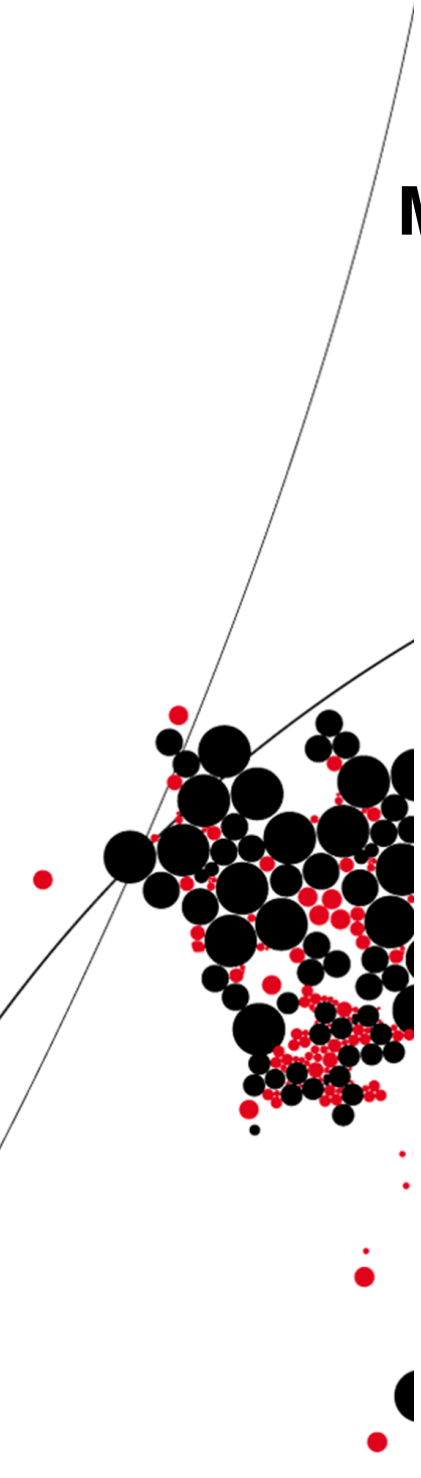
UNIVERSITY OF TWENTE.

**Faculty of Electrical Engineering,
Mathematics & Computer Science**

Almost Core Allocations on Minimum Cost Spanning Tree Games

Boyue Lin

M.Sc. Thesis
October 2022



Supervisors:
prof. dr. M. J. Uetz
dr. M. Walter

Graduation committee:
prof. dr. M. J. Uetz
dr. M. Walter
dr. J. B. Timmer

DMMP Group
Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Abstract

In cooperative game theory, a core allocation guarantees that no subset of players would prefer to deviate. However, the allocation in the core requires distributing the exact cost of the grand coalition to all individual players. This thesis studies the optimization problem to maximize the total amount that can be distributed over the individual players while maintaining the stability with respect to proper coalitions. The corresponding solution concept is referred to as “almost core”. It shows how much one could maximally tax the total cost of the grand coalition, without any proper coalition wanting to deviate. Some complexity theoretic results of almost core allocations are derived. We study almost core allocations to a class of games with non-empty core: the minimum cost spanning tree games. A tight 2-approximation algorithm to compute an almost core optimum is proposed. The algorithms to compute the almost core optimum on some constrained minimum cost spanning tree structures are also given. In addition, the almost core concept is studied for value games. It turns out that the same algorithmic results can be derived also for value games.

Preface

It's a fun and challenging year both in academics and in life. I started working on some directions I had rarely studied before, and I found it very interesting to learn discrete optimization. I am more aware of the fascination of algorithms. I learnt so much during the research with prof. Marc Uetz, dr. Matthias Walter and Rong Zou. I really enjoy it.

First of all, I would like to thank prof. dr. Marc Uetz, for welcoming me into the DMMP group. It is my first time studying abroad for so long, but you turned all my insecurities and worries into delight and contentment with your kindness, and your enthusiasm for academics encouraged me to dig deeper into discrete optimization. In addition, my special thanks go out to dr. Matthias Walter. You have a unique perspective on many problems and always inspire me in academics. Moreover, I am grateful to dr. Judith Timmer for being a member of my graduation committee. Thank you for taking the time and effort to read the thesis as well as to attend and assess the defence.

Furthermore, my gratitude goes out to my supervisor in China, prof. dr. Genjiu Xu. Thank you for the opportunity you gave me to be an exchange student in the Netherlands. You always support me and are not only my supervisor but also my mentor in life.

Also, I would like to say thank you to Rong Zou. Thank you for all the help when I just got Enschede. You are always there for me to give advice when I get confused. Last but not least, I want to thank all my friends of Xin Gen Ju Di. You are like family to me and made me realize I am not alone. I am so lucky to have you as friends because you are so awesome!

It has been a fulfilling year, and I am happy to say that being an exchange student at UT is one of my best memories. I hope I can continue my academics and dig deeper into optimization and algorithms in the future.

Contents

Chapter 1

Introduction

In this section, cost sharing games and some relatively basic concepts are introduced. Before bringing up the definition of almost core, we discuss relations among other various relaxations of the core. The motivation and the organization of the report are also mentioned. In this report, We mainly focused on cost sharing games in Chapter 2- Chapter 4, and the discussion of value games is given in Chapter 5.

1.1 Cost Sharing Games

Game theory studies mathematical models of strategic behaviours among a set of players. It has a wide range of applications in economics, political science, finance, artificial intelligence, psychology, etc. Modern game theory is based on the book of Von Neumann and Morgenstern [1]. In a game theory model, players can take actions, and the outcome depends on both their own actions and other players' action. the outcome of the action can be represented by a well-defined utility function. Normally, players are intelligent, which means they are capable enough to compute their best strategies. There are two types of games in modern game theory: cooperative games and noncooperative games. The main difference is in cooperative games, there are binding agreements among the players. Depending on whether the players get payoffs or are charged fees, we refer to cooperative games as value games or cost sharing games.

Cost sharing games are usually with transferable utility. In this case, the utility can be arbitrarily transferred among the members of a coalition. These kinds of games are called TU games (TU stands for transferable utility). A TU game with player set N consists of a characteristic cost function $c : 2^N \to \mathbb{R}_{\geq 0}$. The intended meaning is that $c(S), S \subseteq N$, denotes the cost for players to form the coalition S and accomplish the task together.

Usually in cost sharing games, for a coalition S, $c(S)$ is given in the following way:

Players in S form a coalition and try to get the lowest cost for S. Other players may form other coalitions against S. In this case, $c(S)$ is the best cost that S can obtain.

To allocate the total cost incurred by the grand coalition N, let x be an allocation vector over all individual players, and define $x^{\mathsf{AC}}(S) = \sum_{i \in S} x^{\mathsf{AC}}(i)$, $x^{\mathsf{AC}}(N) = \sum_{i \in N} x^{\mathsf{AC}}(i)$. A question is how to obtain a stable allocation, it means no coalition would break away and form new coalitions that all its members get a smaller cost share. Coalitional stability is an important indicator, which is $x(S) \leq c(S)$ for all $S \subsetneqq N$. If the allocation x to all players equals the total cost of the grand coalition N, that is $x(N) = c(N)$, the allocation is said to be budget balanced.

The *core* [2] is a solution concept that satisfies both coalitional stability and budget balance:

$$C_{\langle N,c \rangle} \coloneqq \{x \in \mathbb{R}^n : x(S) \leq c(S)\ \forall S \subsetneqq N,\ x(N) = c(N)\}\,. \tag{1.1}$$

Let x^{C} be a core allocation in $C_{\langle N,c \rangle}$. Every allocation in the core guarantees no coalition would have an incentive to deviate from the proposed cost shares x. To make the problem not trivial, we only consider games with more than one player in this thesis.

1.2 Relaxations of the Core

A core allocation requires distributing the total cost incurred by a set of players over individual players so that no subset of players would prefer to deviate. However, the core may be empty. In order to obtain a solution also for the unbalanced game, the core restrictions can be relaxed. Several relaxed concepts to the core have been proposed in the literature.

Shapley and Shubik [3] were the first to extend the concept of the core by introducing the *strong ε-core*:

$$C^{\varepsilon}_{\mathsf{s}}(N,c) \coloneqq \{x \in \mathbb{R}^n : x(S) \leq c(S) + \varepsilon\ \ \forall S \subsetneqq N,\ x(N) = c(N)\}\,.$$

The strong ε-core consists of all pre-cost-sharing vectors that give rise to $x - c$ not greater than ε for all proper coalitions $S \subsetneqq N$. If ε is 0, $C^0_{\mathsf{s}}(N,c)$ is equal to the core. If $\varepsilon_1 \leq \varepsilon_2$, $C^{\varepsilon_1}_{\mathsf{s}}(N,c) \subseteqq C^{\varepsilon_2}_{\mathsf{s}}(N,c)$. If ε is large enough, the strong ε-core is non-empty. We denote the smallest $\varepsilon \geq 0$ for which this set is non-empty by $\varepsilon^{\star}_{\mathsf{s}}$. The corresponding set $C^{\varepsilon^{\star}_{\mathsf{s}}}_{\mathsf{s}}(N,c)$ is called the *least core* [4].

Shapyley and Shubik [3] also introduced the *weak ε-core* as

$$C^{\varepsilon}_{\mathsf{w}}(N,c) \coloneqq \{x \in \mathbb{R}^n : x(S) \leq c(S) + \varepsilon \cdot |S|\ \forall S \subsetneqq N,\ x(N) = c(N)\}\,.$$

Instead of taxing a fixed rate on the coalitions as for strong ε-core, weak ε-core means a proper coalition is taxed with rate ε proportionally to its size. Similar to the

strong ε-core, if ε is large enough, the weak ε-core is non-empty. We denote the smallest $\varepsilon \geq 0$ for which this set is non-empty by $\varepsilon^\star_{\mathsf{w}}$. Note that by definition, for any $\varepsilon \geq 0$, $C^\varepsilon_{\mathsf{s}}(N,c) \subseteq C^\varepsilon_{\mathsf{w}}(N,c)$, and hence $\varepsilon^\star_{\mathsf{w}} \leq \varepsilon^\star_{\mathsf{s}}$.

Rather than using an additive relaxation of the constraints, Faigle and Kern [5] defined the *multiplicative ε-core* as

$$C^\varepsilon_{\mathsf{m}}(N,c) \coloneqq \{x \in \mathbb{R}^n : x(S) \leq (1+\varepsilon)\cdot c(S)\ \forall S \subsetneqq N,\ x(N) = c(N)\}\,.$$

In multiplicative ε-core, the tax imposed on a proper coalition is proportional to its cost $c(S)$. When the multiplicative ε-core is non-empty, we denote the smallest $\varepsilon \geq 0$ for which this set is non-empty by $\varepsilon^\star_{\mathsf{m}}$.

A different viewpoint is called *approximate core* or *γ-core* [6] for some $\gamma \in [0,1]$, it is defined as

$$C^\gamma_{\mathsf{a}}(N,c) \coloneqq \{x \in \mathbb{R}^n : x(S) \leq c(S)\ \forall S \subseteq N,\ \gamma\cdot c(N) \leq x(N)\}\,. \tag{1.2}$$

For games where there exists the non-empty approximation core for some $\gamma \in [0,1]$. Denote the largest $\gamma \leq 1$ for which this set is non-empty by $\gamma^\star_{\mathsf{a}}$.

