that problem P' is at least as hard as problem P. This is the case because a P'-solver can solve any instance of P, and possibly a lot more.

To formally prove that a problem P' is at least as hard as a problem P we have to execute two tasks:

1. We have to show how to translate any P-instance into a P'-instance in polynomial time, and then
2. Show that the answer to the P-instance should be *YES* iff a P'-solver answers with *YES* to the translated problem.

Based on these concepts, the notion of hardness and completeness can be formally introduced. Let $\mathcal{C}$ be a complexity class:

- A problem P is *$\mathcal{C}$-hard* iff any $P' \in \mathcal{C}$ is polynomial-time reducible to P. That means that *$\mathcal{C}$-hard* problems include the hardest problems inside the class $\mathcal{C}$ and even harder ones.
- A problem P is *$\mathcal{C}$-complete* iff P is *$\mathcal{C}$-hard* and $P \in \mathcal{C}$. That means that they are the hardest problems in $\mathcal{C}$, and exclusively those problems.

To sum it up, a standard way to prove that P is *$\mathcal{C}$-hard* proceeds as follows: We have to find a *$\mathcal{C}$-complete* problem P' that is reducible to P, i.e. any instance of P' can be transformed to P in polynomial time and the answers to both problems are corresponding. If this is the case, we can immediately deduce from the fact that P' is *$\mathcal{C}$-complete* and by transitivity of the reduction relation that any $P'' \in \mathcal{C}$ is polynomial time reducible to P. Thus P is *$\mathcal{C}$-hard* by definition.

An extensive list of *$\mathcal{NP}$-complete* problems can be found in Garey and Johnson [34]. A detailed and up to date account of computational complexity can be found in the standard textbook for complexity theory, which is written by Papadimitriou [47]. Finally, for an overview of complexity-theoretic issues in cooperative game theory, we would like to refer the reader to Bilbao, Fernandez and Lopez [10].

2.5 Representation of Games

In this section we give an overview of the application and significance of representations of coalitional games. We furthermore distinguish between the notion of a coalitional game and the representation of a coalitional game and introduce some useful notations.

We introduced N-player coalitional games with transferable utility (in Definition 2.1.1) as characteristic function games $\langle N, v \rangle$, where $v : 2^N \rightarrow \mathbb{R}_0^+$ assigns to each coalition a non-negative real value. By using this formal definition of a coalitional game, we neglected so far the fact that coalitional games, corresponding to real world examples, may have different representations. But this is a very important aspect when we are interested in the computational complexity of a problem, because an explicit representation is based on a table of 2^N rows, thus a structure of exponential size with respect to the number of players. Hence, the representation of a coalitional game plays an important role, not only with respect to the equivalence of coalitional games, but also in the context of computational complexity.

Consequently, it is imperative to have an appropriate representation for TU-games when the computational complexity of a problem should be analysed. Of course, it is not possible to have an appropriate game representation for all existing coalitional games, but for many interesting real-world problems, this is fortunately not a problem as stated by Megiddo [45]. He observed that for many coalitional games the game value can actually be calculated through a succinctly defined structure (game representation), and for these games he suggested that solutions could be found in polynomial time. As will be clear later, most of the games we introduce for our work are actually representable by graphs, thus we work with polynomial structures.

Apart from complexity-theoretic considerations, we are also interested in the expressive power and succinctness of representations of games. In Section 3, for example, we are going present and introduce various classes of game representations that represent the same class of coalitional games, namely all totally balanced games. Having several ways to describe a coalitional game, it is interesting to analyse the differences and similarities between different types of game representations.

The following four criteria, which are also applicable to game represenations, are standard properties to analyse different representation languages (see for example [14, 55]):

- **Expressivity:** The range of coalitional games covered by a particular game representation. Or, what classes of coalitional games are covered by this particular type of game representation?
- **Succinctness:** Is the representation succinct? Is one game representation more succinct than the other?
- **Efficiency/Complexity:** What is the computational complexity of related decision problems?
- **Simplicity:** Is the representation language easy to use to describe problems we would like to express?

Our focus in this thesis will be on efficiency and expressivity. To given an overview over games and game representations and how they are related to each other, we refer to Figure 2.1. Here we can distinguish two different layers:

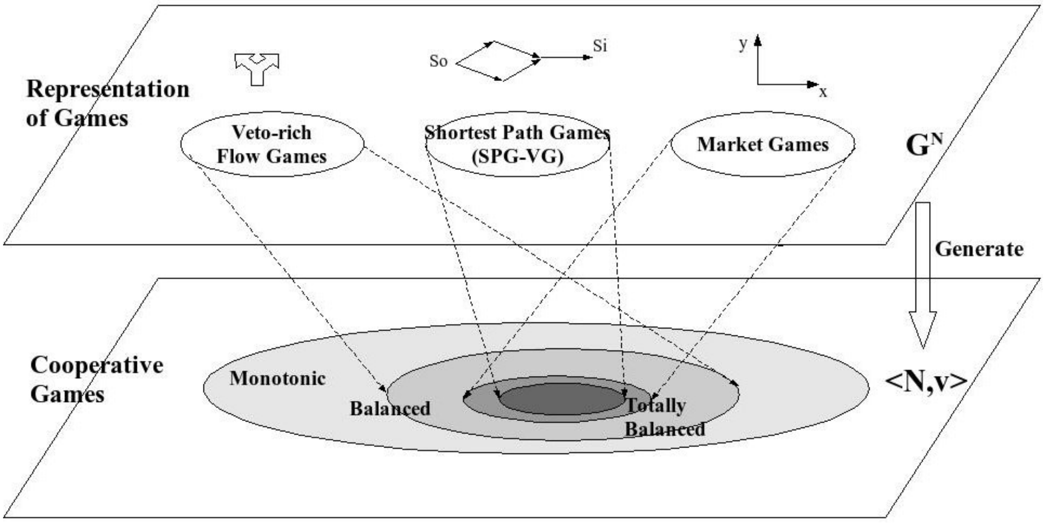

Figure 2.1: Representation of Coalitional Games

- **Upper Layer:** In this layer, a coalitional game is given in its original representation, thus in its actual problem domain. So, we can talk about concepts like shortest paths, maximal flow, etc. of a game.

- **Lower Layer:** In this layer, coalitional games are represented in characteristic function form $\langle N, v\rangle$. This is an elegant way to work with coalitional games, stripped from representation details, as a formal concept in cooperative game theory. The definitions and propositions, like for example balancedness, monotonic, null-player, etc., are all based on this basic representation of coalitional games.

Having talked about representations of coalitional games, we now introduce an appropriate notation to distinguish games from game representations:

We write G^N, G_1^N, G_2^N, etc. to denote particular game representations for coalitional games, which are defined for the set of individuals N in a particular representation language (or game definition). We then refer to the characteristic function of the coalitional game represented by G^N as v_{G^N}. For convenience, we introduce the following notion: we say that a game v is a Z game (e.g. v is a flow game), if $v := v_{G^N}$, where $G^N \in Z$ (G^N is a flow game representation). Thus, the actual type of the game represenation used to generate the game is known, but the explicit game representation is concealed. This is common in the literature, where the notions of game and game representation are not separated in a formal way. Note that in situations where the distinction between games and game representations is essential, we will exercise the appropriate care, otherwise the intended meaning can easily be deduced from the context.

We now introduce important notions involving games and game representations:

Given a game representation G^N, we can not only generate the characteristic function form $\langle N, v_{G^N}\rangle$ for this game, but also distinguish if it is a market game, shortest path game, etc.. This allows us to directly compare coalitional games with the same or similar game representation. Thus, it allows a more fine-grained analysis of coalitional games sharing special properties, like being based on graphs or voting systems. Based on this notation, we can also define classes of coalitional games by specifying which characteristics they should share. These classes can then be analysed separately.

For example, in Chapter 3 we will introduce several classes of shortest path games and compare them with respect to different aspects in later chapters.

Apart from this more fine-grained comparison of games, we also want to be able to compare coalitional games independent of their actual game representation or determine general properties of games. This is easily done by mapping a given game representation G^N to the corresponding characteristic function form $\langle N, v_{G^N}\rangle$, i.e. generating v_{G^N} from G^N and comparing it (in the lower layer) with coalitional games generated from other game representations or by determining if the game has particular properties. Note that if a game representation G^N is of characteristic function form ($G^N = \langle N, v\rangle$), then we can just write v instead of v_{G^N}.

We now define equivalence over game representations:

Definition 2.5.1. *We shall say that two N-player game representations G_1^N and G_2^N with transferable utility are* equivalent *($G_1^N \sim G_2^N$) if the characteristic functions $v_{G_1^N}$ and $v_{G_2^N}$, which are generated by G_1^N and G_2^N respectively are equal ($v_{G_1^N} = v_{G_2^N}$), i.e. for all $S \subseteq N$, $v_{G_1^N}(S) = v_{G_2^N}(S)$.*

We continue by introducing the notion of a “class” of game representations. A class can be defined in two ways:

1. We can define a class “extensionally” as a class of particular game representations (this also includes the characteristic function form $\langle N, v\rangle$).
2. We can define a class “intentionally” via a property, e.g., the class A of representations in characteristic function form $\langle N, v\rangle$ such that v is monotonic. Put otherwise, it is the class A of games $\langle N, v\rangle$ such that v is monotonic.

Due to the fact that we can easily generate the class of games from a given class of game representation, we often abuse notation when we say that A is the class of flow games or shortest path games.

We call two classes of game representations A and B *equivalent* if for all $G_1^N \in A$, there exists a game representation $G_2^N \in B$ such that $G_1^N \sim G_2^N$and vice versa. A class B of game representations *captures* another class A of game representations if for any $G_1^N \in A$, there exists a game $G_2^N \in B$ such that $G_1^N \sim G_2^N$.

We want to express the following relation between two classes of game representations A and B ($A \sqsubseteq B$):

- If B is an *extensionally* defined class of game representations, then we apply the capture relation defined above.
- If B is an *intentionally* defined class of game: This means that for every game $G^N \in A$, the characteristic function v_{G^N} generated from G^N has property B.

The notation for “$\sqsubset$” and “$=$” is defined accordingly. We concentrate in this thesis on the following intentionally defined classes of games: monotonic games (*MO*), balanced games (*BG*) and totally balanced games (*TBG*).

In the context of representations of coalitional games, Aziz et al. [3] introduced the property of a “reasonable representations” to prove a complexity-theoretic relation between the Shapley-Shubik and Banzhaf index (see Proposition 5.2.3).

Definition 2.5.2. *A representation of a simple game is considered* reasonable *if, for a simple game* $\langle N, v\rangle$ *the new game* $\langle N \cup \{i\}, v'\rangle$*(* i *is an arbitrary new player), where* $v(S) = 1$ *iff* $v'(S \cup \{i\}) = 1$*, can also be represented.*

Chapter 3

Shortest Path Games and Related Games

In this chapter we given an overview of different variants of shortest path games, and we furthermore present various graph-based games, like flow games, minimum cost spanning tree games, etc., as well as other interesting coalitional games, like market and linear production games. For each type of game representation, which we consider in our work, we also introduce a formal model. This is essential in the following chapters, when we prove properties and complexity results and determine how the various classes relate to each other.

Market games and linear production games are not directly involved in the research we conduct, but they are nevertheless interesting in a more indirect way. Both types of game representations, with their corresponding games, characterize an important class of coalitional games, namely the class of totally balanced games. This is also the case for flow games. As will be seen later, various variants of shortest path games are actually totally balanced as well. So, they can be related to the other coalitional games via the class of totally balanced games. We furthermore present a direct correspondence, which can be used to transform games, given as a flow, market or linear production games into one of the other two game representations. The correspondence between these classes of coalitional games may offer useful indications regarding properties and results for totally balanced games.

We start with a general introduction to market games, linear production games and flow games, where flow games receive some extra attention. Then we present some additional graph-based coalitional games, which have been analysed in the literature with respect to computational complexity of solution concepts. We continue by determining characteristics for shortest path games and other graph-based coalitional games. Afterwards we introduce different variants of shortest path games, which are based on several preselected characteristics. More details to market games, linear production games and flow games can be found in [20, 54].

For the first reading, we would advise the reader to study Section 3.5, as well as the consequent sections in detail and only skim through the previous sections to get a flavour of the different models. For the rest of this thesis, we will often refer to the different models of coalitional games introduced in this chapter, which can then be studied in detail, if the reader is interested.

3.1 Market Games

Shapley and Shubik [54] introduced a class of coalitional games, the so called market games (*MG*). Before we present this type of monotonic coalitional game, we have to introduce the concept of a market. The model is defined as follows:

A *market* is a mathematical model denoted by the tuple $\langle T, G, A, U \rangle$:

- T is a finite set, the set of traders.
- G is the non-negative orthant of a finite-dimensional vector space, often called the commodity space.
- $A = \{a^i : i \in T\}$ is an indexed collection of points in G, which are the initial endowments and
- $U = \{u^i : i \in T\}$ is an indexed collection of continuous, concave utility functions from G to the real numbers.

If we want to indicate that all players have equal tastes, i.e. for all $i \in T$, $u^i \equiv u$, we can denote the market by the more specific tuple $\langle T, G, A, \{u\} \rangle$. If S is any subset of T ($S \subseteq T$), an indexed collection $X^S = \{x^i : i \in S\} \subset G$ such that $\sum_S x^i = \sum_S a^i$ will be called a *feasible S-allocation* of the market $\langle T, G, A, U \rangle$.

A market $\langle T, G, A, U \rangle$ can be used to "generate" a game $\langle N, v \rangle$ in a natural way. We set $N = T$, and define v by

$v(S) = max_{X^S} \sum_{i \in S} u^i(x^i)$, all $S \subseteq N$,

where the maximum runs over all feasible $S-$ allocations. Any game that can be generated in this way, based on some market, is called a *market game*. The motivation to introduce this class of games is due to the fact that they characterize totally balanced game, a class of games that is quite interesting for our work. The same holds for linear production games, which will be presented next.

3.2 Linear Production Games

Linear production games (*LPG*) were introduced by Owen [46] and are widely used in the context of resource allocation and payoff distribution. They are $|N|$-person games in which the value of a coalition can be obtained by solving linear programming problems. We now introduce their formal model:

There are m kinds of resources and n kinds of products, which can be produced out of the given resources. The amount of resources of the k-th kind needed to produce a unit of product j is denoted by a_k^j.

The actual resources are not valued, i.e., there is no primary demand for them and therefore they cannot be sold on the market. However, there is a secondary demand for these resources. They can be used to produce goods which can be sold at a given market price. The market prices are contained in vector $\vec{c}$. Furthermore each player is initially given a vector $\vec{b^i} = (b_1^i, ..., b_m^i)$ of resources.

The value of a coalition $S \subseteq N$ is the maximum value (profit) the coalition can achieve having all the resources possessed by the members of the coalition. The maximum value can be obtained by solving the following linear programming problem for coalition S:

- $v(S) = max_{x \geq 0}(c_1 x_1 + ... + c_n x_n)$ such that
- $a_j^1 x_1 + a_j^2 x_2 + ... + a_j^n x_n \leq \sum_{i \in S} b_j^i \; \forall j = 1, 2, ..., m$

3.3 Flow Games

We first introduce the standard model for flow games and then introduce more general, as well as more specific types of flow games.

3.3.1 Network Flow Games (NFG)

The class of *network flow games* (*NFG*) (often called flow games) was first introduced by Kalai and Zemel [41]. It is defined as follows:

The game consists of a directed network flow graph $G = \langle V, A \rangle$, with capacities on the arcs $c : A \rightarrow \mathbb{R}_0^+$ and two special vertices, namely the source s and the sink t. Every arc $e \in A$ has a certain capacity $c(e) \geq 0$ and belongs to exactly one player $i \in N$. In most definitions of flow games it is allowed that a player $i \in N$ owns several arcs. For every vertex $x \in V$ let $B(x)$ denote the set of edges which start in x and $End(x)$ the set of arcs which end in x. For each $S \in 2^N$ let G_S be the network obtained from G by keeping all the vertices but removing all arcs which are not owned by a member of S. The new set of arcs is denoted by A_S. Trivially $G_N = G$ and $A_N = A$.

A *flow* from source to sink in such a network is a function f from $A_S$to $\mathbb{R}_0^+$ such that the following conditions are fulfilled:

- $0 \leq f(e) \leq c(e)$ for each $e \in A_S$ and
- $\forall x \in V \setminus \{s, t\} : \sum_{e \in B(x)} f(e) = \sum_{e \in End(x)} f(e)$

This means that the *value* of a flow is the net amount flowing out of the source and into the sink. This is expressed as follows:

Definition 3.3.1. *The* value *of a flow in a flow network $\langle V, A, c, s, t \rangle$ is formally defined as* $\sum_{e \in B(s)} f(e) - \sum_{e \in End(s)} f(e) = \sum_{e \in End(t)} f(e) - \sum_{e \in B(t)} f(e)$.

Based on these notions we can formally define a flow game as follows:

Definition 3.3.2. *Given a network $\langle V, A, c, s, t \rangle$, a* flow game $v(S)$ *based on this network is defined by the value of the maximum flow from source s to sink t in G_S.*

Note that we call a game v a flow game if there exists a network G such that $v(S) = F_G(S)$ for every coalition S.

3.3.2 Specific Flow Games

Cardinal Network Flow Games (CNFG)

Using the same environment as given above, but restricting the game such that every player owns exactly one arc, we can define a restricted class of flow game, called *cardinal network flow game* (*CNFG*). This type of flow game was introduced by Bachrach and Rosenschein [6].

Threshold Network Flow Games (TNFG)

Apart from the standard (value-based) definition of flow games, Bachrach and Rosenschein [6] introduced the variant of a *simple coalitional game of network flow*, which has a threshold k. Based on this threshold, given a coalition S, which controls the arcs A_S (in this case a player owns exactly one arc), it can be easily checked whether the coalition allows a flow of k from So to Si. The coalition wins if it allows such a flow, and loses otherwise:

$$v(S) = \begin{cases} 1 & \text{if } A_S \text{allows a flow of } k \text{ from } So \text{ to } Si; \\ 0 & \text{otherwise;} \end{cases}$$

We use the shortcut *TNFG* when referring to this game.

Connectivity Games (CG)

A simplified version of the threshold network flow game is the *connectivity game* (*CG*), a game where a coalition tries to establish a path from source to target. More precisely, a connectivity game is a threshold network flow game where each of the arcs has identical capacity, $c(e) = 1$, and the target flow value is $k = 1$. In such a scenario, the goal of a coalition is to have at least one path from So to Si:

$$v(S) = \begin{cases} 1 & \text{if } A_S \text{ contains a path from } So \text{ to } Si; \\ 0 & \text{otherwise;} \end{cases}$$

3.3.3 Generalizations of Flow Games

Pseudo Flow Games (PFG)

Kalai and Zemel [41] defined a pseudo-flow game (*PFG*), which is a flow game where some arcs are public. This means that these arcs are owned by all the players in the game.