The *cost of stability* for an unbalanced cooperative value game was introduced by Bachrach et al. [7]. They defined it as the minimal monetary infusion required to stabilize a game with nonnegative allocation. For (unbalanced) cost sharing games it is defined by Meir et al. [8] as

$$\delta^\star_{\mathsf{CoS}} \coloneqq c(N) - \max\{x(N) : x(S) \leq c(S)\ \forall S \subseteq N\}\,.$$

An alternative viewpoint was independently introduced in a paper by Bejan and Gómez [9] who considered, for profit sharing games, the so-called extended core. In order to define it for cost sharing games, let

$$\delta^\star_{\mathsf{ec}} \coloneqq \min\{t(N) : \exists (x,t) \in \mathbb{R}^n \times \mathbb{R}^n_{\geq 0},\ x(N) = c(N),\ (x-t)(S) \leq c(S)\ \forall S \subsetneqq N\}\,. \tag{1.3}$$

The *extended core* is the set of all budget balanced $x \in \mathbb{R}^n$, so all x with $x(N) = c(N)$ for which the minimum above is attained (for suitable $t \in \mathbb{R}^n_{\geq 0}$).

Yet another concept to stabilize an unbalanced game was considered by Zick, Polukarov, and Jennings [10]. Interpreting t_i in the definition of the extended core of Bejan and Gómez [9] as a discount offered to player i, in [10] a coalitional discount t_S is offered to each player set S. This is an exponential blowup of the solution space, which however gives more flexibility.

In this thesis, we introduce a model where the equation constraint in the core is relaxed. We study the optimization problem to maximize the total amount that can be distributed over the individual players while maintaining the stability of proper coalitions $S \subsetneqq N$.

1.3 Minimum Cost Spanning Tree Games

A classic type of cost sharing game is minimum cost spanning tree games (MST games). MST games were first introduced by Bird [11]. After that, Granot and Huberman [12, 13, 14] studied the core and the nucleolus of MST games and proposed some algorithms to compute a core allocation. Faigle and Kern [5, 15, 16] focused on the complexity of the core and the least core of minimum cost spanning tree games.

To motivate MST games, consider the situation in which some houses need electricity from a unique power station. Electricity cables can be installed among the houses and the power station, and can only be laid between two buildings. Different cables between different buildings have different installation costs. We try to minimize the total installation cost such that every building can use electricity. After determining the installation plan, the power station needs an allocation to charge every house for the total installation cost. By modelling the problem as a graph, the vertices in the graph represent the buildings, and there is a "source" node as the power station. The weighted edge between any two vertices is a possible installation cable with a cost, we can then get an MST game.

More specifically, given an edge-weighted, undirected graph $G = (N \cup \{0\}, E)$ with non-negative edge weights $w : E \to \mathbb{R}_{\geq 0}$, where node 0 is a special node referred to as source node. Without loss of generality, we may assume that the graph is complete by adding dummy edges with a large enough cost if necessary. To prevent notation conflicts, we use (u, v) as undirected graph in this thesis, that is $(u, v) = (v, u)$.

The players of the game are the vertices N of the graph, and the characteristic cost function of the game is given by the costs of minimum spanning trees. That is, the cost of any subset of vertices $S \subseteq N$ is defined as the cost of a minimum cost spanning tree on the subgraph induced by vertex set $S \cup \{0\}$. The characteristic cost function is defined as:

$$c(S) \coloneqq \min_{T \in \mathcal{T}(S)} \left\{ \sum_{e \in T} w(e) \right\}.$$

Here, $\mathcal{T}(S)$ is the set of spanning trees for the subgraph induced by vertex set $S \cup \{0\}$.

1.4 Outline

The thesis comprises 6 chapters. In Chapter 1, some relative basic concepts of this thesis are introduced. The motivation and the organization of the report are also mentioned. In Chapter 2, this report develops the "almost core" concept to maximize

the total amount that can be distributed over the players N while maintaining stability with respect to proper coalitions. We also investigate the computational complexity of optimization problems related to the core and the almost core. In Chapter 3, the almost core allocation is applied to minimum cost spanning tree games. The computational complexity of optimizing over the almost core is discussed. A 2-approximation algorithm is proposed. In Chapter 4, we analyze the gap between the almost core optimum and the total cost of the grand coalition under the assumption of different graph structures for minimum cost spanning tree games. In addition, some observations are made about their properties. In Chapter 5, this report analyzes the almost core idea for value games, in particular, the minimum cost spanning tree value games, for which we also propose a 2-approximation algorithm.

Chapter 2

Cost Sharing Beyond the Core

In this chapter, we develop the "almost core" concept to maximize the total amount that can be distributed over the players N while maintaining stability with respect to proper coalitions. We also investigate the computational complexity of optimization problems related to the core and the almost core.

2.1 Almost Core Allocations

A core allocation in (1.1) requires distributing the total cost incurred by a set of players over individual players so that no subset of players would prefer to deviate. However, allocation vectors obtained from the core constraints may not be acceptable from a modelling point of view. In some cases, one may have to, or want to, distribute either less or more than $c(N)$.

While maintaining the stability of proper coalitions in (1.1), we drop the equality constraint that a core allocation is budget balanced, so do not require that $x(N) = c(N)$. This allows us to vary the total cost that is distributed over the set of players, resulting in a problem that always has a feasible solution.

For convenience, we refer to the set of all such allocations as the *almost core*, AC. Formally, given a TU game $\langle N, c\rangle \in \mathcal{G}^N$, define the almost core for $\langle N, c\rangle$ by

$$AC_{\langle N,c\rangle} \coloneqq \{x \in \mathbb{R}^n : x(S) \leq c(S)\ \forall S \subsetneqq N\}\,.$$

Note that $C_{\langle N,c\rangle} \subseteq AC_{\langle N,c\rangle}$ by definition.

If the total cost $c(N)$ of the grand coalition *cannot* be distributed over the set of players while maintaining coalitional stability, i.e., the game is unbalanced, it is a natural question to ask what fraction of the total cost $c(N)$ can be maximally distributed while maintaining coalitional stability. For this case maximizing $x(N)$ over the almost core is in fact equivalent to some of the earlier proposed core relaxations; see Section 2.2.

For games with non-empty core, one may be interested in maximizing the total cost that can be distributed over the set of players. The question is then how much one could maximally tax the total cost $c(N)$ of the grand coalition, without any proper coalition wanting to deviate.

To deal with these two questions, consider the following linear program.

$$\begin{aligned} \max \quad & x(N) \\ s.t. \quad & x(S) \leq c(S), \quad \forall S \subsetneqq N \\ & x \in \mathbb{R}^n \end{aligned} \tag{2.1}$$

Let x^{AC} be some optimal solution to this almost core optimization problem (2.1). We call $x^{\mathsf{AC}}(N)$ the *almost core optimum*, and any maximizer x^{AC} is called an *optimal almost core allocation*. The objective value of this linear program indicates the largest total cost that can be shared amongst the players while retaining stability in the sense that no subset of players $S \subsetneqq N$ would prefer to deviate, as no subset can realize smaller costs on its own.

2.2 Equivalence between the Almost Core and Other Relaxations of the Core

Clearly, the core of a game is non-empty if and only if the almost core optimum is larger than or equal to $c(N)$. For unbalanced games, the total costs $c(N)$ of the grand coalition cannot be distributed over the set of players while maintaining coalitional stability, computing the almost core optimum is equivalent to computing some of the earlier proposed core relaxations. The following theorem summarizes how the different core relaxations are related; the last equality and the inequality are from [17, Section 4].

We include proof of all equalities for convenience.

Theorem 2.2.1 (see also [17])**.** *For any TU game $\langle N, c\rangle$ with empty core, the optimization problems for the weak ε-core, the multiplicative ε-core, the cost of stability and the extended core are equivalent. In particular, the values satisfy*

$$\delta^\star_{ec} = (1-\gamma^\star_a)\cdot c(N) = \frac{\varepsilon^\star_m}{1+\varepsilon^\star_m}\cdot c(N) = \delta^\star_{CoS} = \varepsilon^\star_w \cdot n \geq \frac{n}{n-1}\varepsilon^\star_s .$$

Proof. First, we establish $\delta^\star_{\mathsf{CoS}} = \delta^\star_{\mathsf{ec}}$. We substitute $x - t$ by x' in (1.3) and obtain

$$\delta^\star_{\mathsf{ec}} = \min\{t(N) : \exists (x', t) \in \mathbb{R}^n \times \mathbb{R}^n_{\geq 0},\ x'(N) + t(N) = c(N), \\ x'(S) \leq c(S)\ \forall S \subsetneqq N\}\,. \tag{2.2}$$

Since t is only relevant with the equation constraint in (2.2), the actual entries of t do not matter (except for nonnegativity) , but only the value $t(N)$ is important. This yields $\delta^\star_{\mathsf{CoS}} = \delta^\star_{\mathsf{ec}}$.

Second, we show $\delta^\star_{\mathsf{CoS}} = (1-\gamma^\star_{\mathsf{a}}) \cdot c(N)$. To this end, observe

$$\gamma^\star_{\mathsf{a}} = \max\{\gamma \in \mathbb{R} : \exists x \in \mathbb{R}^n,\ x(S) \leq c(S)\ \forall S \subseteq N,\ x(N) = \gamma c(N)\}\,.$$

In this equation, the maximum is attained by $x^\star \in \mathbb{R}^n$ with $x^\star(N)$ maximum when saitisfying $x(S) \leq c(S)\ \forall S \subseteq N$. Moreover, the value of $\gamma^\star_{\mathsf{a}}$ is then equal to $x^\star(N)/c(N)$. This shows $1-\gamma^\star_{\mathsf{a}} = (c(N)-x^\star(N))/c(N) = \delta^\star_{\mathsf{CoS}}/c(N)$.

Third, we show $1-\gamma^\star_{\mathsf{a}} = \varepsilon^\star_{\mathsf{m}}/(1+\varepsilon^\star_{\mathsf{m}})$. Observe that the map $\pi : \mathbb{R}^n \to \mathbb{R}^n$ defined by $\pi(x) = (1+\varepsilon)x$ induces a bijection between allocations $x \in \mathbb{R}^n$ with $x(S) \leq c(S)$ for all $S \subseteq N$ and allocations $\pi(x)$ with $x(S) \leq (1+\varepsilon)c(S)$ for all $S \subseteq N$. Moreover, $x(\pi(N)) = (1+\varepsilon)x(N)$. Hence we can substitute the x in (1.2) by $\pi(x)$ and get:

$$C^\gamma_{\mathsf{a}}(N,c) \coloneqq \{x \in \mathbb{R}^n : x(S) \leq (1+\varepsilon)c(S)\ \forall S \subseteq N,\ (1+\varepsilon)\gamma \cdot c(N) \leq x(N)\}\,.$$

Hence, $C^\varepsilon_{\mathsf{m}}\langle N,c\rangle$ is (non-)empty if and only if $C^\gamma_{\mathsf{a}}\langle N,c\rangle$ is (non-)empty, where $\gamma = 1/(1+\varepsilon)$ holds. This implies $\gamma^\star_{\mathsf{a}} = 1/(1+\varepsilon^\star_{\mathsf{m}})$.