Flow Games with Committee Control (FGCC)

Curiel, Derks and Tijs [21] introduced flow games with committee control (*FGCC*), which are defined as follows: In *flow games with committee control* the arcs are not owned by players as it is the case of standard flow game, but are controlled by committees consisting of subsets of players. Committee control can be effectively modeled with the aid of simple games. So, to every arc $e \in A$ a simple game w_e is assigned and a coalition S is said to control edge e iff $w_e(S) = 1$. If all the simple games w_e have veto-players, we call the resulting game a *veto rich flow game*.

Analogue to the flow games above, the network G_S is obtained from G by keeping all the vertices and removing all arcs, which are not controlled by S. Again, A_S denotes the resulting set of arcs and $G_N = G$, $E_N = E$. The actual game $v(S)$ is defined to be the value of a maximum flow in G_S.

This class of games is clearly a generalization of flow games, because flow games with ownership (network flow games) can be seen as flow games with committee control as well. In particular, the simple game which describes the control of an arc is then defined to be the game with the owner as dictator.

3.4 Some Graph-based Games

In this section we present some graph-based games from the literature.

3.4.1 Minimum Cost Spanning Tree Game (MCSTG)

The following problem, called a *minimum cost spanning problem* was introduced by Claus and Kleitman [17]: A network $G := \langle V, A \rangle$ is given whose set of nodes is $N \cup \{0\}$. The set $N = \{1, ..., n\}$ corresponds to the set of consumers and 0 to the central supplier. The weight $w(i, j) = w(j, i)$ of an arc $(i, j) \in A$ denotes the cost of connecting i to j. The minimum cost required to connect all the consumers to the central supplier (using arcs of the network) is the weight of the shortest spanning tree of G. Based on this environment, they proposed the following question: How should the total cost of a shortest spanning tree T be allocate to the consumers? The structure of the problem is ideal for a game-theoretic analysis, and there are actually several definitions of coalitional games to solve this problem. The definition, which can be seen as the standard definition, is the one proposed by Bird [11] and Claus and Granot [16]. It is defined as follows:

Based on the environment above, the *minimum cost spanning tree game*, *MCSTG* game for short, is a game on the set $N = \{1, ..., n\}$ of players, the grand coalition, that is to be connected to the supply node 0. The *cost* $c(S)$ of a coalition $S \subset N$ is by definition the weight of a minimum spanning tree in the subgraph induced by $S \cup \{0\}$. So, for every $S \subset N$, let $v(S) = c(S)$.

3.4.2 Vertex Connectivity Games (VCG)

A *vertex connectivity domain* consists of a graph $G = \langle V, E \rangle$, where the vertices are partitioned into primary vertices $V_p \subseteq V$, backbone vertices $V_b \subseteq V$, and standard vertices $V_s \subseteq V$. We require that $V_p \cap V_b = \emptyset$, $V_b \cap V_s = \emptyset$, $V_p \cap V_s = \emptyset$, and that $V = V_p \cup V_b \cup V_s$, so this is indeed a partition.

Given a vertex connectivity domain, Bachrach, Rosenschein and Porat [7] defined a *vertex connectivity game* (*VCG*). In this game, each player controls one of the standard servers. A coalition wins if it connects all pairs of primary vertices (so that it is able to send information between any two such primary servers). Let $|V_s| = n$, and consider a set of n players $N = (a_1, ..., a_n)$, so that agent a_i controls vertex $v_i \in V_s$. Given a coalition $S \subseteq N$ the set of vertices that S controls is denoted as $V(S) = \{v_i \in V_s \mid a_i \in S\}$. Coalition S can use either the vertices in $V(S)$ or the always available backbone vertices V_b. In this model, it is assumed that the coalition can also use any of the primary vertices V_p as well. A set of vertices $V' \subseteq V$ fully connects V_p if for any two vertices $u, v \in V_p$ there is a path $(u, p_1, p_2, ..., p_k, v)$ from u to v going only through vertices in V', for for all i we have $p_i \in V'$.

More formally: A vertex Connectivity Game is a simple coalitional game, where the value of a coalition $S \subseteq N$ is defined as follows:

$$v(S) = \begin{cases} 1 & \text{if } V(S) \cup V_b \cup V_p \text{ fully connects } V_p \\ 0 & \text{otherwise} \end{cases}$$

3.4.3 Spanning Connectivity Game (SCG)

Aziz, Lachish, Paterson and Savani [2] defined a coalitional game, called *spanning connectivity game* (*SCG*), which is based on an undirected weighted multigraph, where edges are players. So, before we can introduce what a spanning connectivity game is, we introduce what a multigraph is:

A *multigraph* is a graph, which is permitted to have multiple edges, that is, edges that have the same start and end nodes. Hence, two vertices may be connected by more than one edge. Formally a multigraph $G := \langle V, E, s \rangle$ consists of an underlying graph $\langle V, E \rangle$ with a multiplicity function $s : E \to \mathbb{N}$. So, for every edge $i \in E$, we have s_i edges in the multigraph.

For each connected multigraph $\langle V, E, s \rangle$, a spanning connectivity game $\langle E, v \rangle$ with players E and characteristic function v can be defined : For all $S \subseteq E$:

$$v(S) = \begin{cases} 1 & \text{if there exists a spanning tree } T = (V, E') \text{ such that } E' \subseteq S \\ 0 & \text{otherwise} \end{cases}$$

In this definition players are assigned to exactly one edge and each edge belongs to one player. Furthermore, it can easily be verified that spanning connectivity games are simple and monotone.

3.5 Shortest Path Games

To give a demonstration of the influence of characteristics of graph-based games, we had to select a promising sample type of a coalitional game, which offers various different characteristics that can be analysed in a complexity-theoretic context. For a start, shortest path games are a promising candidate, because there are two different types of shortest path games in the literature, which have quite different properties (e.g. expressive power). As indicated in the introduction, we introduce shortest path games also for another reason: They are similar to other graph-based games in the literature, which have been analysed from a computational perspective.

Shortest path games are a class of coalitional games that has not been considered, to our knowledge, in the context of computational complexity yet. There are two different game-theoretic approaches to the problem of shortest path problems, which have been introduced by Fragnelli, García-Jurado, and Méndez-Naya [33] and Voorneveld and Grahn [58].

We first introduce a strongly restricted variant of a shortest path game to give a flavour of the type of game representation that we are going to consider. Based on this basic pattern of how shortest path games are modeled, we present both variants from the literature and furthermore introduce a wider range of different variants including threshold versions of shortest path games. So, we introduce a framework of shortest path games, where each game possesses specific characteristics, which we are going to present and analyse in this section.

3.5.1 Basic Concepts of Shortest Path Games

Before we introduce the different variants of shortest path games, we want to give a detailed account of the basic construction ideas of this type of coalitional game. We start by giving an overview of a shortest path problem and the corresponding coalitional game.

The shortest path problems considered here are limited to a finite set of players, where each player owns arcs or vertices in a finite network. There are costs associated to the use of each arc or vertex and there are rewards involved with the transport of an item from the source to the sink of the network. The variant of a shortest path game that we are going to introduce now, borrows its basic outline from the original definition of shortest path games by Fragnelli et al. [33] and Voorneveld et al. [58], whereas we decided to restrict the model considerably.

Definition 3.5.1. *A* shortest path pre-problem Σ *is a tuple* $\langle V, A, So, Si\rangle$*, where*

- (V, A) *is a directed graph with two special elements, namely the* source *(*So*) and the* sink *(*Si*)*
- *We have a set* $A \subseteq V \times V$ *of directed arcs in the network.*

Definition 3.5.2. *A* path P *(*$So \to Si$*) in a directed graph is a sequence of vertices such that from each of its vertices there is an arc to the next vertex in the sequence. A path may be infinite, but a finite path always has a first vertex* So*, called its source, and a last vertex* Si*, called its sink.*

Now we present a minimal class of shortest path games, called *VSPG* (standing for value shortest path game). Let Σ be a shortest path pre-problem, where the arcs of the graph $\langle V, A\rangle$ are owned by a finite set of players N according to a total bijective map $o : A \to N$, such that $o(a) = i$ means that player i is the owner of edge a. Hence, every arc is assigned to exactly one player. We have a cost map c that assigns to every arc $a \in A$ a non-negative real number $c(a)$ ($c : A \to \mathbb{R}_0^+$).

Given the simple structure of *VSPG*, a path owned by players of coalition S is simply a sequence of vertices $(v_1, v_2, ..., v_m)$ such that $v_1 = So$, $v_m = Si$ and for each $k \in \{1, ..., m-1\}$ the arc (v_k, v_{k+1}) is owned by player $i_k \in S$. Let $P(S)$ denote the collection of all paths owned by coalition S. For any path P we denote by $o(P)$ the set of owners of the arcs in P.

Suppose that the transportation of a certain good from the source to the sink of Σ produces an income r and a cost given by the length of the path that was used. In particular, the costs associated to a path $p = (v_1, v_2, ..., v_m) \in P(S)$ are defined as the sum of the costs of its arcs: $cost(p) = \sum_{k=1}^{m-1} c(v_k, v_{k+1})$. Note that a coalition $S \subseteq N$ can transport the good only through paths owned by it and a path P is basically owned by a coalition S if $o(P) \subseteq S$.

Obviously, if a coalition S has to find a path from source to sink, it will choose among its alternatives in $P(S)$ the path with minimal costs. Define for each $S \in 2^N \setminus \{\emptyset\}$:

$$c(S) = \begin{cases} min_{p \in P(S)} cost(p) & \text{if } P(S) \neq \emptyset \\ \infty & \text{otherwise} \end{cases}$$

We overloaded c at this point, but it will be obvious from the context which function is meant. Note that the shortest path can be computed in polynomial time using for instance the well-known algorithm of Dijkstra [25].

We now put all the information necessary to describe a shortest path game together, and introduce the notion of a *shortest path game environment* σ, which is any such tuple $\langle N, \Sigma, o, c, r\rangle$.

Remember that a coalitional game with transferable utility can be represented as a pair (N, v), where N is a finite set of players and $v : 2^N \setminus \{\emptyset\} \to \mathbb{R}$ is a function that assigns to each coalition $S \in 2^N \setminus \{\emptyset\}$ its value $v(S) \in \mathbb{R}$. We can associate with σ the TU-game $\langle N, v_\sigma\rangle$ whose characteristic function $v_\sigma$is given by:

$$v_\sigma(S) = max\{r - c(S), 0\} = \begin{cases} r - c(S) & \text{if } S \text{ owns a path in } \Sigma \text{ and } c(S) < r \\ 0 & \text{otherwise} \end{cases} \quad \text{for every } S \subset N.$$

Hence, the environment σ and the definition of v_σ is our game representation G^N. The coalitional game associated with G^N has the following underlying intuition: if a coalition $S \in 2^N \setminus \{\emptyset\}$ transports its goods from source to sink, it will receive a total reward r and incur costs $c(S)$, the costs of the shortest path owned by S. If $r - c(S) > 0$, coalition S makes a profit. If $r - c(S) \leq 0$, coalition S can generate profit zero by simply doing nothing. Therefore, coalition S's profit is $max\{r - c(S), 0\}$.

Let's sum it up: A shortest path game of type *VSPG* is any such game $\langle N, v\rangle$ generated by a game representation G^N as defined above.

3.5.2 Notions and Properties

In this subsection we give some basic notions and definitions, necessary to reason about properties of shortest path games.

Definition 3.5.3. *Given a source So and a sink Si, a path P ($So \rightarrow Si$) is a* shortest path *if there is no other path P' such that $c(P') < c(P)$, where c is the cost function.*

Definition 3.5.4. *A path P in Σ is said to be a* profitable path *in σ if $r > cost(P)$, where r is the reward.*

Definition 3.5.5. *A shortest-veto (briefly s-veto) player of a shortest path game (N, v) is a player in N who owns at least one edge (vertex) of every shortest path (in Σ).*

To avoid trivial case, it is often assumed that $0 \leq c(N) < r(N)$, where $r(N)$ is the maximal reward with respect to N. This assumption implies that

- we have a strictly positive reward;
- the value $v(N)$ of the grand coalition is positive;

This assumption also avoids the zero game. In some papers it is even assume that $0 < c(N) < r(N)$ to makes sure that there are indeed costs to divide over the players. We do not apply this assumption, because $c(N) = 0$ offers a lot of interesting scenarios for shortest path games, as will be shown later.

3.6 Characteristics for Graph-based and Shortest Path Games

In what follows, we present general characteristics for graph-based games, as well as more specific characteristic for shortest path games. For each characteristic we have two options, which might influence the expressive power and succinctness of a graph-based game and also may have an effect on the computational complexity of solutions concepts.

To the structure of this section, we present for each characteristic the corresponding options and justify our decisions why it is meaningful or not to take both options of a characteristic into account. We also introduce a naming scheme for the options of characteristics, so that we can assign the corresponding options to each variant of a shortest path game. This allows us to distinguish variants of shortest path games properly.

3.6.1 General Characteristics

We start with characteristics of graph-based games in general.

Graphs: Directed vs Acyclic Directed

The options are: The underlying graph of a graph-based game is directed or acyclic directed

For some games, like shortest path games, there does not seem to be a strong motivation at first to distinguish both options. This is the case, because Dijkstra's shortest path algorithm [25] can be applied for both kinds of graphs to determine the shortest path. But, as will be shown later, there are indeed shortest path and other graph-based games where the computational complexity for some problems is actually different. So, it can be quite interesting in some situations to consider the reduction to acyclic directed graphs.

We use the label "DAG" when we want to restrict a game to acyclic directed graphs. We furthermore use the notation M(DAG) to indicate that a class of shortest path game, let's say M , is considered for inputs restricted to acyclic directed graphs.

Graphs: Graph vs Tree

The options are: A graph-based game is based on a standard graph or tree

The reduction of the underlying graph of a graph-based game to a tree is a severe simplification of the game and there are different graph-based games, as will be shown later, where the computational complexity for some problems changes indeed from intractable to tractable. So, it can be interesting in some situations to consider the reduction to trees.

We use the label "TREE" when we want to restrict a game such that the underlying graph is a tree. We use the following notation to indicate that a class of shortest path game, let's say M, is considered for input restricted to trees: M(TREE)

Players are attached to: Arcs vs Vertices

The options are: Players of graph-based games are attached to the graph's arcs or vertices.

Depending on the definition of graph-based games, there does not always seem to be a direct correspondence between variants of graph-based games, which only differ in this particular characteristic. But apart from marginal differences in expressive power for strongly restricted variants of games, we assume that there is no difference regarding computational complexity of solutions concepts applied to different variants of shortest path games. Hence, we decided not to take this characteristic into account.

We use the labels "OWNARC" and "OWNVERTEX" to distinguish both options.

Arc/Vertex can be owned by: One vs Many Players

The options are: An arc or vertex in a graph-based game can be owned by one player, or it can be owned by at least one player.

Similar to the case of flow games and generalizations of flow games, it appears promising to determine the influence of both options, especially with respect to expressive power. Due to the possible differences in expressive power, we think that it might also be interesting in the context of computational complexity. Note, this property only makes sense in combination with the characteristic that we introduce next.

We use the labels: "OWNED*" and "OWNED1"

A player owns arcs: One vs Many Arcs

The options are: A player in a graph-based game can own one arc, or at least one arc.

Similar to the characteristic above, it appears promising to determine the influence of both options of this characteristic.

We use the labels: "OWN*" and "OWN1"

3.6.2 Specific Characteristics

We now introduce some more specific characteristics for shortest path and flow games.

Source/Sink: Vertex vs Set

The options are: The source and sink of a shortest path or flow game are simple vertices or sets of vertices.

The question, if it is meaningful to distinguish these cases is easy to answer. For example, if players can own several arcs and arcs can be owned by several players in shortest path games, then it does not make sense to distinguish these cases. We can easily see that there is an equivalent game, based on a slightly altered game representation G_2^N. Let's take an arbitrary shortest path game representation G_1^N with a corresponding pre-problem $\Sigma = \langle V, A, s, t\rangle$ ($G = \langle V, A\rangle$), where the source s is a set of vertices. We can simply introduce a new vertex So to the graph, connect So to all vertices $t \in s$ (from So to t), assign costs 0 to all these arcs, declare So the new source and assign every player as the owner of the newly added arcs. Hence, the newly added arcs are public. For the target vertex we follow the same procedure, whereas the directionality of the arcs is reversed. The rest of the diagram stays unchanged. Let's call the altered game representation G_2^N. Then $v_{G_1^N}(S) = v_{G_2^N}(S)$ for all $S \subseteq N$. A similar reasoning holds for flow games.

Of course, this "trick" does not work anymore if players can own only one arc and every arc is only owned by exactly one player. In this case we might not be able to represent the game. Hence, there could be a difference in expressive power. But despite this aspect, we could still simulate the function by computing all combinations of source/sink pairs $So \times Si$ ($m * n$ if $m = |So|$ and $n = |Si|$) and choose the best outcome. Hence, the simulation is "polynomial" and therefore there is no difference in computational complexity.

Due to the fact that the impact of this characteristic is rather insignificant for most games, we mostly concentrate on shortest path and flow games where both, the source and sink, are a vertex. The only exception of a shortest path game with sets of vertices is the game introduced by Fragnelli et al. [33].

We use the labels "SoSi-Set" and "SoSi-Vertex" to distinguish both options of the characteristic.

Reward Scheme: Global vs Individual

The options are: A shortest path game can have a global reward scheme or an individual reward scheme.

Shortest path games of type *VSPG*, which we defined above, have a global reward scheme. But another possibility to reward coalitions of players for transporting goods, is to assign an individual reward to each player. By definition, the reward of a coalition in games with an individual reward scheme is the sum of the individual rewards of the coalition's players. Of Course, this reward is only granted in the case that the coalition transports the good successfully from the source to the sink.

As we will see later, this characteristic heavily influences the expressive power of shortest path games. So, there might be some effect with respect to the computational complexity of solution concepts as well.

We use the labels: "IReward" and "GReward"

Value vs Threshold

The options are: A shortest path game can return its normal value or be converted into a simple game by introducing a threshold.

It does not make sense to compare both options with respect to expressive power and computational complexity, because simple games are a very special type of coalitional game. But for some shortest path problems it is interesting to consider threshold versions to represent specific problems. We can also relate threshold variants of shortest path games to other simple graph-based games with respect to expressive power and the computational complexity of solution concepts.

We use the following label: "VALUE" and "THRESHOLD"

3.7 Variants of Shortest Path Games

In what follows, we sketch the models for various interesting variants of shortest path games. In cases where we deviate from the basic shortest path game (*VSPG*), which we defined above, we give additional details for the corresponding model. We have the following standard building scheme for the game representation:

1. pre-problem Σ
2. shortest path cooperative situation σ
3. define shortest path game on σ

Some of the variants of shortest path games introduced below might appear overly simplified for practical use, but they can nevertheless be helpful for several reasons:

- To demonstrate the effects of characteristics on basic variants of shortest path games. When these results are intractable, we can immediately deduce that the more general games are intractable as well.
- Many complexity results for graph-based games have been determined for games with very basic characteristics. So, to be able to compare the results of shortest path games with other graph-based games in a reasonable way, we have to consider games with basic characteristics as well.

Note that both of the originally defined shortest path games are based on acyclic directed graphs, whereas we are interested in directed graphs. Thus, we decided to concentrate on directed graph in general, and treat acyclic directed graphs as special cases.

Value Shortest Path Game* (VSPG*)

We denote by *VSPG** the class of shortest path games having the following characteristics: [SOSI-VERTEX, OWNARC, OWNED1, OWN*, GREWARD, VALUE]

The model is similar to *VSPG*, except the fact that players can own several arcs. Regarding the basic characteristics, it is similar to the definition of network flow games.

Value Shortest Path Game*+ (VSPG*+)

We denote by *VSPG*+* the class of shortest path games having the following characteristics: [SoSi-Vertex, OwnArc, Owned*, Own*, GReward, Value]

The model is similar to *VSPG,* except the fact that an arc can be owned by several players and that players can own several arcs. From an abstract viewpoint, this variant of shortest path game has similar characteristics compared to the generalized variants of flow games.

Threshold Variant of VSPG (TSPG)

We denote by *TSPG* the class of threshold shortest path games having the following characteristics: [SoSi-Vertex, OwnArc, Owned1, Own1, GReward, Threshold]

The model is similar to *VSPG*, except that it is a threshold game. We formalize the details that differ from *VSPG* as follows:

We have to introduce a threshold T and slightly change the definition of v_σ in *VSPG*. A threshold shortest path cooperative environment σ is a tuple $\langle \Sigma, N, o, c, r, T \rangle$. We can associate with σ the TU-game $\langle N, v_\sigma \rangle$ whose characteristic function $v_\sigma$is given by:

$$v_\sigma(S) = \begin{cases} 1 & \text{if } S \text{ owns a path in } \Sigma \text{ and } r - c(S) \geq T \\ 0 & \text{otherwise} \end{cases} \quad \text{for every } S \subset N.$$

A threshold shortest path game is any game $\langle N, v_{G^N} \rangle$ that is generated by a threshold shortest path game representation G^N, which is based on the shortest path cooperative environment σ and the corresponding building construction for the characteristic function v_{G^N}.

Shortest Path Game - Fragnelli (SPG-F)

We denote by *SPG-F* the class of shortest path games having the following properties: [DAG, SoSi-Set, OwnVertex, Owned1, Own*, GReward, Value]

This class of shortest path games was introduced by Fragnelli et al. [33]. Note that we have only minor changes compared to *VSPG*, namely that the shortest path pre-problem Σ has to be modified: We have directed acyclic graphs, the source and sink are sets of vertices and we have to take into consideration that this time the vertices are owned by a set of players N according to a map $o : V \to N$.

The rest of the model follows as in the case of *VSPG*.

Shortest Path Game - Voorneveld and Grahn (SPG-VG)

We denote by *SPG-VG* (Shortest Path Game - Voorneveld and Grahn) a class of shortest path games having the following properties: [DAG, SoSi-Vertex, OwnArc, Owned*, Own*, IReward, Value]

This game was introduced by Voorneveld and Grahn [58]. Opposed to the models of *VSPG**, *VSPG*+* and *SPG-F*, the authors introduced a class of shortest path games with the main difference that we have individual rewards assigned to every player instead of one global reward for the game.

Now we give an overview of the changes compared to *VSPG*: The shortest path pre-problem Σ as defined above stays the same, except for the restriction that we consider directed acyclic graphs.

We have to consider that more than one player can own an arc between two vertices, and that the costs of an arc depends on its owner cost allocation. Hence, we have to define the cost (weight) function slightly different as $c : \cup_{i \in N}\{i\} \times o^{-1}(i) \rightarrow \mathbb{R}_0^+$, i.e. the cost assignment to arc (a, b) owned by player $i \in N$ is $c(i, (a, b)) \in \mathbb{R}_+$.

The transportation of a certain good from a source to a sink of Σ produces an income of $r_i \in \mathbb{R}_+$ for each player $i \in N$ in the coalition and a cost given by the length of the path that was used. In particular the costs associated to a path $p = (v_1, i_1, v_2, i_2, ..., i_{m-1}, v_m) \in P(S)$ are defined as the sum of the costs of its arcs: $cost(p) = \sum_{k=1}^{m-1} c(i_k, (v_k, v_{k+1}))$.

The shortest path cooperative situation σ is defined slightly different as the tuple $\langle \Sigma, N, o, c, (r_i)_{i \in N} \rangle$ and we can associate with σ the TU-game $\langle N, v_\sigma \rangle$ whose characteristic function $v_\sigma$is given by:

$$v_\sigma(S) = max\{r(S) - c(S), 0\} = \begin{cases} r(S) - c(S) & \text{if } S \text{ owns a path in } \Sigma \text{ and } c(S) < r(S) \\ 0 & \text{otherwise} \end{cases}$$

XSPG

We denote by *XSPG* the class of shortest path games having the following characteristics: [SoSi-Vertex, OwnArc, Owned1, Own1, IReward, Value]

The model is similar to *VSPG*, except the fact that we have individual rewards for every player instead of a global reward.

Threshold variant of XSPG (TXSPG)

We denote by *TXSPG* the class of shortest path games having the following characteristics: [SoSi-Vertex, OwnArc, Owned1, Own1, IReward, Threshold]

The model is similar to *TSPG*, except the fact that we have individual rewards for every player instead of a global reward.

XSPG*

We denote by *XSPG** the class of shortest path games having the following characteristics: [SoSi-Vertex, OwnArc, Owned1, Own*, IReward, Value]

The model is similar to *XSPG*, except the fact that a player can own several arcs.

XSPG*+

We denote by *XSPG*+* the class of shortest path games having the following characteristics: [SoSi-Vertex, OwnArc, Owned*, Own*, IReward,Value]

The model is similar to *XSPG*, except the fact that a player can own several arcs and an arc can be owned by several players. Note that *XSPG*+* restricted to DAG graphs is *SPG-VG*.

Chapter 4

Properties and Relations

In this chapter we prove or present several properties for different classes of coalitional games, which were introduced in the previous chapter and relate them to each other with respect to their expressive power. This allows us to work out similarities between the different classes of coalitional games. Note that all these classes of games are non-negative and monotonic by definition.

4.1 Market Games and Linear Production Games

We now present two equivalence results, namely for market games and linear production games.

Theorem 4.1.1. *A game is a market game iff it is non-negative totally balanced. ([54], Theorem 5)*

The authors pointed out that when market games are extended in a way that diseconomies exist, then this extended market games become non totally balanced or non balanced. Diseconomies can for example arise when an arc can be owned by more than one player. This is interesting, because flow games, as we will see later, show a similar behaviour when more than one player can own an arc.

Shapley and Shubik [54] proved that every totally balanced game is a market game in two steps: They generate a special market from an arbitrary totally balanced game, called a direct market and then obtain a market game that is based on this directed game.

Note that the first step, the generation of the direct market, in particular the creation of the utility function is exponential. Hence, the reduction in this constructive equivalence proof takes exponential time. We have a similar result for linear production games.

Theorem 4.1.2. *A game is a linear production game iff it is non-negative totally balanced [20, 51].*

Similar to the above equivalence result, the generation of a linear production game equivalent to a given totally balanced game requires exponential time with respect to the number of players (see [15]).

We primarily presented these equivalences to show that these games characterize totally balanced games. So, the equivalences give us a more concrete idea about possible representations of totally balanced games. Furthermore, it is interesting to see that market games, linear productions games, and flow games, quite different types of games, can be directly translated from one game representation into another.

4.2 Flow Games

Kalai and Zemel [41] proved that flow games are totally balanced (non-negative), and conversely that every non-negative totally balanced game can be derived from a flow situation.

Theorem 4.2.1. *A game is a flow game iff it is non-negative totally balanced. ([41], Theorem 2)*

The corresponding proof is constructive, but similarly to the equivalence results above, the generation of a flow game from a totally balanced game requires exponential time with respect to the number of players.

Now, we give some obvious results: Network flow games are clearly monotonic. Given that *NFG* = *TBG*, we can deduce from Proposition 2.2.13 that *NFG* $\sqsubset$ *MO*.

A result that will be useful later, is the following:

Proposition 4.2.2. *The class of connectivity games (CG) is not balanced.*

Proof. An example of a non-balanced connectivity game is the following: Let G^N be a connectivity game representation with just two disjoint paths (except source and sink) S_1 and S_2, where $S_1, S_2 \subseteq N$, $S_1 \cap S_2 = \emptyset$, $S_1 \cup S_2 = N$. Furthermore, let v_{G^N} be the game generated from G^N, where $v_{G^N}(S_1) = 1$ and $v_{G^N}(S_2) = 1$. So, for all $S \supseteq S_1$ or $S \supseteq S_2$, $v_{G^N}(S) = 1$ and for all the other coalitions $T \subseteq N$, $v_{G^N}(T) = 0$. Now, let's assume for the sake of a contradiction that $Core(v_{G^N}) \neq \emptyset$. Let x be any allocation such that $x \in Core(v_{G^N})$ (there is at least one). By group rationality we have that $x(S_1) \geq v_{G^N}(S_1) = 1$ and $x(S_2) \geq v_{G^N}(S_2) = 1$. But from this we can immediately deduce that $2 \leq x(S_1) + x(S_2) = x(N) \neq v_{G^N}(N) = 1$. Hence, the efficiency conditions does not hold for allocation x, and therefore $x \notin Core(v_{G^N})$ (Contradiction). Applying Theorem 2.2.11 we get that v_{G^N} is not balanced and therefore the class of connectivity games is not balanced. □

4.2.1 Pseudo Flow Games

Kalai and Zemel [41] showed that for every monotonic game there exists an equivalent pseudo-flow game. The converse follows immediately, because by adding players to a coalition the flow will at least stay the same. Hence, there is an equivalence between monotonic games and pseudo-flow games.

Theorem 4.2.3. *v is a pseudo-flow game iff v is monotonic ([41], Theorem 3)*

4.2.2 Flow Games with Committee Control

The following proposition is an immediate corollary of Theorem 4.2.3 and Corollary 4.4.2 (iii).

Corollary 4.2.4. *v is a flow game with committee control iff v is monotonic*

Based on the equivalence result above and Proposition 2.2.13, the class of flow games with committee control does not directly correspond to the class of balanced games.

By restricting the class to flow games with veto control, Curiel et al. [21] showed that the class of non-negative balanced games can be obtained by exactly the class of flow games with veto control. So, the family of non-negative balanced games coincides precisely with the family of veto rich flow games.

Theorem 4.2.5. *v is a veto rich flow game iff v is a non-negative balanced game. ([21], Theorem 1 and Theorem 5)*

4.3 Properties of Shortest Path Games

In this section we determine properties of various variants of shortest path games. We concentrate on their expressive power, as well as the relationship between different games with respect to expressive power.

VSPG

Some coalitional games of type *VSPG* are balanced (totally balanced), but not all of them. We demonstrate this with the following two examples:

Example 1

Let $N = \{1, 2, 3\}$,$V = \{So, Si, v\}$, $A = \{a, b, c\}$ ($a = (So, Si)$, $b = (So, v)$, $c = (v, Si)$), $o(a) = 1$, $o(b) = 2$, $o(c) = 3$, $c(b) = c(c) = 1$ and $c(a) = 2$ and $g = 7$. So, given game representation G^N of type *VSPG*, with $\sigma = \langle \Sigma, c, o, g\rangle$ we have $\langle N, v_{G^N}\rangle$ with $v_{G^N}(\emptyset) = 0$, $v_{G^N}(\{1\}) = 5$, $v_{G^N}(\{2\}) = v_{G^N}(\{3\}) = 0$, $v_{G^N}(\{2, 3\}) = 5$, $v_{G^N}(\{1, 2\}) = v_{G^N}(\{1, 3\}) = 5$ and $v_{G^N}(\{1, 2, 3\}) = 5$.

Let's assume for the sake of a contradiction that $Core(v_{G^N}) \neq \emptyset$. Now, let x be any allocation such that $x \in Core(v_{G^N})$ (there is at least one). By individual rationality we have $x_1 \geq v_{G^N}(\{1\}) = 5$ and by group rationality we get $x_2 + x_3 \geq v_{G^N}(\{2, 3\}) = 5$. But this violates the efficiency condition, because $x(N) > v_{G^N}(N) = 5$. Hence, $x \notin Core(v_{G^N})$ (Contradiction).

So, we have a game of type *VSPG* that has an empty core. Hence, by Theorem 2.2.11 this game is not balanced, and therefore not totally balanced as well.

Example 2

Let $N = \{1, 2, 3\}$,$V = \{So, v, v', Si\}$, $E = \{a, b, c\}$ ($a = (So, v)$, $b = (v, v')$, $c = (v', Si)$), $o(a) = 1$, $o(b) = 2$, $o(c) = 3$ $c(a) = c(b) = c(c) = 1$ and $g = 4$. So, given game representation G^N of type *VSPG*, with $\sigma = \langle \Sigma, c, o, g\rangle$we have $\langle N, v_{G^N}\rangle$ with $v_{G^N}(N) = 1$ and for all $X \subset N$, $v_{G^N}(X) = 0$. This game can easily be expressed as a network flow game and is therefore by Theorem 4.2.1 totally balanced.

Based on these examples we can deduce that *VSPG* is not captured by the class of balanced or totally balanced games (*VSPG* $\not\sqsubseteq$ *TBG*; *VSPG* $\not\sqsubseteq$ *BG*). Clearly, this holds for more general classes, like *VSPG** and *VSPG*+* as well. By applying minor changes to the examples, this also holds for *TSPG*. So, what about the other way around? Are totally balanced games captured by *VSPG*? This can be negated by the following result.

Proposition 4.3.1. *The class of totally balanced games is not captured by* VSPG *(TBG $\not\sqsubseteq$ VSPG).*

Proof. Let $G := \langle V, A\rangle$ be a directed graph with $V = \{So, v_1, v_2, Si\}$ and $A = \{a, b, c, d\}$, where $a = (So, v_1)$, $b = (So, v_2)$,$c = (v_1, Si)$ and $d = (v_2, Si)$. Now, let $N = \{1, 2, 3\}$ be the set of players. We define the ownership relation $o := \{(a, 2), (b, 3), (c, 1), (d, 1)\}$ and the capacity function $c := \{(a, 2), (b, 3), (c, 1), (d, 1)\}$. Based on this definition we can define a flow game v: $v(\{1\}) = v(\{2\}) = v(\{3\}) = v(\{2, 3\}) = 0$, $v(\{1, 2\}) = v(\{1, 3\}) = 1$ and $v(\{1, 2, 3\}) = 2$. Given that v is a flow game, we can immediately deduce by Theorem 4.2.1 that v is totally balanced.

We can now show that v cannot be represented as a shortest path game of type *VSPG*. Given that in games of type *VSPG* each player is assigned to exactly one arc, we are restricted to graphs with 3 arcs. Let's call them a, b and c. Note that it does not make sense to consider graphs with less than 3 arcs, because under these circumstances we are not able to generate all of the three different values of v.

We have three different configurations of graphs that are allowed:

1. All three arcs are in a sequence from So to Si: Under this configuration we would not be able to generate the value for coalition $\{1, 2\}$: $v(\{1, 2\}) = 1$
2. Arc a and b build a path from So and Si, and c connects So and Si immediately: This is also not possible, because apart from the grand coalition there would be another coalition that has the highest value 2.
3. Arc a and b build a path from So and Si, and c leads to an arbitrary vertex that is not the sink: This is not possible, because in this case we can only generate two different values.

Hence, given that no graph configuration would work, there is no shortest path game of type *VSPG* that represents flow game v. □

We now continue with another interesting property, namely monotonicity.

Proposition 4.3.2. *VSPG* $\sqsubset$ *MO*

Proof. It is clearly the case (by definition) that *VSPG* is monotonic (*VSPG* $\sqsubseteq$ *MO*), but *MO* $\sqsubseteq$ *VSPG* does not hold. We present a counterexample: Let $N = \{1, 2\}$ and $v(\emptyset) = 0$, $v(1) = 1$, $v(2) = 2$ and $v(N) = v(12) = 3$. Clearly $v \in$ *MO*. Due to the strict limitation that every player has exactly one arc and every arc belongs to only one player we have only two possible directed graphs, namely a chain of two arcs from So to Si or a single arc from So to Si and the other arc to an arbitrary vertex.

Case 1: If we have a chain, we can immediately deduce that $v_{G^N}(1) = 0$, $v_{G^N}(2) = 0$ and $v_{G^N}(\emptyset) = 0$ for any game representation G^N. So, it is impossible to find a game representation G^N of type *VSPG* such that $v_{G^N} = v$ using a chain.

Case 2: If we have a single arc, we have the same problem, because we cannot find a game representation G^N of type *VSPG* such that all different values of v can be expressed.

Hence, there exist no game representation G^N of type *VSPG* such that $v_{G^N}(S) = v(S)$ for all $S \subseteq N$. □

This result is obvious and there are many more counter-examples, but as we will learn later, it is the case that *VSPG** $=$ *MO*. So, taking both results into account, we see that by changing one characteristic we have a class of shortest path games that generates precisely all monotonic coalitional games.

Another interesting result it that *VSPG* captures the class of connectivity games, a result that will be useful in the next chapter.

Proposition 4.3.3. *The class of connectivity games is captured by* VSPG *(CG* $\sqsubseteq$ *VSPG).*

Proof. Connectivity games, as introduced in subsection 3.3.2, can be easily modeled in *VSPG*. Given a game representation G_1^N of type *CG*, which consists of a network flow graph $G = \langle V, A \rangle$ (directed graph) with capacities on the edges $c : A \rightarrow \mathbb{R}_0^+$, a source vertex So, a target vertex Si, and a set of players N, where each player $i \in N$ controls exactly one arc, let's call it a_i. Furthermore, all arcs $a \in A$ have an identical capacity, $c(a) = 1$, and the target flow value is $k = 1$.

We can now easily define a game representation G_2^N of type *VSPG* by using the settings of the *CG* game representation. Now we just have to define a reward $r := 1$ and assign 0 as the cost value to every arc.