We finally show $\delta^\star_{\mathsf{CoS}} = \varepsilon^\star_{\mathsf{w}} \cdot n$. To this end, in

$$\varepsilon^\star_{\mathsf{w}} = \min\{\varepsilon \geq 0 : \exists x \in \mathbb{R}^n,\ x(S) \leq c(S) + \varepsilon \cdot |S|\ \forall S \subsetneq N,\ x(N) = c(N)\}$$

we substitute x by $x' + (\varepsilon, \varepsilon, \ldots, \varepsilon)$ which yields

$$\varepsilon^\star_{\mathsf{w}} = \min\{\varepsilon \geq 0 : \exists x' \in \mathbb{R}^n,\ x'(S) \leq c(S)\ \forall S \subsetneq N,\ x'(N) + \varepsilon \cdot n = c(N)\}\,.$$

Since $\delta^\star_{\mathsf{CoS}}$ is equal to the minimum $c(N) - x(N)$ when $x(S) \leq c(S)\ \forall S \subseteq N$ is satisfied, the minimum $\varepsilon^\star_{\mathsf{w}}$ is attained if and only if $\varepsilon \cdot n = \delta^\star_{\mathsf{CoS}}$ holds. For the final inequality relating the least core with the weak ε-core, we refer to the corresponding proof in [17]. □

Also notice that by definition of the cost of stability, it is equal to the gap between the almost core optimum and the total cost of the grand coalition $c(N)$. Further relations between the cost of stability $\delta^\star_{\mathsf{CoS}}$ and other core relaxations for specific classes of games appear in [17, 18], e.g., it is true that for superadditive (profit sharing) games, $\delta^\star_{\mathsf{CoS}} \leq \sqrt{n}\varepsilon^\star_{\mathsf{s}}$ and $\sqrt{n}\varepsilon^\star_{\mathsf{w}} \leq \varepsilon^\star_{\mathsf{s}}$.

Most of the previous work in this direction was about determining bounds on the cost of stability (see [7, 8, 17, 19, 20]) or other structural insights [9]. Algorithmic considerations were made for specific (unbalanced) games such as weighted voting games [7] and threshold network flow games [21]. Aziz, Brandt and Harrenstein [22] settle the computational complexity of computing the cost of stability (and other measures) for many other specific games, and Chalkiadakis, Greco and Markakis extend

this work under additional assumptions on the so-called interaction graph [23]. Approximations of $\varepsilon^\star_{\mathsf{m}}$ for the multiplicative $(1+\varepsilon)$-core and corresponding allocations have also been obtained for the symmetric traveling salesman game by Faigle et al. [24], and for the asymmetric case also by Bläser et al. [25].

There are also papers that attack the problem from a computational point of view. Under the name "optimal cost share problem", Caprara and Letchford [26] computationally obtain γ-core solutions for a generalization of certain combinatorial games, named integer minimization games.

Under the name "optimal cost allocation problem", also Liu, Qi and Xu [27] give computational results using Lagrangian relaxation, which also works for nonlinear objective functions.

Also, the problem to compute allocations in the least core has been considered in the literature. For cooperative games with submodular cost functions, it can be computed in polynomial time [28], while for supermodular cost cooperative games it is NP-hard to compute, and even hard to approximate [29]. Finally, Faigle et al. [16] show NP-hardness to compute a cost allocation in the so-called f-least core for minimum cost spanning tree games, which is a tightening of the core constraints to $x(S) \leq c(S) - \varepsilon f(S)$ for certain non-negative set functions f. As we will argue later, their result also implies the hardness of computing optimal almost core allocations for minimum spanning trees.

2.3 On the Complexity of Almost Core Allocations

In this section, we investigate the computational complexity of optimization problems related to the core and the almost core. To capture results for the general and the nonnegative case, we consider linear optimization over the polyhedra

$$AC_{\langle N,c\rangle} \quad \text{and} \quad P_{\langle N,c\rangle} \coloneqq \{x \in \mathbb{R}^n : x(S) \leq c(S)\ \forall S \subseteq N\}.$$

as well as optimization over $P_{\langle N,c\rangle} \cap \mathbb{R}^n_{\geq 0}$ and $AC_{\langle N,c\rangle} \cap \mathbb{R}^n_{\geq 0}$ for families of games $\langle N, c\rangle$. Note that the core of the cost sharing games is the set of optimal solutions when maximizing $\mathbb{1}$ over $P_{\langle N,c\rangle}$. Also note that whenever the core of a game $\langle N, v\rangle$ is empty, so no x exists with $x(N) \geq c(N)$ and $x(S) \leq c(S)\ \forall S \subseteq N$, we have that $x(N) < c(N)$ is implied by the set of constraints $x(S) \leq c(S)$, $S \subsetneq N$, which in turn implies $P_{\langle N,c\rangle} = AC_{\langle N,c\rangle}$. For games with non-empty cores, we get the following correspondence between the optimization problems for the two polyhedra.

Theorem 2.3.1. *For a family of games $\langle N, c\rangle$, linear optimization problems over $AC_{\langle N,c\rangle}$ can be solved in polynomial time if and only if linear optimization problems over $P_{\langle N,c\rangle}$ can be solved in polynomial time.*

Proof. To prove the result we make use of the equivalence of optimization and separation [30, 31, 32], that is, linear optimization problems over a polyhedron can be solved in polynomial time if and only if the separation problem for the polyhedron can be solved in polynomial time. Hence we only need to show that we can solve the separation problem for $P_{\langle N,c\rangle}$ if and only if we can solve the separation problem for $AC_{\langle N,c\rangle}$. Since $P_{\langle N,c\rangle} = \{x \in AC_{\langle N,c\rangle} : x(N) \leq c(N)\}$ holds, separation over $P_{\langle N,c\rangle}$ reduces to separation over $AC_{\langle N,c\rangle}$ plus an explicit check of the inequality $x(N) \leq c(N)$.

It remains to show how to solve the separation problem for $AC_{\langle N,c\rangle}$ when we can solve that for $P_{\langle N,c\rangle}$. For given $\hat{x} \in \mathbb{R}^n$, we construct n points $\hat{x}^k \in \mathbb{R}^n$ ($k = 1, 2, \ldots, n$) which are copies of $\hat{x}$ except for $\hat{x}^k_k \coloneqq \min(\hat{x}_k, c(N) - \sum_{i\in N\setminus\{k\}} \hat{x}_i)$. Note that by construction $\hat{x}^k \leq \hat{x}$ and $\hat{x}^k(N) \leq c(N)$ hold. We then query a separation algorithm of $P_{\langle N,c\rangle}$ for each $\hat{x}^k$.

Suppose such a query yields $\hat{x}^k(S) > c(S)$ for some $S \subseteq N$. Due to $\hat{x}^k(N) \leq c(N)$ we have $S \neq N$. Moreover, $\hat{x} \geq \hat{x}^k$ implies $\hat{x}(S) > c(S)$, and we can return the same violated inequality to testify that $\hat{x} \notin AC_{\langle N,c\rangle}$.

Otherwise, we have $\hat{x}^k \in P_{\langle N,c\rangle}$ for all $k \in N$ and claim $\hat{x} \in AC_{\langle N,c\rangle}$. To prove this claim we assume that, for the sake of contradiction, $\hat{x}(S) > c(S)$ holds for some $S \subsetneq N$. Let $k \in N \setminus S$. Since for all $i \in S$, $i \neq k$ and $\hat{x}^k_i = \hat{x}_i$ holds, we have $\hat{x}^k(S) = \hat{x}(S) > c(S)$. This contradicts the fact that $\hat{x}^k \in P_{\langle N,c\rangle}$ holds. □

It turns out that almost the same result is true when we also require that there are no subsidies, that is $x \geq 0$. For linking the non-negative core to the non-negative almost core, it requires an assumption on the characteristic function.

$$c(N \setminus \{k\}) \leq c(N) \qquad \forall k \in N. \tag{2.3}$$

This condition holds, for instance, for monotone functions c, and implies that the core is contained in $\mathbb{R}^n_{\geq 0}$ (see Lemma 2 and Theorem 1 in [33]).

Theorem 2.3.2. *For a family of games $\langle N, c\rangle$ satisfying* (2.3)*, linear optimization problems over $AC_{\langle N,c\rangle} \cap \mathbb{R}^n_{\geq 0}$ can be solved in polynomial time if and only if linear optimization problems over $P_{\langle N,c\rangle} \cap \mathbb{R}^n_{\geq 0}$ can be solved in polynomial time.*

Proof. The proof for Theorem 2.3.2 is the same as that of Theorem 2.3.1 with only a few exceptions. All vectors must be restricted to $\mathbb{R}^n_{\geq 0}$ to solve the separation problem for $AC_{\langle N,c\rangle} \cap \mathbb{R}^n_{\geq 0}$ when we can solve that for $P_{\langle N,c\rangle} \cap \mathbb{R}^n_{\geq 0}$. For given $\hat{x} \in \mathbb{R}^n_{\geq 0}$, we construct n points $\hat{x}^k \in \mathbb{R}^n_{\geq 0}$ ($k = 1, 2, \ldots, n$) which are copies of $\hat{x}$ except for $\hat{x}^k_k \coloneqq \min(\hat{x}_k, c(N) - \sum_{i\in N\setminus\{k\}} \hat{x}_i)$.

By (2.3), $c(N) - c(N \setminus \{k\}) \geq 0$ and we know $c(N) - \sum_{i\in N\setminus\{k\}} \hat{x}^k_i \geq c(N) - c(N \setminus \{k\}) \geq 0$. Also with $\hat{x}_k \geq 0$, we know $\hat{x}^k_k \geq 0$, and we can conclude $\hat{x}^k \geq 0$. □

We obtain some immediate consequences.

Corollary 2.3.3. *For a family of games $\langle N, c\rangle$ for which $c(\,\cdot\,)$ is submodular (and* (2.3) *holds) one can find a (non-negative) optimal almost core allocation in polynomial time.*

Proof. For submodular $c(\,\cdot\,)$ one can optimize any linear objective function over $P_{\langle N,c\rangle}$ using the Greedy algorithm [34]. The result follows from Theorems 2.3.1 and 2.3.2. □

These results only make statements about optimizing arbitrary objective vectors over these polyhedra. In particular, we cannot draw conclusions about the hardness of the computation of an almost core allocation refer to (2.1). However, it is easy to see that this problem cannot be easier than deciding non-emptiness of the core.

Proposition 2.3.4. *Consider a family of games $\langle N, c\rangle$ for which deciding non-emptiness of the core is (co)NP-hard. Then finding an optimal almost core allocation is also (co)NP-hard.*

Proof. By the premise of the theorem there exists a Karp reduction from some NP-hard problem $\mathcal{P}$ to the non-emptiness decision problem for our family of games. The reduction turns (in polynomial time) an instance $\mathcal{I}$ of $\mathcal{P}$ into a game $\langle N, c\rangle$ such that $\mathcal{I}$ is a YES-instance (resp. NO-instance) if and only if $\langle N, c\rangle$ has a non-empty core.

The almost core optimum is at least $c(N)$ if $\langle N, c\rangle$ has a non-empty core. Also, if the almost core optimum is at least $c(N)$, it means there exists an allocation x so that $x(S) \leq c(S)\ \forall S \subsetneqq N$ and $x(N) \geq c(N)$, and then we can get a core allocation by scaling down $x(N)$. As a result, the almost core optimum is at least $c(N)$ if and only if $\langle N, c\rangle$ has a non-empty core. The same reduction works for the almost core and $\mathcal{I}$ is a YES-instance (resp. NO-instance) if and only if the almost core optimum is at least $c(N)$. □

It is well known that there exist games for which it is NP-hard to decide the non-emptiness of the core, e.g., the weighted graph game [35]. Hence, we cannot hope for a polynomial-time algorithm that computes an optimal almost core allocation for arbitrary games.

In contrast, the maximization of $x(N)$ becomes trivial for games $\langle N, c\rangle$ with superadditive characteristic function $c(\,\cdot\,)$, as the set of constraints $x(\{i\}) \leq c(\{i\})$, $i = 1, \ldots, n$, already imply all other constraints $x(S) \leq c(S)$, $S \subsetneqq N$, hence one can set up a compact linear program.

Proposition 2.3.5. *For games $\langle N, c\rangle$ with superadditive $c(\,\cdot\,)$, finding an optimal almost core allocation can be done in polynomial time.*

Proof. Superadditive cost satisfies: $c(S \cup T) \geq c(S) + c(T)$. Hence with superadditivity, the allocation $x_i = c(\{i\})$ is in the almost core $AC_{\langle N,c\rangle}$ and $x(N)$ can not be improved. □

The same is true for all classes of games where a polynomial number of constraints can be shown to be sufficient to define the complete core. *Matching games* in undirected graphs [36] is one example of such a game, where it is sufficient to consider the polynomially many core constraints induced by all edges of the graph, as these imply all other core constraints.