Let $S \subseteq N$ be any coalition. Now, if $v_{G_2^N}(S) = 1$, then there is a connecting path for coalition S from So to Si with cost 0. So, there is a connection and $v_{G_1^N}(S) = 1$. Otherwise, if $v_{G_2^N}(S) = 0$, then there is no path, and therefore there is no connection. Hence $v_{G_1^N}(S) = 0$. □

Given that the class *VSPG** and *VSPG*+* are more general than *VSPG*, they also capture connectivity games. Hence, *VSPG*, *VSPG** and *VSPG*+* are not totally balanced by Proposition 4.2.2.

VSPG*

As mentioned above, the class *VSPG** precisely correlates with the class of monotonic games. To prove this result we had to adapt the proof of Fragnelli et al. [33] to work without multiple vertices in So and Si and a slightly altered game representation.

Proposition 4.3.4. *VSPG** $=$ *MO*

Proof. We can see immediately that every game generated by a game representation of type *VSPG** is monotonic (*VSPG** $\sqsubseteq$ *MO*). For the other direction, let's assume that $v \in$ *MO*. We now construct a game representation G_N of type *VSPG** that has as many disjoint paths (except vertices So and Si) as there are coalitions with positive value ($v(S) > 0$ for $S \subseteq N$). We take $g \geq v(N)$ and define Σ,c and o such that for every $S \subseteq N$ with $v(S) > 0$ there is a unique path P_S with $o(P_S) = S$ and $c(P_S) = g - v(S)$ (we attach the cost for the whole path to the last arc of the path and all the other arcs of the path are assigned the value 0 as weight). Given that $\langle N, v \rangle$ is monotonic, we can immediately see by construction that $v = v_{G^N}$. This is because given a coalition $S \subseteq N$ the shortest path is always the one corresponding to path P_S and not any of the paths P_T for $T \subset S$ (at best P_T might have equal cost). □

Example: Let $\langle N, v \rangle$ be a monotonic game with $N = \{1, 2, 3\}$ and $v(\emptyset) = 0$, $v(1) = 0$, $v(2) = 1$, $v(3) = v(1, 3) = v(2, 3) = 2$, $v(1, 2) = 3$ and $v(N) = 5$. Now we consider environment $\sigma = \langle N, \Sigma, o, c, g \rangle$ with $g = 5$, where the environment is defined as follows (see Figure 4.1)

The same also holds for another variant of *VSPG**, namely in the case where several players might share an arc (*VSPG*+*). This, of course, follows immediately because it is a more general game.

Threshold Shortest Path Game (TSPG)

Proposition 4.3.5. *The class of connectivity games is captured by* TSPG *(CG $\sqsubseteq$ TSPG).*

Proof. This proof is very similar to the proof of Proposition 4.3.3, why we only mention the differences in the model: We want to show that connectivity games can be modeled in *TSPG*. Differently to the proof of *VSPG* we define a global reward as $r := |A| + 1$ and set the threshold to be $T = 1$. □

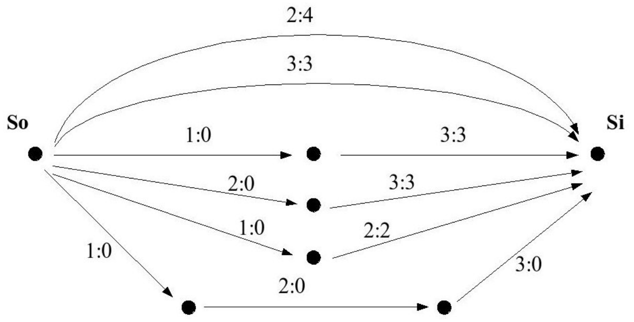

Figure 4.1: Expressing a monotonic game as a shortest path game of type *VSPG**

Shortest Path Game (Fragnelli; SPG-F)

The *SPG-F* class of shortest path games is shown to coincide with the class of monotonic games.

Proposition 4.3.6. *SPG-F* $=$ *MO ([33], Proposition 1)*

Based on the proposition above and Proposition 2.2.13 we can directly deduce that not all shortest path games of type *SPG-F* are balanced (have a non empty core) or totally balanced.

Fragnelli et al. [33] showed that for a shortest path game of type *SPG-F* to be balanced, two rather strong restrictions have to be fulfilled: a certain reduced game needs to be balanced, and certain veto players have to take important positions in the game.

Definition 4.3.7. *With every non-trivial shortest path game* $\langle N, v\rangle$ *with a non-empty set of s-veto players* V*, we associate a TU-game* $\langle V, \overline{v}\rangle$ *such that* $\overline{v}(\emptyset) = 0$ *and* $\overline{v}(T) = v(T \cup (N \setminus V))$ *for every non-empty* $T \subset V$.

Theorem 4.3.8. *Let* $\langle N, v\rangle$ *be a non-trivial shortest path game with a set of s-veto players* V*. Then,* $\langle N, v\rangle$ *is balanced iff the two following conditions hold: ([33], Theorem 2)*

1. V *is non-empty and* $\langle V, \overline{v}\rangle$ *is balanced.*
2. *Every profitable path in* v *contains a node owned by a s-veto player.*

Shortest Path Game (Voorneveld and Grahn; SPG-VG)

Voorneveld and Grahn [58] showed that every shortest path game of type *SPG-VG* is totally balanced. This is based on the following result: every efficient allocation in which each of the players contributes a non-negative amount, not exceeding his reward, to the cost of the shortest path, yields a core element ([58], Proposition 3.1).

Proposition 4.3.9. *Let* G^N *be a game representation of type* SPG-VG *with shortest path cooperative situation* $\sigma := \langle V, (A_i)_{i\in N}, c, (r_i)_{i\in N}\rangle$*, where* v_{G^N} *is the associated game. Take* $B = \{x \in \mathbb{R}^N \mid x(N) = v_{G^N}(N)$ *and* $x_i \in [0, r_i]$ *for each* $i \in N\}$*. Then* $B \subseteq Core(v_{G^N})$.

Given that the proof of this result does not depend on the ownership function o and the condition that the graph is acyclic, the same result holds for *XSPG*, *XSPG** and *XSPG*+*.

Note that the proof is based on the assumption that $0 < c(N) < r(N)$, whereas we have a weakened assumption ($0 \leq c(N) < r(n)$) as mentioned in Chapter 3. So, we have to prove for condition $c(N) = 0$ and $c(N) < r(N)$, whether all games v of type *SPG-VG* (*XSPG*, *XSPG** and *XSPG*+*) are totally balanced.

It can easily be verified that allocation x, an allocation where each player receives his or her individual reward ($\forall i \in N, x_i = r_i$) is efficient and fulfills the group rationality property. Hence, x is an element of the core of v. The same holds for all subgames of v, and therefore v is totally balanced.

Given that any game of type *SPG-VG* it totally balanced, we can immediately deduce that *SPG-VG* $\sqsubset$ *MO* and *SPG-VG* $\sqsubseteq$ *NFG*. The same holds for *XSPG*, *XSPG** and *XSPG*+*. The question, if the class of shortest path games *SPG-VG* captures the class of flow games (hence *SPG-VG* $=$ *NFG*) is still an open problem.

XSPG

By definition of *XSPG*, we have that *XSPG*$(DAG) \sqsubseteq$ *SPG-VG*. So, we have *XSPG*$(DAG) \sqsubseteq$ *SPG-VG* $\sqsubset$ *MO* and therefore *XSPG*$(DAG) \sqsubset$ *MO*. We prove now that *XSPG*(DAG) does not capture the class of connectivity games:

Proposition 4.3.10. *The class of connectivity games is not captured by XSPG*(DAG)

(CG $\not\sqsubseteq$ *XSPG*(DAG)*).*

Proof. Let's assume for the sake of a contradiction that *CG* $\sqsubseteq$ *XSPG*(DAG). We know that all games of the class *SPG-VG* are totally balanced and *XSPG*(DAG) is a subclass of *SPG-VG*. Hence, *XSPG*(DAG) is totally balanced, and by assumption *CG* is totally balanced as well. But this cannot be the case, because *CG* is not balanced by Proposition 4.2.2, and therefore not totally balanced. □

The same holds for *XSPG**(DAG) and *XSPG*+*(DAG).

TXSPG

Proposition 4.3.11. *The class of connectivity games is captured by TXSPG (CG* $\sqsubseteq$ *TXSPG).*

Proof. This proof is very similar to the proof of Proposition 4.3.3, why we only mention the differences of the model: We want to show that for all $G_1^N \in$ *CG* , there exists a $G_2^N \in$ *TXSPG* such that $G_1^N \sim G_2^N$.

Differently to the proof of *VSPG*, we assign to every player $i \in N$ an individual reward $r_i = 1$ and set the threshold to $T = 1$. Now, If $v_{G_2^N}(S) = 1$, then there is a connecting path for coalition S from So to Si. Otherwise, if $v_{G_2^N}(S) = 0$, we can easily deduce that there is not path: Let's assume for the sake of a contradiction that there is a path. So, due to the fact that all arcs have a cost of 0 attached to them and all individual rewards are 1 we have $c(S) = 0$ and $r(S) > 0$, what leads immediately to a contradiction, because $0 < r(S) - c(S) = v_{G_2^N}(S)$. Hence, a connectivity game can also be expressed as a shortest path game of type *TXSPG*. □

Given that *TXSPG* captures all connectivity games, and there are connectivity games that are not balanced (see Proposition 4.2.2), there are *TXSPG* games that are not balanced. Hence, the class *TXSPG* is not balanced or totally balanced.

4.4 Relations between the Different Classes of Coalitional Games

In this section we give an overview of the relations between different classes of coalitional games, which we introduced in the previous chapter.

We start with the equivalence results, which nicely relate different classes of coalitional games with respect to their expressiveness: Given a TU-game $\langle N, v\rangle$,

v is a market game iff v is a linear production game iff v is a flow game

This allows us to see totally balanced games from a different viewpoints. We furthermore know that *SPG-VG* $\sqsubseteq$ *NFG*, and therefore we can express every shortest path game of type *SPG-VG* as a market, linear production and flow game. The same holds for *XSPG, XSPG** and *XSPG*+*.

We now relate the variants of shortest path games given by Fragnelli et al. [33] and Voorneveld et al. [58] with respect to their expressive power:

Proposition 4.4.1. *The following inclusions hold:*

- *SPG-VG* $\sqsubset$ *SPG-F*
- *SPG-VG* $\sqsubset$ *VSPG**

Proof. Let G_1^N be a game representation of type *SPG-VG*. So, $v_{G_1^N}$ is monotonic and by Proposition 4.3.6 we can immediately deduce that there is a game representation G_2^N of type *SPG-F* such that $v_{G_1^N}(S) = v_{G_2^N}(S)$ for all $S \subseteq N$. Hence *SPG-VG* $\sqsubseteq$ *SPG-F*. But, it cannot be the case that *SPG-VG* $=$ *SPG-F*, because *SPG-VG* is totally balanced, whereas *SPG-F* is not. Thus, *SPG-VG* $\sqsubset$ *SPG-F*. The same follows for *VSPG**, where we use Proposition 4.3.4. □

This result does not only show that the class *SPG-F* generates more games than *SPG-VG*, but it also indicates that the global reward scheme has more influence on the expressive power of a shortest path games than the individual reward scheme.

We can verify this as follows: Given that *VSPG** $=$ *MO* and *VSPG*+* is monotonic, we have that *VSPG** $=$ *VSPG*+*, and therefore *SPG-VG* $\subset$ *VSPG*+*. Given that Proposition 4.3.4 does not rely on the distinction between directed graphs and directed acyclic graphs, we have that *SPG-VG* $\subset$ *VSPG*+*(DAG). If we now compare the characteristics for *SPG-VG* ([DAG, SoSi-Vertex, OwnArc, Owned*, Own*, IReward, Value]) and *VSPG*+*(DAG) ([DAG, SoSi-Vertex, OwnArc, Owned*, Own*, GReward, Value]), we see that both classes of games only differ in one characteristic, namely the rewarding scheme. The result above is even stronger, because it states that even by weakening *VSPG*+*(DAG) (downgrading Owned* to Owned1) *SPG-VG* is still a proper subclass.

We have several more results, which follow immediately from propositions proved above:

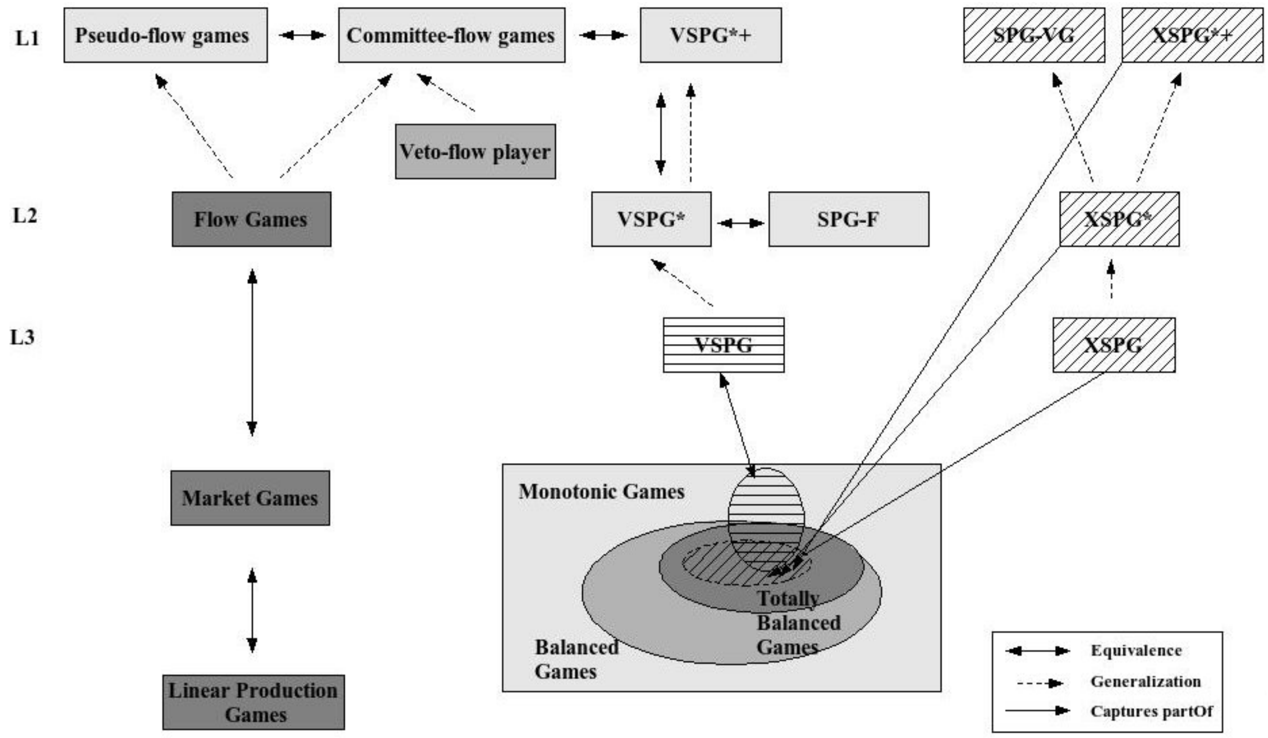

Figure 4.2: Relationship between games

Corollary 4.4.2. *The following holds:*

1. *SPG-F = VSPG* = VSPG*+*
2. *NFG ⊏ SPG-F.*
3. *PFG = FGCC*
4. *All balanced (totally balanced) games are monotonic (BG ⊏ MO and TBG ⊏ MO)*

4.5 Summary

In Figure 4.2 we summarize the existing relations between the different classes of coalitional games that we obtained so far. Note that our games are all non-negative, and therefore we only consider all classes of properties, e.g. monotonic games, balanced games and totally balanced games with respect to this condition.

We use Grey-shades and hatching to indicate what properties (balanced, totally balanced, monotonic) the different classes of coalitional games have. For example, a class of coalitional games that has the same colour as the class of monotonic games, precisely captures this class. If we also added an arrow, we mean that this class only captures a part of the intentionally defined class. For classes were no accurate claim can be made, we sketch what kind of games it does capture or does not capture.

Given that flow games and shortest path games are both based on graphs and therefore share several characteristics, we presented them in a hierarchy, which is based on the ownership relation. In level two (L2) we have games where a player can own several arcs (vertices), but an arc (vertex) is only owned by one player. In level one (L1) we have generalizations, where more than one player can own an arc (vertex) and finally in level 3 (L3) we have games that have a total bijective ownership function.

Having this hierarchy, we see immediately that both characteristics are quite influential in the case of flow games and shortest path games with a global reward scheme. The expressivity of games increases from (L3) to (L1). But on the other side, for shorest path games with an individual reward scheme, we do not have any notable effect. This indicates that in the context of shortest path games the reward scheme is an influential characteristic with respect to expressive power. In particular, we can see that despite the fact that *SPG-VG* (L1) has more general characteristics than *VSPG** (L2), *SPG-VG* is less expressive. Hence, the global reward scheme allows us to generate a larger class of coalitional games. This is quite surprising at first, because the individual reward schemes seems to allow much more freedom to generate games.

What is interesting as well, is the fact that shortest path games with a global reward scheme are not only more expressive than shortest path games with an individual reward scheme, but also more expressive compared to standard flow games (see layer L2).

Outside this hierarchy we indicated that market games, linear production games and flow games are equivalent. That means every game can be converted into the other representation types. The same is the case for pseudo-flow games, committee-flow games, *VSPG**, *VSPG*+* and *SPG-F*, which are indirectly connected to each other. This is the case, because they all capture precisely the class of monotonic game, and therefore can be transformed theoretically from one representation into another.

Chapter 5

Complexity Results for Graph-based Games

In the previous chapter we indicated that shortest path, flow, market and linear production games are similar types of coalitional games with respect to their expressive power. The equivalence proofs are constructive and could actually be used to transfer games from one game representation into another, but we have to be careful at this point to omit misinterpretations. This is the case, because the corresponding reductions of the equivalence proofs require exponential time. Hence, it does not make sense to use those particular equivalences to "transfer" complexity results between those classes. Nevertheless, this must not necessarily be a dead end. We think that it might be reasonable to look for polynomial time reductions between graph-based games, or maybe sub-classes of totally balanced games in the future. This might also give us some hints, how those games are related in a more technical sense. For now, we will concentrate on proving results for each class of coalitional game independently.

The results of the previous chapter did not contribute to solve complexity results for shortest path games to the degree, we originally hoped. But there are some minor, more indirect, gains with respect to complexity problems. Given that all flow games and shortest path games of type *SPG-VG*, *XSPG*, *XSPG**, etc. are totally balanced, it does not make sense to analyse the computational complexity to determine if the core of such a game is empty, whereas this question is reasonable to ask in the case of shortest path games having a global reward scheme, like *SPG-F*. Furthermore, we used some well-known results and properties from cooperative game theory to determine the computational complexity to compute an element of the core, the actual core and to determine if the core is empty for simple monotonic games.

Before we start to present complexity results for shortest path games in the next section, we want to demonstrate that the application of the characteristic function v of a shortest path games $\langle N, v\rangle$ for an arbitrary coalition S is polynomial: Let G^N be a shortest path game representation and $S \subseteq N$ an arbitrary coalition. We start by removing all arcs that are not owned by players in coalition S from the graph. Then we apply a polynomial time algorithm to compute the shortest path from So to Si in the reduced graph. If there is no path we immediately get $v_{G^N}(S) = 0$, otherwise we can easily compute the cost of the shortest path and subtract it from the reward. Hence, we have a polynomial time algorithm to compute $v_{G^N}(S)$ for each coalition $S \subseteq N$.

5.1 Complexity Problems

We now present all the different complexity problems that we are going to analyse in this chapter. We present them in a parametrized form to be able to use them for different game representations. For example, we introduce the X-NULL-PLAYER decision problem as the general null player problems, where X is an arbitrary game representation. So, if we want to consider this problem with respect to shortest path games of type *VSPG*, we refer to this particular problem as VSPG-NULL-PLAYER. On the other side, if we want to refer to the problem without a specific representation or over a range of representations we just use NULL-PLAYER.

We start with complexity results regarding properties of players in coalitional games:

Definition 5.1.1. X-NULL-PLAYER*: Given a game representation G^N of type X and a player $a_i \in N$, test whether a_i is a null-player in game v_{G^N}.*

Definition 5.1.2. X-VETO-PLAYER*: Given a game representation G^N of type X and a player $a_i \in N$, test whether a_i is a veto player in game v_{G^N}.*

Definition 5.1.3. X-DICTATOR: *Given a game representation G^N of type X and a player $a_i \in N$, test whether a_i is a dictator player in game v_{G^N}.*

We have the following decision, function and counting problems with respect to power indices:

Definition 5.1.4. X-SHAPLEY-SHUBIK-INDEX: *Given a game representation G^N of type X and a player $i \in N$, compute the Shapley-Shubik power index of player i in game v_{G^N}, $\varphi_i(v_{G^N})$.*

Definition 5.1.5. X-SHAPLEY-SHUBIK-VALUE: *Given a game representation G^N of type X and a player $i \in N$, compute the Shapley-Shubik value of player i in game v_{G^N}, $\kappa_i(v_{G^N})$.*

Definition 5.1.6. X-BANZHAF-INDEX: *Given a game representation G^N of type X and a player $i \in N$, compute the Banzhaf power index of player i in game v_{G^N}, $\beta_i(v_{G^N})$.*

Definition 5.1.7. X-BANZHAF-VALUE: *Given a game representation G^N of type X and a player $i \in N$, compute the Banzhaf value of player i in game v_{G^N}, $\eta_i(v_{G^N})$.*

Due to the fact that the denominator of the Shapley-Shubik index is fixed, the Shapley-Shubik index and the Shapley-Shubik value have the same computational complexity. The same holds for the Banzhaf index.

Definition 5.1.8. X-DEEGAN-PACKEL-INDEX: *Given a game representation G^N of type X and a player $i \in N$, compute the Deegan-Packel power index of player i in game v_{G^N}, $p_i(v_{G^N})$.*

Definition 5.1.9. X-PUBLIC-GOOD-INDEX: *Given a game representation G^N of type X and a player $i \in N$, compute the Public Good power index of player i in game v_{G^N}, $\delta_i(v_{G^N})$.*

We continue with the complexity problems of the core:

Definition 5.1.10. X-COREMEMBERSHIP*: Given a game representation G^N of type X and an allocation x, check whether $x \in Core(v_{G^N})$.*

Definition 5.1.11. X-ELEMENTSCORE*: Given a game representation G^N of type X, return a set X such that $X \subseteq Core(v_{G^N})$.*

Definition 5.1.12. X-EMPTYCORE*: Given a game representation G^N of type X, test whether $Core(v_{G^N}) = \emptyset$.*

We now introduce a decision problem about a special type of coalitions:

Definition 5.1.13. X-#MWC*: Given a game representation G^N of type X, return the number of minimal winning coalitions of game v_{G^N}.*

5.2 Complexity Results for (Monotonic) Simple Games

We first introduce a general result for monotonic simple games. We give a proof for this rather obvious result, because we think that it is quite instructive.

Proposition 5.2.1. *If $\langle N, v\rangle$ is a monotonic simple game such that for all coalitions $S \subseteq N$, $v(S)$ can be computed in polynomial time, then we can determine for any player $a_i \in N$ in polynomial time whether a_i is a veto-player in v.*

Proof. We have to show that there exists no coalition $S \subseteq N$ such that $v(S) = 1$ and $a_i \notin S$. Therefore it is enough to determine the value of $v(N \setminus \{a_i\})$:

- if $v(N \setminus \{a_i\}) = 0$, we can deduce from the fact that v is monotonic that for all $S \subset N \setminus \{a_i\}$, $v(S) = 0$. Slightly rephrased we have: for all $S \subseteq N$, $a_i \notin S \Rightarrow v(S) = 0$, and therefore a_i is a veto-player.
- Otherwise, if $v(N \setminus \{a_i\}) = 1$, then there is a profitable coalition not including a_i, thus a_i is not a veto-player.

Due to the fact that we can compute $v(N \setminus \{a_i\})$ in polynomial time, we can therefore also determine if a_i is a veto-player in polynomial time. □

Using the generalized definition of a veto-player for coalitional games, which we gave in Chapter 2, we can generalize the result above to hold for monotonic coalitional games as well.

We now analyse the computational complexity to determine if a player is a dictator in a monotonic simple game.

Corollary 5.2.2. *Let $\langle N, v\rangle$ be a monotonic simple game such that for all coalitions $S \subseteq N$, $v(S)$ can be computed in polynomial time and $a_i \in N$. We can determine in polynomial time if a_i is a dictator in v.*

Proof. We have to check two conditions, namely if

(i) $v(\{a_i\}) = 1$. Given that the game is monotonic, every coalition S such that $a_i \in S$ is a winning coalition ($v(S) = 1$). Hence, every coalition containing player a_i is a winning coalition.

(ii) a_i is a veto-player.

It is clearly the case that both conditions can be checked in polynomial time. □

An interesting result for power indices in specific simple games is the following:

Proposition 5.2.3. *For any reasonable representation of a simple game, a polynomial time algorithm to compute the Shapley-Shubik index implies a polynomial time algorithm to compute the Banzhaf index. ([3], Proposition 4.3 and 4.4)*

Now we analyse some complexity problems with respect to the core of simple monotonic games.

Proposition 5.2.4. *For monotonic simple games the decision problem* EMPTY-CORE *is in* $\mathcal{P}$.

Proof. We know by Theorem 2.2.14 that in a simple game the core is empty iff there is no veto-player. Now, given that we can check for every monotonic simple game in polynomial time if a player is a veto-player, we can also determine in polynomial time if there is at least one veto-player in N by testing all players. □

We can even do more, namely compute the elements of the core in polynomial time.

We have the following folk theorem: If there are no veto players in a simple game G, then the core is empty. Otherwise, let $a_{v_1}, ..., a_{v_m}$ be the veto players in G. Then the core is the set of imputations that distribute all the gains only to veto players: $Core(v) = \{\langle p_1, ..., p_n \rangle \mid \sum_{i=1}^{m} p_i = 1\}$. Details and a proof for this theorem can be found in a paper by Bachrach, Meir, Zuckerman and Rosenschein [5].

Hence, to compute the core we just have to compute all veto players, what can be easily done, because to check for a veto player is in $\mathcal{P}$. If there is no veto-player we have already shown that the core is empty, and otherwise the core contains every imputation such that the total value is 1 ($v(N) = 1$) and this value is distributed over all veto players, and only the veto players.

Proposition 5.2.5. *For any monotonic simple game v and imputation x we can check in polynomial time whether $x \in Core(v)$.*

We just have to compute all veto-players and then check if the imputation x fits into the pattern described by the folks theorem above.

Now, given that *TSPG, TXSPG*, *TNFG* and *SCG* are monotonic simple games, we just have to check that for an arbitrary coalition $S \subseteq N$ the value of the game can be computed in polynomial time. For shortest path games, we have Dijkstra's algorithm (see [25]), for flow games Ford and Fulkerson's algorithm (see [32]) and for spanning connectivity games Prim-Jarník algorithm (see [50]). Given that all these algorithm are polynomial, we can immediately apply all the results from above to these types of coalitional games.

5.3 Complexity Results for Shortest Path Games

In this section we analyse the different variants of shortest path games.

Shortest Path Game - Voorneveld Grahn (SPG-VG)

By Proposition 4.3.9, which also holds for *XSPG*, *XSPG** and *XSPG*+*, every shortest path game of type *SPG-VG*, *XSPG*, *XSPG** and *XSPG*+* has a non-empty core, and therefore it does not make sense to analyse the EMPTY-CORE decision problem. Nevertheless it might be interesting to analyse other decision problems related to the core, e.g. CORE-MEMBERSHIP or ELEMENTSCORE.

The following result is an immediate corollary of Proposition 4.3.9 and the corresponding extension we gave for condition: $c(N) = 0$ and $c(N) < r(N)$.

Corollary 5.3.1. *An element of the core of a shortest path game of type* SPG-VG *can be computed in polynomial time.*

Proof. The proof of Proposition 4.3.9 is based on the assumption that $0 < c(N) < r(N)$, but as we stated above, we assume that $0 \leq c(N) < r(N)$. So, we have to treat two different case:

Case 1: $(c(N) = 0)$ By assigning to each player his or her individual reward ($\forall i \in N$, $x_i = r_i$) we can easily generate an element of the core.

Case 2: $(c(N) > 0)$ In this case we can easily determine a core element in polynomial time using the result from above. We start with $x_1 = r_1$, $x_2 = r_2$,...,$x_i = r_i$ etc. until we assigned all utilities $(v(N))$. Then we just assign $x_j = 0$ for all j where $i < j \leq |N|$.

Hence, there is a polynomial algorithm to compute an element of the core. □

Given that the proof of the proposition above does not depend on the ownership function o or the condition that we have an acyclic graph, the result holds for *XSPG* and *XSPG** as well.

Value Shortest Path Games (VSPG, VSPG*, VSPG*+)

We start by analysing the computational complexity of properties like null-player and veto-player and then prove some results for different power indices.

Null-Player and Veto-Player

The corollary, which we are going to present next, depends on the following complexity result from Bachrach and Rosenschein [6], which is about null-players[1] in directed connectivity games.

Theorem 5.3.2. *Testing for null-players in directed connectivity games is co$\mathcal{NP}$*-complete

Corollary 5.3.3. VSPG-NULL-PLAYER *is co$\mathcal{NP}$*-complete

[1] Bachrach and Rosenschein refer to them as dummy-players in [6].

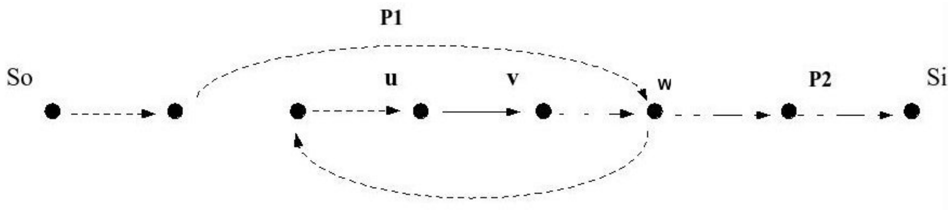

Figure 5.1: Counterexample: P_1 and P_2 intersect (DAG)

Proof. We can easily show that VSPG-NULL-PLAYER is in $co\mathcal{NP}$: Given a shortest path game v of type *VSPG* and a coalition $S \subseteq N$ such that $a_i \notin S$ we can test in polynomial time whether $v(S \cup \{a_i\}) - v(S) \neq 0$.

Given that we can model all directed connectivity games as shortest path games of type *VSPG*s (polynomial transformation) (see Proposition 4.3.3), we can immediately deduce from Theorem 5.3.2 that testing for null-players in *VSPGs* must be *$co\mathcal{NP}$-hard.*

Hence, VSPG-NULL-PLAYER is *$co\mathcal{NP}$-complete.* □

The same result follows immediately for *VSPG**, *VSPG*+*, *TSPG* and *TXSPG*.

Lemma 5.3.4. *To check if a player is a null-player in a shortest path game of type VSPG*(DAG) *is in* $\mathcal{P}$.

Proof. Let G^N be a shortest path game representation with a shortest path cooperative situation σ and the corresponding game v_{G^N}. Furthermore, let a_i be a player and $e = (u, v)$ an arc such that $o(e) = a_i$. We assume that $u \neq So$ and $v \neq Si$. Note that those special cases follow easily. We determine the shortest path from So to u and the shortest path from v to Si. Let's call the paths P_1 and P_2. This can be easily done using one of the available polynomial time algorithms to compute a shortest path in directed graphs. We have two main cases to consider:

Case 1: If both path exist, we define coalition $C := o(P_1) \cup o(P_2) \cup o(\{(u, v)\}$ with $R : So \to Si$ as the shortest path through e as $R := P_1 \circ e \circ P_2$ (where $\circ$ is the obvious path concatenation operator).

Claim 1: P_1 and P_2 are vertex-disjoint, i.e. they do not share vertices except So and Si.

Proof (Claim 1): Let's assume for the sake of a contradiction that this is not the case. So, there is a vertex w that is contained in both paths. Given that w is a vertex in P_1 and P_2, there must be a path from w to u and v to w respectively. Hence, there is a cycle $u \twoheadrightarrow v \twoheadrightarrow w \twoheadrightarrow u$ and therefore the assumption that the graph is an acyclic directed graph (DAG) is violated (see Figure 5.1).

Claim 2: There is no path $T \in P(C)$ such that $cost(T) < cost(R)$. Furthermore it is the case that $c(C \setminus \{a_i\}) = \infty$.

Proof (Claim 2): By definition of path R, it is clearly the case that there is no path $T \in P(C)$ such that $e \in T$ and $cost(T) < cost(R)$. Hence, if there should be some $T \in P(C)$ such that $cost(T) < cost(R)$, then it must be the case that $e \notin T$. So, let's assume that $e \notin T$. We now have to check if there is another path from So to Si in C. But given that P_1 and P_2 are vertex-disjoint (Claim 1), there does not even exist a path $T \in P(C \setminus \{a_i\})$ from So to Si. Hence, $\infty = cost(T) > cost(R)$ and $c(C \setminus \{a_i\}) = \infty$.

Now we have two sub-cases to consider:

Case 1.1: ($c(C) \geq r$). In this case all possible paths P through arc e automatically have the property $c(P) \geq r$. Hence, for all these paths P with the corresponding coalition $D = o(P) \subseteq N$ we have $v_{G^N}(D) = 0$ and after removing are e, there is no path anymore and therefore $v_{G^N}(D \setminus \{a_i\}) = 0$ as well. But these paths are the only possibility for player a_i to influence the value of the game. So, without participation in a profitable path, a_i must be a null player.

Case 1.2: ($c(C) < r$). So, $v_{G^N}(C) = r - c(C) > 0$ and by claim 2 we get $v_{G^N}(C \setminus \{a_i\}) = r - c(C \setminus \{a_i\}) = 0$. Hence, a_i is not a null player.

Case 2: If P_1 or P_2 does not exist, there is no path from So to Si that includes arc e. Hence, by including a_i to a coalition the value of v_{G^N} will not change. The same holds, when we remove it. Hence, a_i is a null player.

To sum it up: The shortest paths $P_1$and P_2 can be computed in polynomial time. Then it has to be check if both paths exist (Condition 1) and if this is the case yet another condition has to be check, namely if $c(C) \geq r$ (Condition 2). By Claim 2 we even know that $c(C) = cost(R)$ and therefore it is enough to check if $cost(R) = cost(P_1) + cost(e) + cost(P_2) \geq r$. All this can be done in polynomial time.

Hence, it can be tested in polynomial time if a_i is a null-player in v_{G^N}. □

We can easily adapt this proof for *TSPG*(DAG) by adding a third condition, namely a condition that tests if the threshold T has been reached. The same result, slightly more involved, but pursuing the same proof strategy follows for *VSPG**(DAG).

Lemma 5.3.5. *To check if a player is a null-player in a shortest path game of type VSPG**(DAG) *is in* $\mathcal{P}$.

Proof. The proof is similar to the proof for *VSPG*(DAG), why we only sketch it: Let G^N be a shortest path game representation with a shortest path cooperative situation σ and the corresponding game v_{G^N}. Let's assume without loss of generality that we have a player a_i that owns two arcs, namely $e_1 = (u_1, v_1)$ and $e_2(u_2, v_2)$.

We determine the shortest path from So to u_1 and the shortest path from v_1 to Si, and the same for u_2 and v_2. Let's call the paths P_1 and P_2, resp. P'_1 and P'_2. This can be easily done using one of the available polynomial time algorithms to compute a shortest path in directed graphs. If one of the paths does not exist, we can continue as for *VSPG*, because the corresponding arc (e_1 or e_2) cannot be used to contribute to a shortest path. So, from this point on, we assume that all paths P_1, P_2, P'_1, P'_2 exist.

We have three scenarios to consider:

1. $e_1$and e_2 are in a sequence: P_2 contains e_2 or P'_2 contains e_1
2. e_1 and e_2 are parallel: e_1 is not part of $P'_1 \circ e_2 \circ P'_2$ and e_2 is not part of $P_1 \circ e_1 \circ P_2$
3. e_1 and e_2 are such that $u_1 = u_2$ and $v_1 \neq v_2$.

The first scenario is proved similar to *VSPG*. For the second and third scenario we use the same proof strategy as for *VSPG*. We only sketch the proof for the second scenario.

Now, we define coalitions C_1 and C_2 as defined in the proof for *VSPG*. Let $R_1 : So \rightarrow Si$ be the shortest path through e_1 defined as $R_1 := P_1 \circ e_1 \circ P_2$ and (where $\circ$ is the obvious path concatenation operator), R_2 defined respectively.

We have four cases to consider:

Case 1: ($c(C_1) \geq r$ and $c(C_2) \geq r$). In this case all possible paths P through arc e_1 or e_2 automatically have the property $c(P) \geq r$. Hence, for all these paths P with the corresponding coalition $D = o(P) \subseteq N$ we have $v_{G^N}(D) = 0$ and after removing arc e_1 and e_2, there is no path anymore and therefore $v_{G^N}(D \setminus \{a_i\}) = 0$ as well. But these paths are the only possibility for player a_i to influence the value of the game. So, without participation in a profitable path, a_i must be a null player.