Note that Proposition 2.3.5 also includes supermodular cost functions. It is therefore interesting to note that for supermodular cost games, it is NP-hard to approximate the least core value $\varepsilon^\star_{\mathsf{s}}$ better than a factor $17/16$ [29].

It also turns out that condition (2.3) implies that the value of an almost core allocation cannot exceed that of a core allocation by much.

Proposition 2.3.6. *Let $\langle N, c\rangle$ be a game that satisfies* (2.3)*. Then every $x \in AC_{\langle N,c\rangle}$ satisfies $x(N) \leq \big(1 + \frac{1}{n-1}\big)c(N)$.*

Proof. Let $x \in AC_{\langle N,c\rangle}$. We obtain

$$(n-1) \cdot x(N) = \sum_{k\in N} x(N \setminus \{k\}) \leq \sum_{k\in N} c(N \setminus \{k\}) \leq \sum_{k\in N} c(N) = n \cdot c(N),$$

where the first inequality follows from the feasibility of x and the second follows from (2.3). □

Condition (2.3) implies non-negativity for all core allocations and all optimal almost core allocations. However, this does not mean that a non-negativity requirement implies that the almost core optimum is close to $c(N)$. In the next section, we will show that this gap can be arbitrarily large, even for MST games (see Proposition 3.2.1).

Chapter 3

Almost Core Allocations on MST games

This chapter applies the almost core concept to minimum cost spanning tree games. It is a class of games for which the core is non-empty. The computational complexity of almost core for MST games is also discussed. For non-negative allocations, a 2-approximation algorithm is proposed.

3.1 The Core of MST games

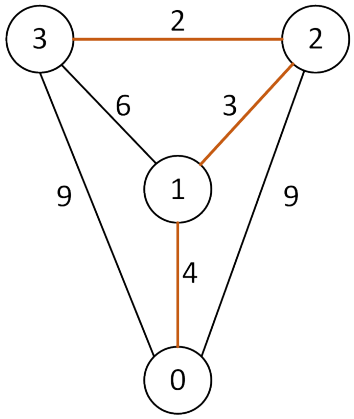
To describe a minimum cost spanning tree game, given an edge-weighted, undirected graph $G = (N \cup \{0\}, E)$ with non-negative edge weights $w : E \to \mathbb{R}_{\geq 0}$, where node 0 is a special node referred to as "source" node. Without loss of generality, we may assume that the graph is complete by adding dummy edges with a large enough cost if necessary.

The players of the game are the vertices N of the graph, and the characteristic function of the game is given by the costs of minimum spanning trees. That is, the cost of any subset of vertices $S \subseteq N$ is defined as the cost of a minimum cost spanning tree on the subgraph induced by vertex set $S \cup \{0\}$. The characteristic function $c(S)$ is defined as:

$$c(S) \coloneqq \min_{T \in \mathcal{T}(S)} \left\{ \sum_{e \in T} w(e) \right\}.$$

Here, $\mathcal{T}(S)$ is the set of spanning trees for the subgraph induced by vertex set $S \cup \{0\}$. $c(S)$ is the cost of a minimum spanning tree on the subgraph induced by vertex set $S \cup \{0\}$.

Definition 3.1.1. *The **monotonic cover** of a cooperative cost game $\langle N, c \rangle$ is the*



S	$c(S)$	$\bar{c}(S)$
{1}	4	4
{2}	9	7
{3}	9	9
{1,2}	7	7
{1,3}	10	9
{2,3}	11	9
{1,2,3}	9	9

Figure 3.1: Coalitional costs in MST game in Example 3.1.2 and its monotonic cover

cost game $\langle N, \bar{c}\rangle$ *given by*

$$\bar{c}(S) := \min\{c(T) \mid S \subseteq T\} \quad \forall S \in 2^N \setminus \{\emptyset\}$$

By definition, the monotonic cover possesses the following properties.

- $\bar{c}(S) \leq c(S) \quad \forall S \in 2^N$
- $\langle N, \bar{c}\rangle$ is monotonic: $\bar{c}(S) \leq \bar{c}(R) \quad \forall\, S \subseteq R$,

Note that for the associated core of a cooperative cost game and its monotonic cover, we always have that $C_{\langle N,\bar{c}\rangle} \subseteq C_{\langle N,c\rangle}$.

Example 3.1.2. *Let* $N = \{1, 2, 3\}$*. Consider the graph in Figure 3.1. The weights of the edges are as in the figure, and the MST is marked in orange. The table on the right-hand side shows the cost of various coalition in MST game* $\langle N, c\rangle$ *and its monotonic cover* $\langle N, \bar{c}\rangle$*.*

Proposition 3.1.3. *The core of a minimum cost spanning tree game and the core of its monotonic cover are not empty.*

Proof. The core is non-empty if we can get an allocation in the core. Actually, we can get one core element by "reading" directly from a minimum cost spanning tree graph [12], and the algorithm proposed later will also prove this result. □

3.2 Gap Between the core and the Almost Core on Minimum Cost Spanning Tree Games

The linear program (2.1) may include negative solutions, that is, a negative cost share may be considered, which is actually a payoff. One may wish to avoid this situation by adding the constraints $x_i \geq 0,\ i = 1, \cdots, n$.

In this case, the almost core allocation problem will be:

$$\begin{aligned} \max \quad & x(N) \\ s.t. \quad & x(S) \leq c(S), \quad \forall S \subsetneqq N \\ & x \in \mathbb{R}^n_{\geq 0} \end{aligned} \tag{3.1}$$

Let x^{OPT} be some optimal solution to this optimization problem.

Proposition 3.2.1. *The almost core optimum can be arbitrarily larger than $c(N)$, even for minimum cost spanning tree games and when we require that $x \geq 0$ for Problem (3.1).*

Proof. Consider the instance depicted in Figure 3.2, for some value $k > 0$. Then $c(N) = 0$, while $x = (0, 0, 2k)$ is an optimal non-negative almost core allocation with value $2k$. □

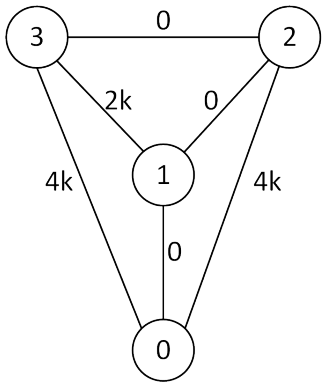


Figure 3.2: MST game with large relative gap between almost core optimum and $c(N)$.

3.3 Computational Complexity

Proposition 3.3.1. *Computing an optimum almost core allocation in* (2.1) *for minimum cost spanning tree games is NP-hard.*

Proof. Let $\varepsilon^\star$ be the largest ε for which the linear inequality system

$$x(S) \leq c(S) - \varepsilon c(S) \; \forall S \subsetneqq N, \quad x(N) = c(N) \tag{3.2}$$

has a solution. In [16] it is shown that finding a feasible solution x for (3.2) with respect to $\varepsilon^\star$ is NP-hard. Note that in the reduction leading to this hardness result, $c(N) > 0$. Then, given an optimum almost core allocation x^{AC}, $x^{\mathsf{AC}}(N) \geq c(N) > 0$,

and we can obtain $\varepsilon^\star \coloneqq 1-c(N)/x^{\mathsf{AC}}(N)$. We can see that the vector $x' \coloneqq (1-\varepsilon^\star)x^{\mathsf{AC}}$ is a feasible solution for (3.2). To see that the so-defined $\varepsilon^\star$ is indeed maximal, observe that scaling any feasible vector in (3.2) by $1/(1-\varepsilon^\star)$ yields an almost core allocation. Hence, computation of an almost core optimum for MST games yields a solution for an NP-hard problem. □

Proposition 3.3.2. *Testing membership in the almost core $AC_{\langle N,c\rangle}$ for minimum cost spanning tree games is coNP-hard.*

Proof. It has been proved that the core membership testing is coNP-hard in [15]. We follow the same reduction from the exact cover by 3-sets problem (X3C) to show that the almost core membership testing is also coNP-hard. Given an instance of X3C: A finite set $A = \{a_1, \cdots, a_{3q}\}$ and a collection $F = \{f_1, \cdots, f_{|F|}\}$ of 3-element subsets of A. Construct the following MST game corresponding to graph (N, E) in Figure 3.3. So the player set in this game is $\{1, \cdots, 3q\} \cup \{3q+1, \cdots, 3q+k\} \cup \{g, St\}$ consist of element-players, set-players, Steiner-point-player and guardian. For each set $f_i = \{j, e, l\}(i = 1, \cdots, |F|)$, the weights of edges are given by:

$$w(u,v) = \begin{cases} q+1 & \forall u = 3q+i, v \in \{j,e,l\} \\ q & \forall u = 3q+i, v = St \\ q+1 & \forall u = 3q+i, v = g \\ 2q-1 & \forall u = g, v = 0 \\ q+1 & \forall u = St, v = g \end{cases} \tag{3.3}$$

The weights of other edges are induced from triangle inequality of these edges.

An allocation of this MST game is on player set $\{1, \cdots, 3q, 3q+1, \cdots, 3q+k, st, g\}$. Let x be an allocation given by:

$$x_i = \begin{cases} q+2 & \forall i = 1, \cdots, 3q \\ q & \forall i = 3q+1, \cdots, 3q+k \\ 0 & \forall i \in \{St, g\} \end{cases} \tag{3.4}$$

By using the greedy algorithm in [12], we can get a core allocation y as:

$$y_i = \begin{cases} q+1 & \forall i = 1, \cdots, 3q \\ q & \forall i = 3q+1, \cdots, 3q+k \\ q+1 & i = St \\ 2q-1 & i = g \end{cases} \tag{3.5}$$

Recall that $3q$ is the number of the elements in finite set A hence $q \geq 1$. Consider $d \coloneqq x - y$, we get:

$$d_i = \begin{cases} 1 & \forall i = 1, \cdots, 3q \\ 0 & \forall i = 3q+1, \cdots, 3q+k \\ -q-1 & i = St \\ -2q+1 & i = g \end{cases} \tag{3.6}$$

Assume x is not in $AC_{\langle N,c\rangle}$ of the MST game, so there exists $S \subsetneqq N$ that $x(S) > c(S)$. Since y is in the core, we know $y(S) \leq c(S)$ and $d(S) > 0$. We can have two conclusions from it, first, there is some element-player in S, second, g, St cannot be in S at the same time.