Case 2: ($c(C_1) < r$ and $c(C_2) \geq r$). Hence, $v_{G^N}(C_1) = r - c(C_1) > 0$ and by Claim 2 of $VSPG$ we get $v_{G^N}(C_1 \setminus \{a_i\}) = r - c(C_1 \setminus \{a_i\}) = 0$. So, we found a witness that a_i is not a null-player

Case 3: ($c(C_1) \geq r$ and $c(C_2) < r$) Same as case 2.

Case 4: ($c(C_1) < r$ and $c(C_2) < r$) Same as case 2.

So we can determine if a_i is a null player for the second scenario. The rest of the proof follows similar to $VSPG$. Now, taking all scenarios into account, we have shown that it can be determined in polynomial time if a_i is a null-player in a game v_{G^N} of type $VSPG*$. □

By adding an additional conditions to the proof for *VSPG**(DAG) above, it can be proved immediately that for games of type *VSPG*+*(DAG) the following holds.

Corollary 5.3.6. *To check if a player is a null-player in a shortest path game of type VSPG*+*(DAG) *is in* $\mathcal{P}$.

Now we turn to another complexity problem, namely the VSPG-VETO-PLAYER decision problem. This proof is similar to the proof of Proposition 5.2.1.

Proposition 5.3.7. VSPG-VETO-PLAYER *is in* $\mathcal{P}$.

Proof. Let a_i be a player, G^N a game representation of type *VSPG*, where e is the arc player a_i owns ($o(e) = a_i$). For player a_i to be a veto-player (extended interpretation), arc e must be part of every profitable path. So, we have to show that there exists no coalition $S \subseteq N$ with a profitable path such that $a_i \notin S$. Therefore it is enough to determine the value of $v_{G^N}(N \setminus \{a_i\})$.

- if $v_{G^N}(N \setminus \{a_i\}) = 0$, we can deduce from the fact that v_{G^N} is monotonic that for all $S \subset N \setminus \{a_i\}$, $v_{G^N}(S) = 0$. Slightly rephrased we have: for all $S \subseteq N$, $a_i \notin S \Rightarrow v_{G^N}(S) = 0$, and therefore a_i is a veto-player.
- Otherwise, if $v_{G^N}(N \setminus \{a_i\}) > 0$, then there is a profitable coalition not including a_i, thus a_i is not a veto-player.

Given that $v_{G^N}(N \setminus \{a_i\})$ can be determined in polynomial time, we have that VSPG-VETO-PLAYER is in $\mathcal{P}$. □

The same follows for *VSPG**, *VSPG*+* and also for *XSPG*, *XSPG**, *XSPG*+* and *SPG-VG*.

Due to the fact that *VSPG**(DAG) and *SPG-F* are similar (only differ in two tags, namely SOSI-SET/SOSI-VERTEX and OWNVERTEX/OWNARC) and by taking into account that these characteristics do not influence the computational complexity (see Section 3.6), the following result holds.

Proposition 5.3.8. *To check if a player is a null-player or a veto-player in a shortest path game of type* SPG-F *is in* $\mathcal{P}$.

Power Indices

After determining the computational complexity of the null-player problem, we can directly use Proposition 5.3.3 to determine the computational complexity of the Shapley-Shubik, Banzhaf, Deegan-Packel and Public good power index.

Lemma 5.3.9. *To compute* VSPG-SHAPLEY-SHUBIK-INDEX, VSPG-BANZHAF-INDEX, VSPG-DEEGAN-PACKEL-INDEX *and* VSPG-PUBLIC-GOOD-INDEX *is intractable.*

Proof. To compute the power index for a player is at least as difficult as testing for a null-player. This is the case because the Banzhaf, Shapley-Shubik, Deegan-Packel and Public Good power index of a player are zero iff a player is a null-player (see Proposition 2.3.6). So, if we would be able to compute any of the indices in polynomial time, we could answer the VSPG-NULL-PLAYER decision problem in polynomial time. But this leads to a contradiction, because VSPG-NULL-PLAYER is *co$\mathcal{NP}$-complete* by Proposition 5.3.3. □

The same result follows for *VSPG**, *VSPG*+* and *TSPG*.

Now we want to prove an even stronger result for VSPG-BANZHAF-INDEX: It is *#$\mathcal{P}$-complete* to determine the Banzhaf index for *VSPG*.

Remember that the Banzhaf index is defined as $\beta_i(N, v) = \frac{1}{2^{n-1}} \sum_{S \subseteq N \setminus \{i\}} [v(S \cup \{i\}) - v(S)] = \frac{1}{2^{n-1}} \#\{S \subseteq N \setminus \{i\} \mid v(S \cup \{i\}) - v(S) = 1\}$. For convenience we introduce the following notation: $\overline{\beta}_i = \{S \subseteq N \setminus \{i\} \mid v(S \cup \{i\}) - v(S) = 1\}$.

Theorem 5.3.10. VSPG-BANZHAF-INDEX *is* #$\mathcal{P}$-complete.

Proof. We reduce the S-T-CONNECTEDNESS problem [56], one of the standard and first problems known to be #P-complete, to our problem:

S-T-Connectedness
Instance: $G = \langle V, A \rangle$; $s, t \in V$
Question: Number of subgraphs of G in which there is a (directed) path from s to t

We first prove *#$\mathcal{P}$-hardness*:

We have an instance of S-T-CONNECTEDNESS, hence a graph $G = \langle V, A \rangle$ and two distinct vertices $s, t \in V$. We first add another vertex Si to the graph and another arc $a' = (t, Si)$. So, we have $G' = \langle V', A' \rangle$, where $V' = V \cup \{Si\}$ and $A' = A \cup \{a'\}$. Let $So = s$. Then we define the set of players $N = \{1, 2, ...\}$, where $|N| = |A'|$ and we assign to every player exactly one arc. We refer to the player owning arc a' as i'. So, let $\pi : N \setminus \{i'\} \rightarrow A$ be an arbitrary bijective mapping and define the ownership mapping o as follows: $o(a) = \begin{cases} i' & \text{if } a = a' \\ \pi(a) & \text{otherwise} \end{cases}$ for any $e \in A'$. We define a cost function that assigns cost 0 to every arc of G' and set $r := 1$. Let's call the corresponding *VSPG* game v. Note that this transformation takes only polynomial time. We show now that there is a direct correspondence of the following form:

Claim: $X \in \overline{\beta}_{i'}(N, v)$ iff subgraph H of G induced by $o^{-1}(X)$[2] is such that there exists a path P in H.

(⇒): Let $X \in \overline{\beta}_{i'}(N, v)$. By definition we have $X \subseteq N \setminus \{i'\}$, $v(X \cup \{i'\}) = 1$ and $v(X) = 0$. Hence, we can deduce from $v(X \cup \{i'\}) = 1$ and the definition of the shortest path game (VSPG) v that there is a path from $s = So$ to Si, and therefore also a path from s to t in the subgraph induced by $o^{-1}(X)$.

(⇐): We assume that H is a subgraph of G, where $S \subseteq A$ is the set of arcs of H, such that there exists a path in H. Given that i' is a veto-player of v and $i' \notin o^{-1}(S)$ we have $v(S) = 0$. Given that there is a path in S (from $s = So$ to t), we get a path So to Si by adding a' to the coalition S. Thus $v(S \cup \{i'\}) = 1$, and therefore $S \in \overline{\beta}_{i'}(N, v)$.

So, we have a bijection between the two sets, and therefore the cardinality of both sets is equal. Hence, if we were able to compute the Banzhaf index $\beta_{i'}(N, v)$ for player $i' \in N$ in polynomial time, we can multiply it with 2^{N-1} to get $\#\overline{\beta}_{i'}(N, v)$. This would give us the number of solutions for the S-T-CONNECTEDNESS instance.

Now we prove $\#\mathcal{P}$*-membership*:

The Banzhaf index of player $i \in N$ of a shortest path game (N, v) of type *VSPG* is $\beta_i(N, v)$. It is the proportion of all winning coalitions where i is critical, out of all winning coalitions that contain i. Let $S \subseteq N$ be any coalition, it can be checked in polynomial time whether $i \in S$, $v(S) = 1$ and $v(S \setminus \{i\}) = 0$. The last two conditions are polynomial by definition of shortest path games (based on Dijkstra's algorithm). Hence, it can be easily checked if $i \in S$ and if player i is critical for a coalition S and therefore $v(S) - v(S \setminus \{i\}) = 1$.

Due to the fact that we can construct a deterministic polynomial Turing machine M that tests if a player is critical in a coalition, as shown above, we can now construct a non-deterministic Turing machine M' that first non-deterministically chooses a coalition, under the conditions that i is in the coalition, and then tests if i is critical for that coalition. The number of accepting paths of M' is the number of coalitions that contain i where i is critical. Let $|N| = n$. As introduced above we denote the number of such accepting paths of M' as $\#\overline{\beta}_i(N, v)$. Then the Banzhaf power index of agent i is $\beta_i(N, v) = \frac{\#\overline{\beta}_i(N,v)}{2^{n-1}}$.

Calculating the numerator $\#\overline{\beta_i}(N, v)$ is according to Definition 2.4.1, a $\#\mathcal{P}$ problem. Since the denominator is constant (given a domain with n players), VSPG-BANZHAF-Index is in $\#\mathcal{P}$.

So, we have shown that VSPG-BANZHAF-INDEX is #P-complete. □

For *VSPG**, *VSPG*+* and *SPG-F* we immediately have that the Banzhaf value counting problems are $\#\mathcal{P}$*-complete* as well. This also holds for *TSPG* when we set the threshold to 1 ($T := 1$).

Shortest Path Game - Fragnelli (SPG-F)

Contrary to *SPG-VG*, the class of *SPG-F* contains games that are non balanced (have an empty core). Hence, it makes sense to determine the computational complexity of EMPTYCORE.

Our first attempt to determine the complexity of EMPTYCORE was based on Theorem 4.3.8, but we were not successful so far.

[2]Normally we speak of a graph being induced by a set of vertices, but we can also use arcs instead.

Threshold Variant of VSPG (TSPG)

Based on the result for TSPG-BANZHAF-INDEX and Proposition 5.2.3 we get the following result:

Corollary 5.3.11. TSPG-SHAPLEY-SHUBIK-INDEX *is* $\#\mathcal{P}$-complete.

Proof. Given that TSPG-BANZHAF-INDEX is $\#\mathcal{P}$*-complete* we just have to check that *TSPG* is a simple game with a reasonable representation. But this can be easily verified. So, by Proposition 5.2.3 we can immediately deduce that TSPG-SHAPLEY-SHUBIK-INDEX is $\#\mathcal{P}$*-complete.* □

The same holds for *TXSPG*.

Minimal Winning Coalitions

As stated in Chapter 2, minimal winning coalitions are an important kind of coalition. Expressed informally, they can be seen as limit cases that have maximal power, but minimal effort. Due to its significance, we want to determine the computational complexity to count the number of minimal winning coalitions in a shortest path game of type *TSPG*. This problem is also interesting with respect to several solution concepts, e.g the Deegan-Packel index and the Public good index, which are based on the number of minimal winning coalitions in a game.

We already introduced a formal notion of a minimal winning coalition (see Definition 2.2.9). Now we want to show how minimal winning coalitions look like in the context of *TSPG*? This is easy to see, because minimal winning coalitions are exactly all the (simple) paths P (having non repeating vertices) from So to Si such that $r - cost(P) \geq T$, thus all the profitable paths.

Theorem 5.3.12. *The counting problem* TSPG-#MWC *is* $\#\mathcal{P}$-complete.

Proof. We reduce the S-T-PATHS[3] problem [56], one of the standard problems known to be $\#\mathcal{P}$-*complete* to our decision problem.

S-T-Paths
Instance: $G = \langle V, A\rangle$; $s, t \in V$
Question: Number of paths from s to t that visit every vertex at most once.

We first prove $\#\mathcal{P}$*-hardness*:

We have an instance of S-T-PATHS, hence a graph $G = \langle V, A\rangle$ and two distinct vertices $s, t \in V$. We now define the set of players $N = \{1, 2, ...\}$, where $|N| = |A|$ and we assign to every player exactly one arc. We also define a cost function that assigns cost 0 to every arc of G and set $r := 1$ and $T := 1$. So, we have a game representation G^N of type *TSPG*. Note that this transformation takes only polynomial time. Now we have to show that there is a direct correspondence between the concept of a minimal winning coalition (MWC) and an s-t-path.

Claim: X is a MWC iff X is a s-t-path

(⇒): Let X be a MWC. Then, as stated previously, X must be a profitable (simple) path from s to t. Hence, X is an s-t-path.

[3]The decision problem heavily relies on the notion of a simple path (also referred to as self-avoiding walks). For many similar decision problems, leaving this property out, polynomial time algorithms have been found. The decision problem applies to both, directed and undirected graphs.

($\Leftarrow$): Let P be an s-t-path. So, we have a (simple) path from s to t. Given that we assigned a cost of 0 to all arcs, $cost(P) = 0$ and therefore $r - c(P) = 1 - 0 = 1 \geq T = 1$. Hence, $v_{G^N}(P) = 1$. Given that P is a path, we have $v_{G^N}(P') = 0$ for every $P' \subset P$, because $r - c(P') = 0 < T = 1$. Thus, by definition P is a minimal winning coalition.

Now we prove $\#\mathcal{P}$*-membership:*

Let $S \subseteq N$ be any coalition, it can be checked in polynomial time if S is a MWC as follows:

Step 1: We first have to check if there is a shortest path P from So to Si involving all players in S. This can be done by applying Dijkstra's algorithm to determine the shortest path P in S. Then we have to check if $o(P) = S$. Note that both steps can be done in polynomial time.

Step 2: If this is not the case, S cannot be a MWC. Otherwise we check if the corresponding path P is profitable, what can be done in polynomial time as well.

Step 3: If path P is not profitable, then S cannot be a MWC. Otherwise it must be a MWC, because $v_{G^N}(S) = 1$ and given that it is a path, the reduction of the coalition by any player $T \subset S$ will lead to an interrupted path, and therefore infinite costs. Hence, $v_{G^N}(T) = 0$.

Due to the fact that we can construct a deterministic polynomial Turing machine M that tests if a coalition $S \subseteq N$ is a MWC, as shown above, we can now construct a non-deterministic Turing machine M' that first non-deterministically chooses a coalition S and then tests if S is a MWC. The number of accepting paths of M' is then the number of MWC. Let's denote it by k. Now, according to the Definition 2.4.1,TSP-#MWC is in $\#\mathcal{P}$.

So, finally we have that TSPG-#MWC is $\#\mathcal{P}$*-complete*. $\square$

It can be easily deduced that not only $M(v)$, but also $M_i(v)$ is $\#\mathcal{P}$*-complete.*

Theorem 5.3.13. *The counting problem* TXSPG-#MWC *is* $\#\mathcal{P}$-complete.

Proof. The proof is similar to the one for *TSPG*. We have to introduce a new sink s with an arc from Si to s and a corresponding player i^*. Note that the addition of an extra arc will not influence the amount of minimal winning coalitions. Finally, let $r_{i^*} = 1$ and $r_i = 0$ for all $i \in N \setminus \{i^*\}$. $\square$

XSPG

Null-Player and Veto-Player

Given that *XSPG* does not directly model the class of connectivity games, we cannot use the same proof strategy as for *VSPG.*

Lemma 5.3.14. XSPG-NULL-PLAYER *is co*$\mathcal{NP}$-complete.

Proof. co$\mathcal{NP}$*-membership*:

We can easily show that XSPG-NULL-PLAYER is in *co*$\mathcal{NP}$: Given a shortest path game v of type *XSPG* and a coalition $S \subseteq N$ such that $a_i \notin S$, we can test in polynomial time whether $v(S \cup \{a_i\}) - v(S) \neq 0$.

co$\mathcal{NP}$*-hardness*:

We now reduce an instance of a directed connectivity game to a shortest path game of type *XSPG*. We know by Theorem 5.3.2 that it is *co$\mathcal{NP}$-complete* to determine if a player is a null player in a directed connectivity game.

Let $\langle N, v\rangle$ be a directed connectivity game with a directed graph $G = \langle V, A\rangle$ and two distinct vertices $s, t \in V$. Furthermore, let a be an arbitrary player from N. We introduce a new vertex t', which is the new sink, and a new arc $e' = (t, t')$. So $G' = \langle V', A'\rangle$, where $V' = V \cup \{t'\}$ and $A' = A \cup \{e'\}$ and $s, t' \in V'$ are source and sink. Then we introduce a new player i', and a new set of players $N' = N \cup \{i'\}$ ($|N'| = |A'|$). We keep the assignment of arcs of the directed connectivity game, assign the new arc e' to the new player i' and assign cost 0 to all arcs. Finally we define $r_{i'} := 1$ and $\forall i \in N \setminus \{i'\}\ r_i := 0$. Based on the shortest path game representation G^N, where $\sigma := \langle \Sigma, N', o, c, r\rangle$ is a shortest path cooperative situation and $\Sigma = \langle V', A', s, t'\rangle$ we can define a shortest path game $\langle N', v_{G^{N'}}\rangle$. Note that this transformation can be done in polynomial time.

Claim: a is not a null-player in $v_{G^{N'}}$ iff a is not a null-player in v

($\Rightarrow$): If a is not a null-player in $v_{G^{N'}}$, then there exists a coalition $S \subseteq N' \setminus \{a\}$ such that $v_{G^{N'}}(S \cup \{a\}) = 1$ and $v_{G^{N'}}(S) = 0$. So, there is a path from s to t' and $i' \in S$. Hence, there is a path from s to t in G and therefore $v(S \setminus \{i'\}) = 1$. Furthermore, we can deduce from $v_{G^{N'}}(S) = r(S) - c(S) = 0$ and the fact that $i' \in S$ that $c(S) > 0$ and that is only possible if there is no path. So, there is not path from s to t in graph G and therefore $v(S \setminus \{i'\}) = 0$. Thus, v is not a null-player.

($\Leftarrow$): If a is a null player in $v_{G^{N'}}$, then $\forall S \subseteq N' \setminus \{a\}\ v_{G^{N'}}(S \cup \{a\}) = v_{G^{N'}}(S)$. Let's assume for the sake of a contradiction that there is a coalition $T \subseteq N \setminus \{a\}$ such that $v(T \cup \{a\}) \neq v(T)$. Du to the fact that v is monotonic it follows immediately that $v(T \cup \{a\}) = 1$ and $v(T) = 0$. Now by adding i' to T we immediately get $v_{G^{N'}}(T \cup \{a\} \cup \{i'\}) = 1$ and $v_{G^{N'}}(T \cup \{i'\}) = 0$. So, $T' = T \cup \{i'\}$ leads to a contradiction.