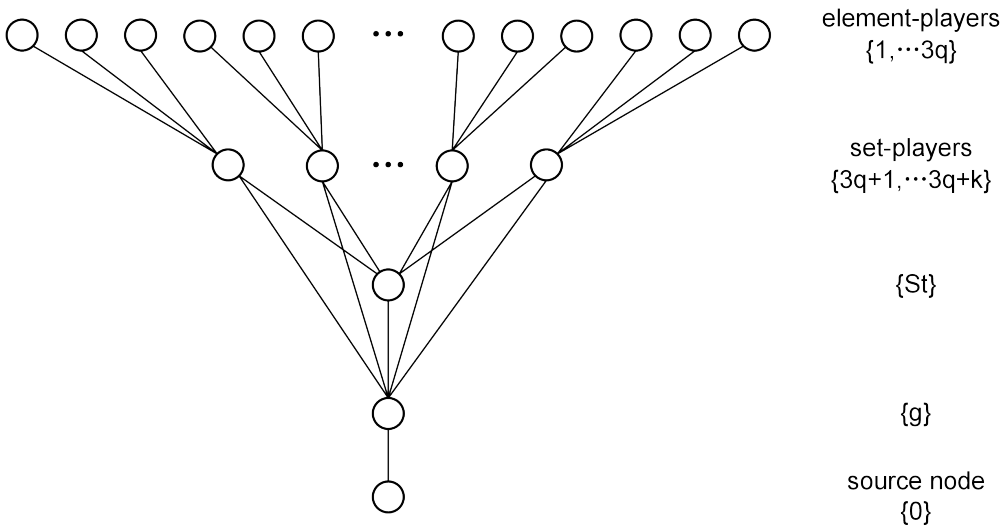


Figure 3.3: (N,E) corresponding to the MST game induced by X3C

Any path from an element-player to 0 in (N, E) has to visit a covering set-player and g. Since the weights of edges not in E are induced, we may assume w.l.o.g that $g \in S$ and S contains a covering set-player for each element-player in S. (Otherwise, there exists such S with bigger $x(S)$ and we can consider this set instead). So $St \notin S$. Consider the cost and x over S:

$$c(S) = (q+1)|S \cap \{1, \cdots, 3q\}| + (q+1)|S \cap \{3q+1, \cdots, 3q+k\}| + 2q - 1 \tag{3.7}$$

$$x(S) = (q+2)|S \cap \{1, \cdots, 3q\}| + q|S \cap \{3q+1, \cdots, 3q+k\}| \tag{3.8}$$

By Equation (3.7) and Equation (3.8), we can get:

$$0 < x(S) - c(S) = |S \cap \{1, \cdots, 3q\}| - |S \cap \{3q+1, \cdots, 3q+k\}| - 2q + 1 \tag{3.9}$$

which implies $|S \cap \{3q+1, \cdots, 3q+k\}| \leq q$.

By the assumption of S, every set-player covers at most 3 element-players, thus we have:

$$|S \cap \{1, \cdots, 3q\}| \leq 3|S \cap \{3q+1, \cdots, 3q+k\}| \tag{3.10}$$

Together with Equation (3.9) this yields $2|S \cap \{3q+1, \cdots, 3q+k\}| > 2q-1$, which imples $|S \cap \{3q+1, \cdots, 3q+k\}| \geq q$ and hence $|S \cap \{3q+1, \cdots, 3q+k\}| = q$. Using Equation (3.9) we get:

$$0 < |S \cap \{1, \cdots, 3q\}| - 3q + 1 \tag{3.11}$$

So we can conclude that $|S \cap \{1, \cdots, 3q\}| \geq 3q$ and thus $\{1, \cdots, 3q\} \subseteq S$. By assumption, S contains a covering set-player for each of its element-players, and $St \notin S$ hence $S \subsetneqq N$. These computations imply that the set-players in S must form an exact cover of A.

On the other hand, if F admits an exact 3-cover, the coalition S consisting of all element-players, some set-players which form an exact cover and the guardian satisfies:

$$c(S) = 3q(q+1) + q(q+1) + (2q-1) = 4q^2 + 6q - 1 \tag{3.12}$$

$$x(S) = 3q(q+2) + q^2 = 4q^2 + 6q \tag{3.13}$$

hence $x(S) > c(S)$ and $S \subsetneqq N$, which implies that x is not in the core of the MST game.

As a result, x is not in the core of this MST game if and only if F contains an exact cover. Since X3C is one of the six basic NP-complete problems [37]. Testing membership in the almost core $AC_{\langle N,c\rangle}$ for minimum cost spanning tree games is coNP-hard. □

Theorem 3.3.3. *Computing an optimal non-negative almost core allocation in* (3.1) *for MST games is NP-hard.*

Proof. We have proved the almost core optimization problem (2.1) for MST games is NP-hard in Proposition 3.3.1. The claim follows by showing that problem (2.1) can be reduced in polynomial time to problem (3.1). The reduction works as follows. Given a MST game instance of (2.1) with edge weights w, we define a new MST game instance by $w'(e) \coloneqq w(e) + M$, $e \in E$, for some large enough constant M, hence we can get $c'(S) = c(S) + |S| \cdot M$, $S \subsetneqq N$ and $c'(N) = c(N) + n \cdot M$.

Now consider an optimal solution x' to problem (3.1) for cost function c', and define $x \coloneqq x' - (M, \ldots, M)$. We have $x(S) = x'(S) - |S| \cdot M \leq c'(S) - |S| \cdot M = c(S)$ for all $S \subsetneqq N$, so x is feasible for problem (2.1).

We claim that x is also optimal for problem (2.1). This is true since for *any* solution $\tilde{x}$ that is optimal for (2.1) , we have $y' \coloneqq \tilde{x} + (M, \ldots, M) \geq 0$ for large enough M, and $y'(S) = \tilde{x}(S) + |S| \cdot M \leq c(S) + |S| \cdot M = c'(S)$, so y' is feasible for

(3.1) with cost function c'. Hence, assume x is not optimal for (2.1) so $\tilde{x}(N) > x(N)$, we know $y'(N) = \tilde{x}(N) + n \cdot M > x(N) + n \cdot M = x'(N)$, and it yields the contradiction $y'(N) > x'(N)$.

Finally, observe in problem (2.1) we maximize $x(N)$, hence for any optimal solution $\tilde{x}$ there exists $M > 0$ so that $\tilde{x}_i \geq -M$ for all $i \in N$, e.g. one can see that $M \coloneqq n \cdot \max_{j \in N} c(\{j\})$ suffices. This is true because for an optimal solution $\tilde{x}$, for all $i \in N$, some $S \ni i$ exists so that constraint $x(S) \leq c(S)$ is tight, and $\tilde{x}_i \leq c(\{i\})$ for all $i \in S$. Hence we can get $x'_i = \tilde{x}_i + M = \tilde{x}(S) - \sum_{j \in S \setminus \{i\}} \tilde{x}_j + M \geq c(S) - \sum_{j \in S \setminus \{i\}} c(\{j\}) + M \geq 0$ because $c(S) \geq 0$. □

3.4 Algorithm to Compute Almost Core Allocations on MST games

For minimum cost spanning tree games, we developed a tight 2-approximation algorithm to compute the $AC_{\langle N,c \rangle}$ optimum in polynomial time for the non-negative case (3.1).

3.4.1 A 2-approximation Algorithm on the Almost Core Allocations

We propose the following polynomial time algorithm to compute an approximately optimal almost core allocation. For notational convenience, let us define for all $i = 1, \ldots, n$,

$$N_i \coloneqq N \setminus \{i\} .$$

Algorithm 1: Approximation algorithm for the almost core maximization problem (3.1) for MST games

Input: Players N, edge set E of complete graph on $N \cup \{0\}$ and edge weights $w : E \to \mathbb{R}_{\geq 0}$

Output: Almost core allocation x.

Initialize $I_0 \coloneqq \{0\}$ and $T \coloneqq \emptyset$.
for $k = 1, 2, \ldots, n$ **do**
  Let $i \in I_{k-1}$, $j \in N \setminus I_{k-1}$ with minimum $w(i,j)$ (among those i, j).
  Let $I_k \coloneqq I_{k-1} \cup \{j\}$ and augment the tree $T \coloneqq T \cup \{(i,j)\}$.
  Assign player j the cost share $x_j \coloneqq w(i,j)$.
end
Let $\ell \in I_n \setminus I_{n-1}$ be the last assigned player.
Update player ℓ's cost share $x_\ell \coloneqq \min_{k \in N_\ell} \{c(N_k) - x(N \setminus \{k, \ell\})\}$.

The backbone of Algorithm 1 is effectively Prim's algorithm to compute a minimum cost spanning tree [38]. The additional line 5 yields the core allocation by Granot and Huberman [12], which we extend by adding lines 7 and 8. Let us first collect some basic properties of Algorithm 1. Henceforth, we assume w.l.o.g. that the players get assigned their cost shares in the order $1, \ldots, n$ (so that $\ell = n$ in lines 7 and 8). We denote by x^{ALG} a solution computed by Algorithm 1.

In order to show how the algorithm functions, Example 3.1.2 walks through the algorithm step by step:

Step 1: $I_0 = \{0\}$ and $T = \emptyset$. Find the edge with the minimum weight to connect I_0 and $N \setminus I_0$. $I_1 = \{0, 1\}$ and add edge $(0, 1)$ to the tree $T = \{(0, 1)\}$. Assign player 1 with the weight of edge $(0, 1)$: $x_1 = 4$.

Step 2: Find the edge with the minimum weight to connect I_1 and $N \setminus I_1$. $I_2 = \{0, 1, 2\}$ and augment edge $(1, 2)$ to the tree $T = \{(0, 1), (1, 2)\}$. Assign player 2 with the weight of edge $(1, 2)$: $x_2 = 3$.

Step 3: Find the edge with the minimum weight to connect I_2 and $N \setminus I_2$. $I_3 = \{0, 1, 2, 3\}$ and augment edge $(2, 3)$ to the tree $T = \{(0, 1), (1, 2), (2, 3)\}$. Assign player 3 with the weight of edge $(2, 3)$: $x_3 = 2$.

Step 4: Assign the last player $I_3 \setminus I_2 = 3$ with a cost share $x_3 = \min_{k \in N_3} \{c(N_k) - x(N \setminus \{k, 3\})\} = 6$.

The allocation of Algorithm 1 is $x^{\mathsf{ALG}} = (4, 3, 6)$ with a total value of 13. While the almost core optimal value is 14 with the allocation $x^{\mathsf{OPT}} = (3, 4, 7)$, which is better than the result of the algorithm, the algorithm has improved the core allocation by 4. Actually, even though the gap between the core and the almost core optimum can be large, Algorithm 1 has a performance guarantee of 2. This will be demonstrated later in this chapter.

3.4.2 Correctness of the Algorithm

Lemma 3.4.1. *We have that $x^{ALG}(I_k) = c(I_k)$ for all $k = 1, \ldots, n-1$.*

Proof. The line 3 up to line 5 are the steps of forming a minimum spanning tree by Prim's [38] algorithm, so $x^{\mathsf{ALG}}(I_k)$ equals the cost of the minimum spanning tree on the vertex set $\{0, 1, \ldots, k\}$, which is $c(I_k)$ in minimum cost spanning tree games. □

Lemma 3.4.2. *For all $S \subseteq N_n$ we have $x^{ALG}(S) \leq c(S)$.*

Proof. The proof follows by [12, Thm. 3] and is given here for completeness. Let $S \subseteq N_n$ and assume S is not the empty set. Let Γ_S be a minimum cost spanning tree on vertices $S \cup \{0\}$. Define p_i as the predecessor of $i \in N_n$ so that edge (p_i, i) is added in Algorithm 1 and define the edge set $\bar{E}$ by $\bar{E} := \{(p_i, i) \mid i \in N_n \setminus S\}$. Note that p_i is the first vertex after i on the unique $(i, 0)$-path in Γ_N.

From the definition, we know that if an edge $(i, j) \in \bar{E}$, then either $i \in N_n \setminus S$ or $j \in N_n \setminus S$, whereas if $(i, j) \in \Gamma_S$, then both $i \in S \cup \{0\}$ and $j \in S \cup \{0\}$. Therefore, the edge set $\Gamma_S \cup \bar{E}$ forms a connected graph on $N_n \cup \{0\}$. The weight of all edges in the connected graph $< N_n \cup \{0\},\ \Gamma_S \cup \bar{E} >$ is:

$$w(\Gamma_S \cup \bar{E}) = c(S) + \sum_{i \in N_n \setminus S} w(p_i, i) = c(S) + x^{\mathsf{ALG}}(N_n \setminus S)$$

Since $c(N_n)$ is the cost of the *minimum* cost spanning tree on $N_n \cup \{0\}$, we know $c(N_n) \leq c(S) + x^{\mathsf{ALG}}(N_n \setminus S)$.

By Lemma 3.4.1, $x^{\mathsf{ALG}}(N_n) = c(N_n)$, so we can conclude:

$$x^{\mathsf{ALG}}(S) = c(N_n) - x^{\mathsf{ALG}}(N_n \setminus S) \leq c(S)$$

As a result, $x^{\mathsf{ALG}}(S) \leq c(S)$ □

The same procedures in the proof of Lemma 3.4.1 and Lemma 3.4.2 also show that by lines 3- 5 in Algorithm 1, we can get a feasible core allocation.

Lemma 3.4.3. *Suppose $x^{ALG}(S) > c(S)$ for some set S with $n \in S \subsetneq N$. Then there is a superset $T \supseteq S$ with $|T| = n - 1$ such that $x^{ALG}(T) > c(T)$.*

Proof. Recall that the players got assigned their cost shares in order $1, \ldots, n$. Define $k := max\{i \mid i \notin S\}$ to be the largest index of a player not in S. Let $i_1, \ldots, i_\ell$ be the set of players so that $N_k = N \setminus \{k\} = S \cup \{i_1, \ldots, i_\ell\}$ and w.l.o.g. $i_1 < \cdots < i_\ell$. We show that $x^{\mathsf{ALG}}(S) > c(S)$ implies $x^{\mathsf{ALG}}(S \cup \{i_1\}) > c(S \cup \{i_1\})$. Then repeating the same argument, we inductively arrive at the conclusion that $x^{\mathsf{ALG}}(N_k) > c(N_k)$. So observe that

$$x^{\mathsf{ALG}}(S \cup \{i_1\}) = x^{\mathsf{ALG}}(S) + x_{i_1} > c(S) + x_{i_1}\,,$$

and $c(S)$ is the cost of a minimum cost spanning tree for S, call it MST(S). Moreover, as $i_1 \neq n$, x_{i_1} is the cost of the edge, call it e, that the algorithm used to connect player i_1. We claim that MST$(S) \cup \{e\}$ is a tree spanning vertices $S \cup \{0, i_1\}$, hence $c(S) + x_{i_1}$ is the cost of some tree spanning $S \cup \{0, i_1\}$. Then, as required we get

$$x^{\mathsf{ALG}}(S \cup \{i_1\}) > c(S) + x_{i_1} \geq c(S \cup \{i_1\})\,,$$

because $c(S \cup \{i_1\})$ is the cost of a *minimum cost* tree spanning $S \cup \{0, i_1\}$. If MST$(S) \cup \{e\}$ was not a spanning tree for vertices $S \cup \{0, i_1\}$, then edge e would connect i_1 to some vertex outside S, but this contradicts the choice of i_1 as the vertex outside S. □

Lemma 3.4.4. *We have* $x^{ALG} \geq 0$.

Proof. Recall that in minimum cost spanning tree games [12, 39], the weight of edges are non-negative. Since Algorithm 1 computes the allocation for players in line 5 by the edge weight of the first edge on the unique path to 0, there is $x_k^{\mathsf{ALG}} \geq 0$ for all $k = 1, 2, \cdots, n-1$. So we only need to argue about x_n^{ALG}. To that end, note that an equivalent definition of x_n^{ALG} in line 8 of the algorithm is

$$\text{max. } x_n \text{ s.t. } x_n \leq c(N_k) - x^{\mathsf{ALG}}(N \setminus \{k, n\}) \text{ for all } k = 1, \ldots, n-1\,. \tag{3.14}$$

We claim that $\tilde{x}_n := c(N) - c(N_{n-1}) \geq 0$ is a feasible solution to this maximization problem, hence the actual value of x_n^{ALG} after the update in line 8 can only be larger, and therefore in particular it is non-negative. First, note that indeed, $\tilde{x}_n \geq 0$, as this is the cost of the last edge that Prim's algorithm uses to connect the final vertex n to the minimum cost spanning tree. That $\tilde{x}_n$ is feasible in (3.14) follows from the fact that $\tilde{x}_n$ is the cost share that is assigned to player n in the core allocation of [12]. Indeed, letting $\tilde{x}$ be equal to x except for $\tilde{x}_n = c(N) - c(N_{n-1})$, we have that $\tilde{x}$ is precisely the cost allocation as proposed in [12]. By the fact that this yields a core allocation, we have that $\tilde{x}(S) \leq c(S)$ for all $S \subseteq N$, so in particular for all $k = 1, \ldots, n-1$,

$$\tilde{x}_n + x^{\mathsf{ALG}}(N \setminus \{k, n\}) = \tilde{x}(N_k) \leq c(N_k)\,,$$

and hence the claim follows. □

Theorem 3.4.5. *Algorithm 1 yields a feasible solution.*

Proof. Denote by x^{ALG} a solution by Algorithm 1. For $S : n \notin S$, this follows from Lemma 3.4.1. For $S : n \in S$, assume $x(S) > c(S)$. Then Lemma 3.4.3 yields that there exists some $N_k = N \setminus k$ with $n \in N_k$ satisfies $x^{\mathsf{ALG}}(N_k) > c(N_k)$. However by definition of x_n in Line 8 of the algorithm, we have for all $k = 1, \ldots, n-1$

$$x_n^{\mathsf{ALG}} \leq c(N_k) - x^{\mathsf{ALG}}(N \setminus \{k, n\})\,,$$

which yields a contradiction to $x^{\mathsf{ALG}}(N_k) > c(N_k)$. □

3.5 Performance of the Algorithms

Theorem 3.5.1. *Algorithm 1 is a 2-approximation algorithm for the almost core maximization problem* (3.1) *for minimum cost spanning tree games, and this bound is tight.*

Proof. To show that the performance guarantee is indeed 2, let x^{OPT} be some optimal solution to the almost core maximization Problem (3.1). Let $k^\star \in N_n$ be the

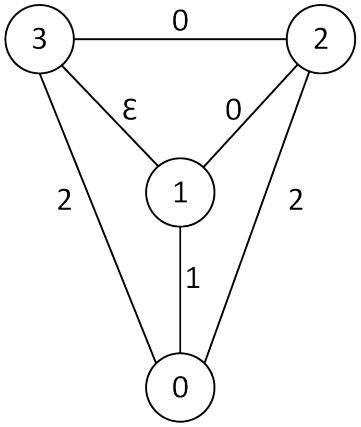


Figure 3.4: MST game showing that the analysis of Algorithm 1 cannot be improved.

index for which the minimum in line 8 is attained. Observe that x_n^{ALG} is updated such that $x^{\mathsf{ALG}}(N_{k^\star}) = c(N_{k^\star})$ holds. Then by non-negativity of x^{OPT} and because of Lemma 3.4.4,

$$x_n^{\mathsf{OPT}} \leq x^{\mathsf{OPT}}(N_{k^\star}) \leq c(N_{k^\star}) = x^{\mathsf{ALG}}(N_{k^\star}) \ \leq x^{\mathsf{ALG}}(N)\,.$$

Moreover, by definition of x^{ALG}, we have $x^{\mathsf{ALG}}(N_n) = c(N_n)$, and by Lemma 3.4.4,

$$x^{\mathsf{OPT}}(N_n) \leq c(N_n) = x^{\mathsf{ALG}}(N_n) \leq x^{\mathsf{ALG}}(N)\,.$$

Hence we get $x^{\mathsf{OPT}}(N) = x_n^{\mathsf{OPT}} + x^{\mathsf{OPT}}(N_n) \leq 2x^{\mathsf{ALG}}(N)$. To see that the performance bound 2 is tight for Algorithm 1, consider the instance in Figure 3.4.

Here, Algorithm 1 computes the solution $x^{\mathsf{ALG}} = (1, 0, \varepsilon)$ with value $1 + \varepsilon$, as the order in which players get assigned their cost shares is $1, 2, 3$, and in line 8 of the algorithm we get $x_3^{\mathsf{ALG}} = c(\{1, 3\}) - x_1 = (1 + \varepsilon) - 1 = \varepsilon$. An almost core optimum would be $x^{\mathsf{OPT}} = (0, 1, 1)$ with value 2. □

Chapter 4

Almost Core Allocations on Constrained MST Structures

For minimum cost spanning tree games, we analyze the gap between the almost core optimum and the total cost of the grand coalition under the assumption of different graph structures for minimum cost spanning tree games. In addition, some observations are made about their properties. For clarity of expression, we use round brackets to represent undirected edges, that is, $(u, v) = (v, u)$.

4.1 When Some MST is Not a Path

There is a class of MST games where there is a minimum cost spanning tree that is not a path in the corresponding graph:

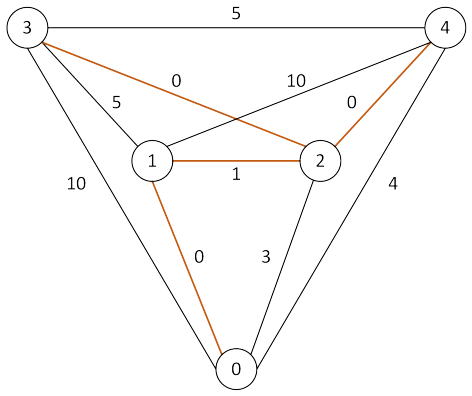
$$\exists T \in \mathcal{T}(N), T \text{ is not a path} \tag{4.1}$$

Where $\mathcal{T}(N)$ is the set of spanning trees induced by vertex set $N \cup \{0\}$. For such games, the almost core optimum is bounded by $2c(N)$.

Example 4.1.1. *Let $N = \{1, 2, 3, 4\}$. Consider the graph in Figure 4.1. The weights of the edges are as in the figure, and the MST is marked in orange. The table on the right-hand side shows the costs of various coalitions in MST game.*

Theorem 4.1.2. *For games that satisify Equation* (4.1)*, the almost core optimum $x^{OPT}(N)$ of linear program (3.1) with non-negative domain is bounded and $x^{OPT}(N) \leq 2c(N)$.*

Proof. Let T be MST as in Example 4.1.1. Since the minimum cost spanning tree of the graph is not a path, in MST graph $\langle N, E(T)\rangle$, there exists node incident to more than 3 edges, take such node with the smallest index (according to the order players



S	$c(S)$	S	$c(S)$
{1}	0	{2,4}	3
{2}	3	{3,4}	9
{3}	10	{1,2,3}	1
{4}	4	{1,2,4}	1
{1,2}	1	{1,3,4}	9
{1,3}	5	{2,3,4}	3
{1,4}	4	{1,2,3,4}	1
{2,3}	3		

Figure 4.1: MST game with four players and the costs of coalitions

got assigned their cost shares in line 3 up to line 5 for Algorithm 1) be v, so the degree of v in T satisfies $d(v) \geq 3$. Notice there is a unique $(0, v)$-path p_0 in T, and we can partition $T \setminus p_0$ into two disjoint subtrees, say T_1 and T_2, that meet at v. Let the vertices which are on $p_0 \cup T_1$ be P_1, the vertices which are on $p_0 \cup T_2$ be P_2. In Example 4.1.1, $P_1 = \{1, 2, 3\}$ and $P_2 = \{1, 2, 4\}$.