Hence, XSPG-NULL-PLAYER is in *co$\mathcal{NP}$-complete.* □

The same clearly holds for *XSPG** and *XSPG*+*. If we now restrict the input to DAG, we can prove that XSPG-NULL-PLAYER can be solved in polynomial time.

Proposition 5.3.15. *To check if a player is a null-player in* XSPG*, restricting the input to cases where the graph is a directed acyclic graph (DAG), is in $\mathcal{P}$.*

Proof. Let a_i be an arbitrary player. We have two cases:

Case 1: ($r_i = 0$)

Under this assumption player a_i can only increase the value of a coalition he joins by offering cost improvements with respect to path routing, thus offering a shorter path. The proof is similar to the proof of *VSPG*: We first generate coalition C and given that $r_i = 0$, we have $r(C) = r(C \setminus \{a_i\})$. Hence, we have a fixed reward with respect to the coalitions we are interested in and can therefore continue as in the proof of *VSPG*.

Case 2: ($r_i > 0$)

We first check if there is a coalition $S \subseteq N$ such that $v(S) > 0$. This can be easily done by determining the shortest path for the grand coalition N, calculating the cost of the shortest path and subtracting it from $r(N)$. If $v(N) = 0$, then we can deduce from the fact that v is monotonic that $v(S) = 0$ for all $S \subseteq N$, and therefore a_i is clearly a null-player. If this is not the case, then we can reason as follows: For any coalition $S \subseteq N$ ($v(S) > 0$) containing a_i, if we remove a_i from the coalition, the coalition's value decreases due to the fact that r_i is subtracted and furthermore given that the routing can clearly not improve with one arc removed, we have $v(S \setminus \{a_i\}) < v(S)$. Now, if we assume that there is at least one effective path, we know that $v(N) > 0$ must hold for the grand coalition N and therefore by the deduction step above, a_i is not a null-player. □

We can easily adapt this proof for *TXSPG*(DAG) by adding a third condition, namely a condition that tests if the threshold T has been reached. This condition can clearly be checked in polynomial time. The same result, slightly more involved, but pursuing the same proof strategy follows for *XSPG**, *XSPG**+ and *SPG-VG*, where the input is reduced to DAG.

Banzhaf and Shapley-Shubik Index

After determining the computational complexity of the null-player problem for shortest path games of type *XSPG*, we can directly use Proposition 5.3.3 to determine the computational complexity of the power indices: Shapley-Shubik, Banzhaf, Deegan-Packel and Public Good.

Lemma 5.3.16. *To compute* XSPG-SHAPLEY-SHUBIK-INDEX, XSPG-BANZHAF-INDEX, XSPG-DEEGAN-PACKEL-INDEX *and* XSPG-PUBLIC-GOOD-INDEX *is intractable.*

The proof is exactly the same as we used for Lemma 5.3.9 . Note that the same result follows for *XSPG**, *XSPG**+ and *TXSPG*.

In the case of the Banzhaf index we can even prove more.

Theorem 5.3.17. XSPG-BANZHAF-INDEX *is* $\#\mathcal{P}$-complete.

Proof. The proof is similar to the proof of *VSPG*. The only difference is the translation of the S-T-CONNECTEDNESS instance to a *XSPG* game:

We have an instance of S-T-CONNECTEDNESS, hence a graph $G = \langle V, A\rangle$ and two distinct vertices $s, t \in V$. We first add another vertex Si to the graph and another arc $a' = (t, Si)$. So, we have $G' = \langle V', A'\rangle$, where $V' = V \cup \{Si\}$ and $A' = A \cup \{a'\}$. Then we define the set of players $N = \{1, 2, ...\}$, where $|N| = |A'|$ and we assign to every player exactly one arc. We refer to the player owning a' as i'. We also define a cost function that assigns cost 0 to every arc of G' and set the individual rewards, namely $r_{i'} := 1$ and $r_e := 0$ for all $e \in N \setminus \{i'\}$. Let's call the corresponding *XSPG* game v. This transformation takes only polynomial time.

For the rest of the proof we can mimic the proof for *VSPG*. □

So, for all other, more general coalitional games, *XSPG**, *XSPG**+ and *SPG-VG* we have that the Banzhaf-value is $\#\mathcal{P}$*-complete*. It also holds for *TXSPG* $(T := 1)$

5.4 Complexity Results for Related Games

Here we given an overview of results for graph-based games that we encountered in the literature.

Network flow game

Bachrach and Rosenschein [6] proved the following: TNFG-NULL-PLAYER is *co$\mathcal{NP}$-complete.* It stays *co$\mathcal{NP}$-complete* when the input is restricted to acyclic directed graphs (TNFG-NULL-PLAYER(DAG)). For CNFG-NULL-PLAYER it is *co$\mathcal{NP}$-complete* as well, whereas CNFG-NULL-PLAYER(DAG) is in $\mathcal{P}$. Furthermore TNFG-BANZHAF-INDEX is *$\#\mathcal{P}$-complete* and TNFG-VETO-PLAYER is in $\mathcal{P}$ and therefore all problems regarding the core as well.

Based on these results, Aziz, Lachish, Paterson and Savani [2] showed that TSPG-SHAPLEY-SHUBIK-INDEX is *$\#\mathcal{P}$-complete.*

Vertex Connectivity Game

Bachrach and Rosenschein [7] proved the following: VCG-VETO-PLAYER is in $\mathcal{P}$, VCG-BANZHAF-INDEX is *$\#\mathcal{P}$-complete* and VCG-BANZHAF-INDEX(TREE) is in $\mathcal{P}$.

Based on these results, Aziz et al. [2] showed that VCG-SHAPLEY-SHUBIK-INDEX is *$\#\mathcal{P}$-complete.*

Minimum Cost Spanning Tree games

Megiddo proved in [45] that MCSTG-SHAPLEY-SHUBIK-INDEX is in $\mathcal{P}$. Bird [11] and Granot and Huberman [36] showed that minimum cost spanning tree games are balanced (have a non-empty core). Faigle, Fekete, Hochstätter and Kern [29] proved that MCSTG-COREMEMBERSHIP is *co$\mathcal{NP}$-complete* .

Spanning Connectivity Games:

Aziz et al. [2] proved the following results:

Complexity Problems	**Input**	**Complexity**
SCG-SHAPLEY-SHUBIK-INDEX	Multigraph, simple graphs	*$\#\mathcal{P}$-complete*
SCG-BANZHAF-INDEX	Multigraph, simple graphs	*$\#\mathcal{P}$-complete*
SCG-BANZHAF-INDEX	Multigraph with bounded treewidth	$\mathcal{P}$
SCG-PUBLIC-GOOD-INDEX	Multigraph	$\mathcal{P}$
SCG-DEEGAN-PACKEL-INDEX	Multigraph	$\mathcal{P}$

Chapter 6

Interpretation and Discussion of Results

As mentioned in the introduction, our goal in this thesis is to analyse the influence of different characteristics on the computational complexity of solution concepts applied to shortest path games, as well as on the expressive power of games. Furthermore, we want to compare different types of graph-based coalitional games with respect to the computational complexity of solution concepts. For this purpose we have chosen the class of shortest path games, which is a rather interesting class on its own and furthermore similar to existing graph-based coalitional games, like network flow games, vertex connectivity games, etc., which have already been analysed with respect to computational complexity of solution concepts.

In the previous chapter we proved results for different variants of shortest path games and presented results for various graph-based games found in the literature. Based on these results, we can now start to look-out for patterns regarding the computational complexity of solution concepts applied to graph-based games. But we have to be careful to avoid misinterpretations or overly general claims regarding potential patterns. For example, a tempting but unreasonable endeavour in this research area would be to classify solution concepts by their complexity as mentioned by Deng and Fang [23]. The reason for this is that very often solution concepts may display different orders in the complexity hierarchy from game representation to game representation. Some concepts may, very well, be easier to compute in one coalitional game but more difficult in another coalitional game.

Taking this into account and by exercising the necessary care when interpreting the results, we nevertheless think that especially for restricted classes of coalitional games, like graph-based games, interesting patterns and indicators for similar complexity-theoretic results might emerge.

In this chapter we proceed as follows:

1. We present our main results, namely properties and complexity results for shortest path games, point out interesting observations and suggest interpretations for these observations.

2. We summarize and compare results of various graph-based games. We look for indicators with respect to open problems and present and comment on some of our observations.

6.1 Results for Shortest Path Games

We were successful in proving the computational complexity of several player-based properties, as well as power indices applied to different variants of shortest path games. But we failed in determining the complexity of the core for most cases. Now, we summarize all the complexity results that we gained in Chapter 5 (see Table 6.1). The notions and symbols that we used in the table have the following interpretation:

- Complexity classes in bold print are results, which have been proved in the previous chapter. If there are two results separated by "|", we want to indicate that we have results for different kinds of complexity problems (e.g. decision, function or counting) or a result for a restricted class of problems (e.g. DAG or Tree).
- If a field in the table contains "-", then the property or result either cannot be applied for this game or it does not make sense to apply it.
- We refer to an open problem by "?". If we have a complexity class (not bold) with a "?" attached to it, then we conjecture that this problem is an element of this particular complexity class.

We now give an interpretation of the results presented in Table 6.1. As can be seen immediately, there is no difference in the computational complexity of power indices between those variants of shortest path games, where the global reward scheme has been used (top table). The same is the case for shortest path games having an individual reward scheme (bottom table). Furthermore, contrary to our expectations, the reward scheme has no effect on the computational complexity of power indices. This can be deduced by comparing the columns of the first and second table one by one.

We can generally say that for all the different variants of shortest path games, which we considered, we have quite robust complexity results for power indices. The reason for this is that power indices applied to very basic variants of shortest path games, as well as the determination of player-based properties for these games is already intractable. Hence, the characteristics that we have considered so far cannot have any real influence on the computational complexity of power indices for shortest path games. But despite the results for power indices and various player-based properties, there might still be some effects in the case of stability concepts, which we have not considered yet.

An interesting fact is that restricting shortest path games to directed acyclic graphs (DAG), a restriction that we have not considered in a systematic way, had some influence on the computational complexity of the NULL-PLAYER problem. This suggests that it is necessary to look for characteristics and options of characteristics that have much more influence on shortest path games. We will comment on some promising characteristics and options, which considerably simplify graph-based games with respect to computational issues of solution concepts in the following section.

Global reward scheme

	VSPG	VSPG*	VSPG+*	TSPG	SPG-F
Shap-Shub-index	**NP-hard**	**NP-hard**	**NP-hard**	**NP-hard | #P-complete**	#P-complete ?
Banzhaf-index	**NP-hard | #P-complete**	**NP-hard | #P-complete**	**NP-hard | #P-complete**	**NP-hard | #P-complete**	**#P-complete**
D-P index	-	-	-	**NP-hard**	-
Holler index	-	-	-	**NP-hard**	-
Null-player	**CoNP-complete | P** (DAG)	**CoNP-complete | P** (DAG)	**CoNP-complete | P** (DAG)	**CoNP-complete | P** (DAG)	**P**
Veto-player	**P**	**P**	**P**	**P**	**P**
Dictator	-	-	-	**P**	-
MWC	-	-	-	**#P-complete**	-
EmptyCore	CoNP-complete ?	CoNP-complete ?	CoNP-complete ?	**P**	CoNP-complete ?
CoreMembership	CoNP-complete ?	CoNP-complete ?	CoNP-complete ?	**P**	CoNP-complete ?
ElementCore	?	?	?	**P**	?
Monotonic	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**
MO=SPG?	**No**	**Yes**	**Yes**	**No**	**Yes**
Totally Balanced	**No**	**No**	**No**	**No**	**No**

Individual reward scheme

	XSPG	XSPG*	XSPG+*	TXSPG	SPG-VG
Shap-Shub-index	**NP-hard**	**NP-hard**	**NP-hard**	**NP-hard | #P-Complete**	#P-complete ?
Banzhaf-index	**NP-hard | #P-complete**	**NP-hard | #P-complete**	**NP-hard | #P-complete**	**NP-hard | #P-Complete**	**#P-complete**
D-P index	-	-	-	**NP-hard**	-
Holler index	-	-	-	**NP-hard**	-
Null-player	**CoNP-complete | P** (DAG)	**CoNP-complete | P** (DAG)	**CoNP-complete | P** (DAG)	**CoNP-complete | P** (DAG)	**P**
Veto-player	**P**	**P**	**P**	**P**	**P**
Dictator	-	-	-	**P**	-
MWC	-	-	-	**#P-complete**	-
EmptyCore	-	-	-	**P**	-
CoreMembership	CoNP-complete ?	CoNP-complete ?	CoNP-complete ?	**P**	CoNP-complete ?
ElementCore	**P**	**P**	**P**	**P**	**P**
Monotonic	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**
MO=SPG?	**No**	**No**	**No**	**No**	**No**
Totally Balanced	**Yes**	**Yes**	**Yes**	**No**	**Yes**

Table 6.1: Complexity Table - Variants of Shortest Path Games

Contrary to our success to prove results for power indices and player-based properties, we had only limited success in determining the computational complexity of problems related to the core. So, an interesting consideration at this point would be the following: Is there a complexity-theoretic relationship between different complexity problems regarding the core? A positive answer to this question would allow us to easily extend our results for the core in some cases. But this is unfortunately not the case. At a first glance, testing non-emptiness of the core, checking membership of the core and finding a member of the core seems closely related, but we have to be careful with conclusions, because they may in general possess different complexities (see [23]). We have four examples, namely two "positive" and two "negative" ones:

- Deng and Papadimitriou [24] showed that for the class of weighted graph games the decision problem CORE-MEMBERSHIP is *co$\mathcal{NP}$-complete* and EMPTY-CORE is *$\mathcal{NP}$-complete*.
- They also showed that for weighed voting games CORE-MEMBERSHIP and EMPTY-CORE are polynomial.
- Fang et al. [30] showed that for flow games ELEMENTCORE is polynomial, but COREMEMBERSHIP is *co$\mathcal{NP}$-complete*.
- For linear production games ELEMENTCORE is polynomial and COREMEMBERSHIP is *co$\mathcal{NP}$-complete* [30].

Hence, we cannot rely blindly on similarities between complexity problems when results for the core have to be determined. But we might want to take it as a sort of hint to have a starting point for what we might want to prove. For example, both decision problems, CORE-MEMBERSHIP and EMPTY-CORE, seem to be related in a stronger sense for many graph-based games, which is interesting when the result for one problem has already been determined, but not for the other.

Another strategy would be to study results for related games, e.g. graph-based games. We now give an overview of results for flow games and minimum cost spanning tree games. Both graph-based games are similar to shortest path games and therefore interesting to consider.

- As stated above, for flow games (type CNFG) ELEMENTCORE is polynomial and COREMEMBERSHIP is *co$\mathcal{NP}$-complete* [30].
- For minimum cost spanning tree games (MCSTG), that can also be expressed as a generalized linear production game [35], ELEMENTCORE is polynomial [36] and COREMEMBERSHIP is *co$\mathcal{NP}$-complete* [29].

Again, we cannot blindly rely on results of related games when analysing the computational complexity of a problem, because minor differences in the definition of a game can sometimes strongly influence the result for particular solution concepts. Nevertheless, there seems to be a strong similarity between graph-based games and the complexity of solution concepts in general, as we will indicate in the next section. So, we should not dismiss these similarities entirely, because they can give us a first indication what the result might be. So, we think that it would be reasonable to check if COREMEMBERSHIP is *co$\mathcal{NP}$-complete* for shortest path games. Furthermore, we pursue the goal to prove that EMPTYCORE is computationally intractable for shortest path games having a global reward scheme.

The complexity-theoretic results for power indices that we gained so far might also suggest that the basic outline of a game, here graph-based games, has a huge impact on the computational complexity of solution concepts and properties in general. We think that it would be interesting to compare different graph-based games to investigate this further. This might allow us to see if these results are similar for many graph-based games or if shortest path games are an exception and most characteristics that we considered for shortest path games might, very well, have an effect on other graph-based games.

6.2 Case Study of Graph-based Coalitional Games

As indicated before, we have chosen shortest path games also for the reason to extend the sample space of complexity-theoretic results for graph-based coalitional games. Based on this extended corpus of graph-based games, we can now look for differences in the computational complexity of solution concepts over different graph-based games. We observed in the previous section that for shortest path games the complexity results of power indices are mostly intractable and invariant with respect to changes of characteristics. So, it might be interesting to check, if the intractability of power indices also extends over graph-based games, which share similar characteristics. This might give us some indications about complexity results of graph-based games in general.

This time we are not particularly interested in results for different characteristics, but more in the computational complexity of solution concepts over different graph-based games, which share basic properties. Having an overview of results for several graph-based games, we also hope to obtain some indications, which kind of results we might expect for open complexity problems of graph-based games. Due to the fact that we cannot predict how much influence the characteristics of graph-based games really have for different graph-based coalitional games, we decided to minimize the influence of those characteristics by comparing games, which share basic properties. Note that we distinguish between non-simple and simple graph-based coalitional games.

In Table 6.2 we list complexity results for all those graph-based coalitional games, which were introduced in Chapter 5. The results for *VSPG*, *XSPG*, *TSPG* and *TXSPG* were proved in this thesis, whereas the rest of the results were taken from the literature (see Section 5.4) or they are immediate corrolaries of results that we proved in Section 5.2. The notions and symbols that we used in the table have the following interpretation:

- Complexity classes in bold print are results, which have been proved in this thesis or the literature. If there are two results separated by "|", we want to indicate that we have results for different kinds of complexity problems (e.g. decision, function or counting) or a result for a restricted problem (e.g. DAG or Tree).
- If a field in the table contains "-", then the property or result either cannot be applied for this game or it does not make sense to apply it.
- We refer to an open problem by "?". If we have a complexity class (not bold) with a "?" attached to it, then we conjecture that this problem is an element of this particular complexity class.

Non-simple games

	VSPG	XSPG	CNFG	MCSTG
Shap-Shub-index	NP-hard	NP-hard	NP-hard	NP-hard ? | P (tree)
Banzhaf-index	NP-hard | #P-complete	NP-hard | #P-complete	NP-hard	NP-hard ?
Null-player	CoNP-complete | P (DAG)	CoNP-complete | P (DAG)	CoNP-complete | P (DAG)	?
Veto-player	P	P	P	P
EmptyCore	?	-	?	-
CoreMembership	CoNP-complete ?	CoNP-complete ?	CoNP-complete	CoNP-complete
ElementCore	?	P	?	P

Simple games

	TSPG	TXSPG	TNFG	VCG	SCG
Shap-Shub-index	NP-hard | #P-complete	NP-hard | #P-complete	NP-hard | #P-complete	#P-complete	#P-complete
Banzhaf-index	NP-hard | #P-complete	NP-hard | #P-Complete	NP-hard | #P-complete	#P-complete | P (Tree)	#P-complete | P (tree-bound)
D-P index	NP-hard	NP-hard	NP-hard	?	P
Holler index	NP-hard	NP-hard	NP-hard	?	P
Null-player	CoNP-complete | P (DAG)	CoNP-complete | P (DAG)	CoNP-complete (DAG)	?	?
Veto-player	P	P	P	P	P
Dictator	P	P	P	P	P
MWC	#P-complete	#P-complete	#P-complete ?	?	?
EmptyCore	P	P	P	P	P
CoreMembership	P	P	P	P	P
ElementCore	P	P	P	P	P

Table 6.2: Complexity Table - Graph-based Coalitional Games

A first observation (see Table 6.2) is that the computational complexity of power indices applied to those graph-based coalitional games, which we considered in the previous chapter, is pretty robust.