Note that $P_1 \cup P_2 = N$. Construct two MST games corresponding to P_1 and P_2: $\langle P_1, p_0 \cup T_1 \rangle$ and $\langle P_2, p_0 \cup T_2 \rangle$, and consider following 2 linear programs on P_1 and P_2:

$$\begin{aligned} z_1 = \max \quad & x(P_1) \\ s.t. \quad & x(S) \leq c(S), \quad \forall S \subseteq P_1 \\ & x \in \mathbb{R}^n_{\geq 0} \end{aligned} \tag{4.2}$$

$$\begin{aligned} z_2 = \max \quad & x(P_2) \\ s.t. \quad & x(S) \leq c(S), \quad \forall S \subseteq P_2 \\ & x \in \mathbb{R}^n_{\geq 0} \end{aligned} \tag{4.3}$$

We know the core of an MST game is nonempty, so any optimum solution $x^\star(P_i)$ is in the core of MST game $\langle P_i, p_0 \cup T_i \rangle$ and the optimum value z_i is equal to $c(P_i)$. Let x^c be the core allocation by Granot and Huberman's [12] algorithm. Since we construct $\langle P_1, p_0 \cup T_1 \rangle$ and $\langle P_2, p_0 \cup T_2 \rangle$ by maintaining the core allocation edges of P_1 and P_2, we have $x^{\mathsf{OPT}}(P_1) \leq c(N), x^{\mathsf{OPT}}(P_2) \leq c(N)$, so there is:

$$z_1 + z_2 = x^{\mathsf{OPT}}(P_1) + x^{\mathsf{OPT}}(P_2) \leq 2c(N)$$

Consider the following linear programming:

$$\begin{aligned} z_3 = \max \quad & x(P_1) + x(P_2) \\ s.t. \quad & x(S) \leq c(S), \quad \forall S \subseteq P_1 \text{ or } S \subseteq P_2 \\ & x \in \mathbb{R}^n_{\geq 0} \end{aligned} \tag{4.4}$$

The polyhedron of (4.4) lies in the intersection of the two polyhedra of (4.2) and (4.3) and there are more constraints for (4.4), so there is $z_3 \leq z_1 + z_2 \leq 2c(N)$.

For the following linear programming:

$$\begin{aligned} z_4 = \max \quad & x(N) + x(P_1 \cap P_2) \\ s.t. \quad & x(S) \leq c(S), \qquad \forall S \subsetneq N \\ & x \in \mathbb{R}^n_{\geq 0} \end{aligned} \tag{4.5}$$

Since $P_1, P_2 \subsetneq N$, the polyhedron constraints (4.5) lies in the polyhedron of (4.4) and there are more constraints for (4.5). By non-negativity, we know $z_3 \geq z_4 \geq x^{\mathsf{OPT}}(N)$.

In conclusion, there is: $x^{\mathsf{OPT}}(N) \leq z_3 \leq 2c(N)$ when MST of the graph is not a path. □

However, for a MST game where every MST of the graph is a path, the almost core optimum $x^{\mathsf{AC}}(N)$ can be arbitrarily larger than $c(N)$, even when we require that $x \geq 0$ (see 3.2).

4.2 (0,M)-MST games

We refer to a MST game as a $(0, M)$-*MST game* when the following holds: the edges of the corresponding complete graph have only two possible weights: 0 and $M(M > 0)$, and the edges with weight 0 form a unique MST and there is only one edge with weight 0 incident to the source node (if not, $x^{\mathsf{AC}} \leq 2c(N) = 0$). That is, the corresponding complete graph with T being a spanning tree graph (the graph of the spanning tree for G) satisfies:

$$w(e) = \begin{cases} 0 & \textit{if } e \in E(T) \\ M & \forall e \in E(G \setminus T) \end{cases} \tag{4.6}$$

$$|\delta(0) \cap T| = 1 \tag{4.7}$$

Define the parent of a node v to be the node connected to v on the unique path from v to the source node 0 in $(V, E(T))$. A child of a node v is a node of which v is the parent.

Algorithm 2: Algorithm of $(0, M)$-MST game for the almost core maximization problem (3.1)

Input: An $(0, M)$-MST game network $G = (V, E, w)$

Output: $(0, M)$-Almost core allocation x

Let $E(T)$ be the edge set with all edges of weight 0 in G.

$M :=$ the largest weight in G.

for $v = 1, 2, \ldots, n$ **do**

  $K_v :=$ number of child nodes of node v.

  $x_v := -(K_v - 1)M$.

end

Output the almost core allocation x for $(0, M)$-MST game.

Define a child tree $\hat{T}$ of v as a subtree rooted from node v and contain all the child nodes of $\hat{T}$. In Example 4.2.4,$\{2, 4, 5, 6\}$ is a child tree, but $\{3, 8\}$ is not a child tree because 7 is a child of 3, but it is not in the set.

Lemma 4.2.1. *We have* $x(\hat{T}) = M$*.*

Proof. Since for any child tree $\hat{T}$, $x(\hat{T}) = kM - (k-1)M = M$. By induction, If the allocation value to a child tree $\hat{T}$ of v is M, let v' be a parent of v, and then the allocation value to the child tree $\hat{T}'$ of v' is also M As a result, for any child tree $\hat{T}$ in graph G, there is $x(\hat{T}) = M$. □

Theorem 4.2.2. *The result of Algorithm 2 is in the almost core* $AC_{\langle N,c\rangle}$*.*

Proof. For any $S \subsetneq N$, there exists a partition of S induced by MST: $S = S_1 \cup ... \cup S_q$, where every cell S_i is a maximal connected subset in MST, that is, all nodes in S_i are connected in MST while S_i and S_j are disjoint in MST. Take the player induced by the node adjacent to the source node in MST to be player 1.

If $\{1\} \notin S$, to connect the source node and every unconnected cell, $c(S) = qM$. By the allocation rule of Algorithm 2, the largest allocation to any connected subset is M, that is, for any connected cell S_i, $x(S_i) \leq M$ and there is $x(S) = \sum_{i=1}^{q} x(S_i) \leq qM = c(S)$.

Otherwise, if $\{1\} \in S$, $c(S) = (q-1)M$. Suppose $\{1\} \in S_1$, Since $S_1 \subseteq S \subsetneq N$, $x(S_1) \leq 0$. In this case, $x(S) = \sum_i x(S_i) \leq (q-1)M = c(S)$.

□

Theorem 4.2.3. *The result of Algorithm 2 is an optimal almost core allocation for any* $(0, M)$*-MST game.*

Proof. Take player 1 to be the player node incident to the source node by MST. Since there is only one edge with weight 0 incident to the source node in $(0, M)$-MST game, $c((N \setminus \{1\}) \leq M$ to connect $c((N \setminus \{1\}) \leq M$ with 0 , For any almost

core allocation $\hat{x}$ in $AC_{\langle N,c\rangle}$:

$$\begin{aligned}\hat{x}(N) &= \hat{x}(1) + \hat{x}(N \setminus \{1\}) \\ &\leq c(1) + c((N \setminus \{1\}) \\ &\leq M\end{aligned}$$

We have proved the allocation in Algorithm 2 distribute a total value M, that is $x(N) = M$, and it is in $AC_{\langle N,c\rangle}$. Hence it is one optimum almost core allocation for $(0, M)$-MST games. □

Example 4.2.4. *Consider a $(0, M)$-MST game with 10 players. Figure 4.2 is the corresponding complete graph when only edges with weight 0 are shown in the figure, all other edges have a weight of M. Note that all edges with weight 0 form the unique MST of this graph. In this example, the solution computed by algorithm Algorithm 2 is marked in blue in Figure 4.2 with a total value of M.*

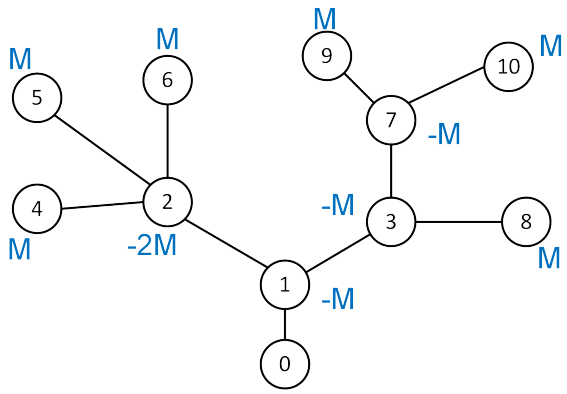


Figure 4.2: MST graph $(V), E(T))$ of a $(0, M)$-MST game example

$(0, M)$-MST game can be applied to the real-life example of a cable-laying problem where an old cable system already exists and it costs M to lay a new cable between two buildings. The electricity supplier can get a maximum benefit by the allocation of algorithm 2.

Chapter 5

Almost Core Allocations on Value Games

This chapter adapts the almost core concept to value games, in particular the minimum cost spanning tree value games, for which we also propose a 2-approximation algorithm.

5.1 Value Games

We described the characteristic functions of a cooperative game to be cost functions in Section 1.1. However, when the purpose of the game is to distribute payoff to every single player, we can define the characteristic value function to be $v : 2^N \rightarrow \mathbb{R}_{\geq 0}$. These kinds of games are called value games or profit games.

For value games, the almost core allocation is defined as follows:

$$\begin{aligned} \min \quad & x(N) \\ s.t. \quad & x(S) \geq v(S), \quad \forall S \subsetneq N \\ & x \in \mathbb{R}^n \end{aligned} \tag{5.1}$$

The difference between 5.1 and the almost core allocation in the previous chapters is that coalition values are induced by the costs of coalitions for a cost sharing game, now they represent payoffs.

5.2 Almost Core Allocation for MST Value Games

To convert minimum cost spanning tree games to values games, recall the example of 3 cities and the source, but in this case, every city will get a payoff for the service. The cost of connecting every two buildings still exists. The value of every coalition is the saving they can obtain by forming the coalition.

In conclusion, for MST value games, define the characteristic value function as $v(S) = \sum_{i \in S} c_i - c(S)$, and $c(S)$ maintains to be the definition in Section 3.1, that is the cost of a minimum spanning tree on the subgraph induced by vertex set $S \cup \{0\}$.

Notice that in value games, every feasible allocation in the (almost) core is non-negative since $x(i) \geq v(i) = c_i - c_i = 0$.

5.2.1 Computational Complexity

Consider linear optimization over the polyhedra

$$AC_{\langle N,v \rangle} \quad \text{and} \quad P_{\langle N,v \rangle} \coloneqq \{x \in \mathbb{R}^n : x(S) \geq v(S) \; \forall S \subseteq N\}.$$

as well as optimization over $P_{\langle N,v \rangle} \cap \mathbb{R}^n_{\geq 0}$ and $AC_{\langle N,v \rangle} \cap \mathbb{R}^n_{\geq 0}$ for families of value games $\langle N, v \rangle$. Note that the core of the value games is the set of optimal solutions when maximizing $\mathbb{1}$ over $P_{\langle N,v \rangle}$. Also note that whenever the core of a game $\langle N, v \rangle$ is empty, so no x exists with $x(N) \leq v(N)$ and $x(S) \geq v(S) \; \forall S \subseteq N$, we have that $x(N) > v(N)$ is implied by the set of constraints $x(S) \geq v(S)$, $S \subsetneq N$, which in turn implies $P_{\langle N,v \rangle} = AC_{\langle N,v \rangle}$. For games with non-empty core, we get the following correspondence between the optimization problems for the two polyhedra.

Corollary 5.2.1. *For a family of games $\langle N, v \rangle$, linear optimization problems over $AC_{\langle N,v \rangle}$ can be solved in polynomial time if and only if linear optimization problems over $P_{\langle N,v \rangle}$ can be solved in polynomial time.*

Proof. The proof is similar to the proof of Theorem 2.3.1 for cost games by converting the cost game to the corresponding value game. We can use the equivalence of optimization and separation problems. Separation over $P_{\langle N,v \rangle}$ can be reduced to separation over $AC_{\langle N,v \rangle}$ plus an explicit check of a single inequality $x(N) \geq v(N)$, and we can construct the separation oracle for $AC_{\langle N,v \rangle}$ by solving the separation oracle for $P_{\langle N,v \rangle}$ as in the proof of Theorem 2.3.1. □

Corollary 5.2.2. *Consider a family of value games $\langle N, v \rangle$ for which deciding non-emptiness of the core is (co)NP-hard. Then finding an optimal almost core allocation is also (co)NP-hard.*

Proof. The proof is identical to the proof of Proposition 2.3.4. □

5.3 Algorithm for MST Value Games

5.3.1 Algorithm and Example

Algorithm 3: Approximation algorithm for the almost core minimization problem (2.1) for MST value games

Input: Players N, edge set E of complete graph on $N \cup \{0\}$ and edge weights $w : E \to \mathbb{R}_{\geq 0}$

Output: Almost core allocation x.