We will now analyse this in more detail: We start by comparing the following monotonic simple games: *TSPG*, *TXSPG*, *TNFG*, *VCG* and *SCG*. The reader should be aware that the results for monotonic simple games, like the complexity to determine a veto-player, dictator and the results regarding the core all follow immediately from the property that these coalitional games v are simple and monotonic and $v(S)$ can be computed in polynomial time for any coalition $S \subseteq N$. So, we ignore these results in our interpretation. On the other side, the results for power indices are quite interesting.

For all graph-based games that we considered in this thesis, it is $\#\mathcal{P}$*-complete* to determine the Shapley-Shubik and Banzhaf power index. We can also observe that flow games of type *TNFG* and threshold versions of shortest path games *TSPG* and *TXSPG* are very similar in all the results that we determined so far. What is quite interesting is the fact that they differ only in one aspect namely to determine the null-player for acyclic directed graphs ($co\mathcal{NP}$*-complete* vs $\mathcal{P}$), whereas there is no difference between non-simple games of similar construction *VSPG*, *XSPG* and *CNFG*. What is surprising as well is the fact that contrary to the other games, the Deegan-Packel and Public Good power index are computable in polynomial time for *SCG*.

Of course, when we talk about exceptions in this context, we have to keep in mind that for a given game representation, some solution concept might be easy to compute because the representation is ideal for the computation of this particular solution concept. So, it is not too unlikely to find aberrant results over similar games and therefore we should not misjudge the occurrence of exceptions. Nevertheless, we think that it is worthwhile to point out these exceptions.

Given that many of the non-simple games: *VSPG*, *XSPG*, *CNFG* and *MCSTG* have similar results as well, we think that it is indeed interesting to continue our work to show that graph-based games have rather similar complexity results with respect to power indices. Due to the similarities of graph-based games and already existing results, we conjecture that the following holds:

- For *MCSTG* it is intractable to determine the Shapley-Shubik and Banzhaf index.
- For *VSPG* and *XSPG* CoreMembership is computationally intractable

Up to this point, we solely made observations and some assumptions regarding the similarity of graph-based games. To put our assumptions on a more sound base, it would be necessary to analyse more graph-based games and complete the complexity results for the games we considered already, especially with respect to stability concepts.

Apart from comparing results over graph-based games, we also want to mention the following observation. We have noticed that "minor" changes in characteristics do not seem to have much effect on the computational complexity of power indices applied to graph-based coalitional games. But more "severe" modifications on the other side, like restricting the graphs to directed acyclic graphs, trees, etc., influence notably the computational complexity of various solution concepts. We have the following examples:

- The Shapley-Shubik power index can be computed in polynomial time for *MCSTG* with graphs reduced to trees.
- The Banzhaf power index can be computed in polynomial time for *VCG* with graphs reduced to trees.

- The Banzhaf power index can be computed in polynomial time for *SCG* using the notion of bounded treewidth.

Hence, it could be rewarding to analyse these stronger characteristics to find variants of graph-based coalitional games, where solution concepts are tractable, rather than intractable.

Chapter 7

Conclusion

In this thesis we concentrated on two things: We analysed the influence of different characteristics on the computational complexity of solution concepts, as well as the expressive power of graph-based games. Having proved various complexity-theoretic results for shortest path games, we extended the corpus of graph-based games, which have been analysed with respect to computational complexity of solution concepts. So, this allowed us to relate complexity-theoretic results for various graph-based coalitional games and look for potential complexity-theoretic patterns or indications for open problems in the context of graph-based games.

We now give a detailed account of our results, point out some of our main contributions and finally propose possible directions for future work.

7.1 Summary

At first we discussed which characteristics are interesting in the context of graph-based coalitional games, and especially for shortest path games, with respect to computational issues and expressivity of games. Then we introduced several variants of shortest path games, which vary over those characteristics and are based on the original models for shortest path games introduced by Voorneveld and Grahn [58] and Fragnelli et. al. [33]. We analysed the expressive power of the different variants of shortest path games, compared them and related them to other types of coalitional games. It was interesting to see what effect some of the characteristics, e.g. the reward scheme and the ownership relation had on the expressive power of some of the games. We concluded that some of the characteristics are quite useful to influence the expressive power of coalitional games.

We also mentioned in Chapter 4 that there is a direct relation between flow games, linear production games and market games. This is a quite interesting, because games can be directly translated from one game representation into another, but the results cannot be used to analyse the computational complexity of shortest path games, flow games and related coalitional games, as one might have hoped. Nevertheless, many of the properties we proved in Chapter 4 were useful to prove complexity-theoretic results for shortest path games or dismiss complexity problems if not applicable.

We continued by proving several complexity results in the context of shortest path games. Particular interesting are the following results:

- We proved that it is *co*$\mathcal{NP}$*-complete* to determine if a player is a null player for shortest path games of type *VSPG*. This result is then used to show that all power indices that we considered are intractable.
- We showed that it is $\#\mathcal{P}$*-complete* to determine the Banzhaf index for shortest path games of type *VSPG*.
- We demonstrated that the null player problem for shortest path games of type *VSPG*, where the input is restricted to directed acyclic graphs, is computable in polynomial time.
- We proved that it is $\#\mathcal{P}$*-complete* to determine all minimal winning coalitions of a shortest path game of type *TSPG*.

Apart from our technical results, we made some interesting observations, namely that most of the analysed complexity problems are computationally hard to solve and therefore most characteristics that we considered for shortest path games have no notable influence on the computational complexity of power indices. A reason for this is that basic shortest path games, with minimal properties, already lead to situations where solution concepts are hard to compute. So, it is impossible that the considered characteristics, which basically add more general properties to shortest path games, can influence the complexity-theoretic results.

This suggests that it would be reasonable to look for characteristics and options of characteristics that simplify those basic variants of shortest path games even further. A candidate for such a simplification seems to be the reduction of directed graphs to directed acyclic graphs. So, it could be interesting to analyse effects of DAG on more complexity-theoretic problems of shortest path games, and graph-based games in general. When we reviewed results for different graph-based games in the literature, we also encountered another promising characteristic: By restricting the underlying graphs of games to a tree, many intractable problems, become actually tractable.

Remember, shortest path games were originally meant as a positive example to show that it is worthwhile to consider different characteristics of games when the computational complexity of solution concepts for a coalitional game should be determined. So far, we were not successful, but it should be taken into account that we have not checked for most stability concepts yet. Hence, a general judgement about the influence of characteristics of graph-based games, including all solution concepts, is not possible at this point, because the results for power indices and stability concepts can be rather different. An example for two coalitional games where the results for stability concepts and power indices are quite different, are weighted graph games (see [24] for results) or weighted voting games (see [24, 49, 43, 44] for results).

Encouraged by the observation that complexity results for power indices are quite robust over many characteristics for shortest path games, we compared the complexity results for various graph-based games in a case study to see if they are similar. We were interested to investigate, if there is a tendency for graph-based games to be computationally hard for power indices. Comparing the results, we noticed indeed that graph-based coalitional games seem to share a lot of complexity problems, especially with respect to power indices and player-based properties. As mentioned in the previous chapter, we have to be cautious when interpreting results over different game representations, because they might vary from game to game and there might be exceptions. Nevertheless, we think that there is a tendency that these kinds of problems are very hard for standard graph-based games. Based on this assumption of similarity between graph-based games we also suggested outcomes for several open problems.

7.2 Future Work

At this point, we would like to give an overview of possible future directions of research and ideas that could be pursued to expand the work of this thesis.

To obtain a more precise idea of the influence of characteristics on the computational complexity of solution concepts applied to shortest path games, we think that it would be particularly interesting to analyse various stability concepts for different non-simple variants of shortest path games. Especially, the core, least-core and nucleolus would be important stability concepts to analyse in this context.

We also think that it would be interesting to prove several open problems for those graph-based games, which we presented in this thesis, and to look for further graph-based coalitional games to extend the sample space of graph-based games. So, having a more complete set of results we might be able to isolate influential properties and characteristics of graph-based games. This may even allow researchers, interested in transferring problems to graph-based coalitional games, to use it as heuristic to specify coalitional games in such a way that the application of interesting solution concepts is computationally tractable.

We observed during our studies that for many real-world motivated graph-based games, which have been analysed in the literature, the determination of power indices is intractable. For practical applications, where the existence of an efficient algorithm is imperative, this situation is of course unsatisfactory. But we have also noticed during our research that particular simplifications of graph-based games, most notably simplifications of the underlying graph, can lead to computationally favourable results. For example, the reduction of graphs to trees seemed to be a promising reduction. This is also interesting in a more practical context, because there are many real-world problems with tree-structures in computer science (Internet and networking), which could be analysed from a game-theoretic perspective. Hence, the reduction to trees and also acyclic directed graphs for graph-based games is not only a theoretical consideration to obtain polynomial complexity results, but also a promising way to determine classes of coalitional games, where interesting solution concepts applied to games, which are motivated from real-word problems, are tractable.

Bibliography

[1] Haris Aziz. Complexity of comparison of influence of players in simple games. *CoRR*, abs/0809.0519, 2008.

[2] Haris Aziz, Oded Lachish, Mike Paterson, and Rahul Savani. Power indices in spanning connectivity games. In *AAIM*, pages 55–67, 2009.

[3] Haris Aziz, Oded Lachish, Mike Paterson, and Rahul Savani. Spanning connectivity games. *CoRR*, abs/0906.3643, 2009.

[4] Haris Aziz and Mike Paterson. Computing voting power in easy weighted voting games. *CoRR*, abs/0811.2497, 2008.

[5] Yoram Bachrach, Reshef Meir, Michael Zuckerman, Jörg Rothe, and Jeffrey S. Rosenschein. The cost of stability in weighted voting games. In *AAMAS '09: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, pages 1289–1290. International Foundation for Autonomous Agents and Multiagent Systems, 2009.

[6] Yoram Bachrach and Jeffrey S. Rosenschein. Power in threshold network flow games. *Autonomous Agents and Multi-Agent Systems*, 18(1):106–132, 2009.

[7] Yoram Bachrach, Jeffrey S. Rosenschein, and Ely Porat. Power and stability in connectivity games. In *AAMAS '08: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems*, pages 999–1006, Richland, SC, 2008. International Foundation for Autonomous Agents and Multiagent Systems.

[8] John F. Banzhaf. Weighted voting doesn't work: A mathematical analysis. *Rutgers Law Review*, 19(2):317–343, 1965.

[9] S. Barberá, W. Bossert, and P. K. Pattanaik. Handbook of utility theory. volume ii: Extensions. chapter Utility Theory in Cooperative Games, pages 1065–1098. Kluwer Academic Publishers, October 2004.

[10] J. M. Bilbao, J. R. Fernández, and J. J. López. Complexity in cooperative game theory.

[11] C. J. Bird. On cost allocation for a spanning tree: A game theoretic approach. *Networks*, 3:335–350, 1976.

[12] Olga N. Bondareva. Some applications of linear programming methods to the theory of cooperative games (in russian). *Problemy Kybernetiki*, 10:119–139, 1963.

[13] P. Borm, H. Hamers, and R. Hendrickx. Operations research games : a survey. Discussion Paper 45, Tilburg University, Center for Economic Research, 2001.

[14] Yann Chevaleyre, Ulle Endriss, and Jérôme Lang. Expressive power of weighted propositional formulas for cardinal preference modelling. In P. Doherty, J. Mylopoulos, and C. Welty, editors, *Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR-2006)*, pages 145–152. AAAI Press, June 2006.

[15] Pardalos P.M. Migdalas A. Pitsoulis L. Chinchuluun, A. *Pareto Optimality, Game Theory and Equilibria*. Springer, 2008.

[16] A. Claus and D. Granot. Game theory application to cost allocation for a spanning tree. Working paper no. 402, Faculty of Commerce and Business Administration, University of British Columbia, June 1976.

[17] A. Claus and D. J. Kleitman. Cost allocation for a spanning tree. *Network*, 6:335–350, 1973.

[18] Vincent Conitzer and Tuomas Sandholm. Complexity of determining nonemptiness of the core. In *In Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI*, pages 613–618, 2002.

[19] Péter Csóka, P. Jean-Jacques Herings, and László Á. Kóczy. Balancedness conditions for exact games. Working paper series, Budapest Tech, Keleti Faculty of Economics, September 2007.

[20] I. Curiel. *Cooperative Game Theory and Applications*. Kluwer Academic, Boston, 1997.

[21] I. Curiel, J. Derks, and S. Tijs. On balanced games and games with committee control. *OR Spectrum*, 11(2):83–88, 1989.

[22] J. Deegan and Packel E.W. A new index of power for simple n-person games. *International Journal of Game Theory*, 7:113–123, 1978.

[23] X. Deng and Q. Fang. Algorithmic cooperative game theory. In *Pareto Optimality, Game Theory And Equilibria*, volume 17, pages 159–185. Springer, 2008.

[24] Xiaotie Deng and Christos H. Papadimitriou. On the complexity of cooperative solution concepts. *Math. Oper. Res.*, 19(2):257–266, 1994.

[25] Edsger Wybe Dijkstra. A note on two problems in connection with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[26] Theo Driessen. *Cooperative Games, Solutions and Applications*. Kluwer Academic Publishers, 1988.

[27] E. Elkind, L. A. Goldberg, P. W. Goldberg, and M. Wooldridge. Computational complexity of weighted threshold games. In *Twenty-Second Conference on Artificial Intelligence (AAAI-07)*, 2007.

[28] Edith Elkind, Leslie Ann Goldberg, Paul W. Goldberg, and Michael Wooldridge. A tractable and expressive class of marginal contribution nets and its applications. *Mathematical Logic Quarterly*, 2009. to appear.

[29] U. Faigle, S. Fekete, W. Hochstätter, and W. Kern. On the complexity of testing membership in the core of min-cost spanning tree games. *International Journal of Game Theory*, 26:361–366, 1997.

[30] Q. Fang, S. Zhu, M. Cai, and X. Deng. On computational complexity of membership test in flow games and linear production games. *International Journal of Game Theory*, 31:39–45, 2002.

[31] Qizhi Fang, Shanfeng Zhu, Mao-cheng Cai, and Xiaotie Deng. Membership for core of lp games and other games. In *COCOON '01: Proceedings of the 7th Annual International Conference on Computing and Combinatorics*, pages 247–256, London, UK, 2001. Springer-Verlag.

[32] L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.

[33] Vito Fragnelli, Ignacio Garcia-Jurado, and Luciano Mendez-Naya. On shortest path games. *Mathematical Methods of Operations Research*, 52:251–264, 2000.

[34] M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[35] D. Granot. A generalized linear production model: A unified model. *Mathematical Programming*, 34:212–222, 1986.

[36] D. Granot and G. Huberman. Minimum cost spanning tree games. *Mathematical Programming*, 21:1–18, 1981.

[37] Manfred Holler. Forming coalitions and measuring voting power. *Political Studies*, 30(2):262–271, 1982.

[38] Manfred J. Holler, Jose M. Alonso-Meijide, Balbina Casas-Mendez, and G. Fiestras-Janeiro. The deegan-packel index for simple games with a priori unions. University of Hamburg Working Paper No. 156.

[39] Manfred J. Holler, Jose M. Alonso-Meijide, Balbina Casas-Mendez, and G. Fiestras-Janeiro. Two variations of the public good index for games with a priori unions. Working paper 15. Public Choice Research Center University of Turku, Finland.

[40] Samuel Ieong and Yoav Shoham. Marginal contribution nets: a compact representation scheme for coalitional games. In *EC '05: Proceedings of the 6th ACM conference on Electronic commerce*, pages 193–202, New York, NY, USA, 2005. ACM.

[41] E. Kalai and E. Zemel. Totally balanced games and games of flow. *Math. Oper. Res.*, 7:476–478, 1982.

[42] Mamoru Kaneko and Myrna H. Wooders. Utility theories in cooperative games, 1996.

[43] Tomomi Matsui and Yasuko Matsui. A survey of algorithms for calculating power indices of weighted majority games. *J. Oper. Res. Soc. Japan*, 43:71–86, 2000.

[44] Yasuko Matsui and Tomomi Matsui. Np-completeness for calculating power indices of weighted majority games. *Theor. Comput. Sci.*, 263(1-2):306–310, 2001.

[45] Nimrod Megiddo. Computational complexity of the game theory approach to cost allocation for a tree. *Mathematics of Operation Research*, 3(3):189–196.

[46] Guillermo Owen. On the core of linear production games. *Mathematial Programming*, 9:358–370, 1975.

[47] Christos M. Papadimitriou. *Computational complexity*. Addison-Wesley, Reading, Massachusetts, 1994.

[48] Bezalel Peleg and Peter Sudhölter. *Introduction to the Theory of Cooperative Games (Theory and Decision Library C)*. Springer, 2nd ed. edition, October 2007.

[49] K. Prasad and J. S. Kelly. Np-completeness of some problems concerning voting games. *Int. J. Game Theory*, 19(1):1–9, 1990.

[50] R. C. Prim. Shortest connection networks and some generalizations. *Bell System Technical Journal*, 36:1389–1401, 1957.

[51] Lloyd S. Shapley and Pradeep Dubey. Totally balanced games arising from controlled programming problems. *Mathematical Programming*, 29(3):245–267, 1984.

[52] L.S. Shapley. On balanced sets and cores. *Naval Research Logistics Quarterly*, 14:453–460, 1967.

[53] L.S. Shapley and M. Shubik. A method for evaluating the distribution of power in a committee system. *American Political Science Review*, 48:787–792, 1954.

[54] L.S. Shapley and M. Shubik. On market games. *Journal of Economic Theory*, 1:9–25, 1969.

[55] Joel Uckelman, Yann Chevaleyre, Ulle Endriss, and Jérôme Lang. Representing utility functions via weighted goals. *Mathematical Logic Quarterly*, 2009.

[56] Leslie G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.

[57] John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior (Commemorative Edition) (Princeton Classic Editions)*. Princeton University Press, 1944.

[58] Mark Voorneveld and Sofia Grahn. Cost allocation in shortest path games. *Mathematical methods of operations research*, 56(2):323–340, 2002.