1 Initialize $I_0 \coloneqq \{0\}$ and $T \coloneqq \emptyset$.
2 **for** $k = 1, 2, \ldots, n$ **do**
3 Let $i \in I_{k-1}$, $j \in N \setminus I_{k-1}$ with minimum $w(i, j)$ (among those i, j).
4 Let $I_k \coloneqq I_{k-1} \cup \{j\}$ and augment the tree $T \coloneqq T \cup \{(i, j)\}$.
5 Assign player j the value share $x_j \coloneqq c_j - w(i, j)$.
6 **end**
7 Let $\ell \in I_n \setminus I_{n-1}$ be the last assigned player.
8 Update player ℓ's cost share $x_\ell \coloneqq \max_{k \in N_\ell} \{v(N_k) - x(N \setminus \{k, \ell\})\}$.

W.l.o.g. assume the players get assigned their payoffs in the order $1, \ldots, n$ (so that $\ell = n$ in lines 7 and 8). The algorithm Algorithm 3 is similar to the algorithm Algorithm 1, but we assign $x_j \coloneqq c_j - w(i, j)$ instead of $w(i, j)$ in line 5 and try to update the minimal feasible amount to the last player in line 8. We denote by x^{ALG} a solution computed by Algorithm 3.

In order to show how the algorithm functions, the example corresponding to Figure 5.1 walks through the algorithm step by step:

Step 1: $I_0 = \{0\}$ and $T = \emptyset$. Find the edge with the minimum weight to connect I_0 and $N \setminus I_0$. $I_1 = \{0, 1\}$ and augment edge $(0, 1)$ to the tree $T = \{(0, 1)\}$. Assign player 1 with $x_1 = c_1 - w(0, 1) = 0$.

Step 2: Find the edge with the minimum weight to connect I_1 and $N \setminus I_1$. If there are two same weights, choose one arbitrary. Here one possible result may be $I_2 = \{0, 1, 2\}$. Augment edge $(1, 2)$ to the tree $T = \{(0, 1), (1, 2)\}$. Assign player 2 with $x_2 = c_2 - w(1, 2) = 3$.

Step 3: Find the edge with the minimum weight to connect I_2 and $N \setminus I_2$. $I_3 = \{0, 1, 2, 3\}$ and augment edge $(2, 3)$ to the tree $T = \{(0, 1), (1, 2), (2, 3)\}$. Assign player 3 with $x_3 = c_3 - w(2, 3) = 0$.

Step 4: Assign the last player $I_3 \setminus I_2 = 3$ with $x_3 = \max_{k \in N_3} \{v(N_k) - x(N \setminus \{k, 3\})\} = 3$.

The allocation of Algorithm 3 is $x^{\mathsf{ALG}} = (0, 3, 3)$ with a total value of 6. Actually, Algorithm 3 also has a performance guarantee of 2. This will be demonstrated later in this chapter.

5.3.2 Correctness of the Algorithm

Lemma 5.3.1. *We have that $x^{ALG}(I_k) = v(I_k)$ for all $k = 1, \ldots, n-1$, and for all $S \subseteq \{1, \ldots, n-1\}$ we have $x^{ALG}(S) \geq v(S)$.*

Proof. The first claim follows because $x^{\mathsf{ALG}}(I_k) = \sum_{i \in I_k}(c_i - w(i,j))$ and we have $\sum_{i \in I_k} w(i,j) = c(I_k)$ because the $w(i,j)$ follows Prim's[38] algorithm which computes a minimum cost spanning tree on the vertex set $\{1, \ldots, n-1\}$. As a result, we can conclude $x^{\mathsf{ALG}}(I_k) = \sum_{i \in I_k} c_i - c(I_k) = v(I_k)$.

Now we prove the second claim. Similar to the proof of (3.4.2), let $S \subseteq N_n$ and assume S is not the empty set. Let Γ_S be a minimum cost spanning tree on vertices $S \cup \{0\}$. Define p_i as the predecessor of $i \in N_n$ so that edge (p_i, i) is added in Algorithm 1 and define the edge set $\bar{E}$ by $\bar{E} := \{(p_i, i) \mid i \in N_n \setminus S\}$. Note that p_i is the first vertex other than i on the unique $(i, 0)$-path in Γ_N.

From the definition, we know that if an edge $(i,j) \in \bar{E}$, then either $i \in N_n \setminus S$ or $j \in N_n \setminus S$, whereas if $(i,j) \in \Gamma_S$, then both $i \in S \cup \{0\}$ and $j \in S \cup \{0\}$. Therefore, the edge set Γ_S and $\bar{E}$ forms a connected graph on $N_n \cup \{0\}$. The weight of all edges in the connected graph $(N_n \cup \{0\},\ \Gamma_S \cup \bar{E})$ is $w(\Gamma_S \cup \bar{E}) = c(S) + \sum_{i \in N_n \setminus S} w(p_i, i)$. Since $c(N_n)$ is the cost of the *minimum* cost spanning tree on $N_n \cup \{0\}$, we know $c(N_n) \leq c(S) + \sum_{i \in N_n \setminus S} w(p_i, i)$.

To convert the cost of player sets to the value of player sets, substitude $c(S)$ by $\sum_{i \in S} c_i(S) - v(S)$ we will get:

$$\sum_{i \in N_n} c_i - v(N_n) \leq \sum_{i \in S} c_i - v(S) + \sum_{i \in N_n \setminus S} w(p_i, i) \tag{5.2}$$

Notice that $\sum_{i \in N_n \setminus S} w(p_i, i)$ is equal to $\sum_{i \in N_n \setminus S}(c_i - x_i^{\mathsf{ALG}})$, together with Equation (5.2), we can get $\sum_{i \in N_n} c_i - v(N_n) \leq \sum_{i \in S} c_i + \sum_{i \in N_n \setminus S} c_i - v(S) - x^{\mathsf{ALG}}(N_n \setminus S)$, that is:

$$v(N_n) \geq v(S) + x^{\mathsf{ALG}}(N_n \setminus S)$$

We know that $x^{\mathsf{ALG}}(N_n) = v(N_n)$, so we can conclude:

$$x^{\mathsf{ALG}}(S) = v(N_n) - x^{\mathsf{ALG}}(N_n \setminus S) \geq v(S)$$

So for all $S \subseteq \{1, \ldots, n-1\}$ we have $x^{\mathsf{ALG}}(S) \geq v(S)$. □

Lemma 5.3.2. *Suppose $x^{ALG}(S) < v(S)$ for some set S with $n \in S \subsetneq N$. Then there is a superset $T \supseteq S$ with $|T| = n - 1$ such that $x^{ALG}(T) < v(T)$.*

Proof. Recall the players got assigned their cost shares in order $1, \ldots, n$. Define $k := max\{i \mid i \notin S\}$ to be the largest index of a player not in S. Let $i_1, \ldots, i_\ell$ be the set of players so that $N_k = N \setminus \{k\} = S \cup \{i_1, \ldots, i_\ell\}$ and w.l.o.g. $i_1 < \cdots < i_\ell$. We show that $x^{\mathsf{ALG}}(S) < v(S)$ implies $x^{\mathsf{ALG}}(S \cup \{i_1\}) < v(S \cup \{i_1\})$. Then repeating the same argument, we inductively arrive at the conclusion that $x^{\mathsf{ALG}}(N_k) < v(N_k)$. So observe that

$$x^{\mathsf{ALG}}(S \cup \{i_1\}) = x^{\mathsf{ALG}}(S) + x_{i_1} < v(S) + x_{i_1} = \sum_{i \in S} c_i - c(S) + x_{i_1}\,,$$

and $c(S)$ is the cost of a minimum cost spanning tree for S, call it MST(S). Moreover, as $i_1 \neq n$, x_{i_1} is the cost of the edge, call it e, that the algorithm used to connect player i_1. We claim that MST$(S) \cup \{e\}$ is a tree spanning vertices $S \cup \{0, i_1\}$, hence $c(S) + x_{i_1}$ is the cost of some tree spanning $S \cup \{0, i_1\}$. Then, as required, we get

$$x^{\mathsf{ALG}}(S \cup \{i_1\}) < \sum_{i \in S} c_i - c(S) + x_{i_1} \leq \sum_{i \in S \cup \{i_1\}} c_i - c(S \cup \{i_1\}) = v(S \cup \{i_1\})\,,$$

because $c(S \cup \{i_1\})$ is the cost of a *minimum cost* tree spanning $S \cup \{0, i_1\}$. If MST$(S) \cup \{e\}$ was not a spanning tree for vertices $S \cup \{0, i_1\}$, then edge e would connect i_1 to some vertex outside S, but this contradicts the choice of i_1 as the vertex outside S with smallest index. □

Theorem 5.3.3. *Algorithm 3 yields a feasible solution.*

Proof. Denote by x^{ALG} a solution by Algorithm 3. For $S : n \notin S$, this follows from Lemma 5.3.1. For $S : n \in S$, assume $x(S) < v(S)$. Then Lemma 5.3.2 yields that there exists some $N_k = N \setminus k$ with $n \in N_k$ satisfies $x^{\mathsf{ALG}}(N_k) < v(N_k)$. However by definition of x_n in Line 8 of the algorithm, we have for all $k = 1, \ldots, n-1$

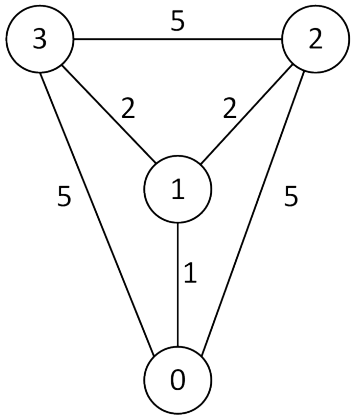
$$x_n^{\mathsf{ALG}} \geq v(N_k) - x^{\mathsf{ALG}}(N \setminus \{k, n\})\,,$$

which yields a contradiction to $x^{\mathsf{ALG}}(N_k) < v(N_k)$. It shows Algorithm 3 computes a result in $AC_{\langle N,v\rangle}$. □

5.3.3 Performance of the Algorithms

Theorem 5.3.4. *Algorithm 3 is a 2-approximation for the almost core minimization problem* (5.1) *for MST value games, and this bound is tight.*

Proof. To show that the performance guarantee is indeed 2, let x^{OPT} be some optimal solution to the almost core maximization problem (5.1). Let $k^\star \in N_n$ be the index for which the maximum in line 8 is attained. Observe that x_n^{ALG} is updated



S	$c(S)$	$v(S)$
{1}	1	0
{2}	5	0
{3}	5	0
{1,2}	3	3
{1,3}	3	3
{2,3}	10	0

Figure 5.1: MST game showing that the analysis of Algorithm 3 cannot be improved.

such that $x^{\mathsf{ALG}}(N_{k^\star}) = v(N_{k^\star})$ holds. Then by non-negativity of x^{OPT} and since $x^{\mathsf{ALG}}(i) \geq v(i) = 0$,

$$x_n^{\mathsf{ALG}} \leq x^{\mathsf{ALG}}(N_{k^\star}) = v(N_{k^\star}) \leq x^{\mathsf{OPT}}(N_{k^\star}) \leq x^{\mathsf{OPT}}(N).$$

Moreover, by Lemma 5.3.1, we have $x^{\mathsf{ALG}}(N_n) = v(N_n)$,

$$x^{\mathsf{ALG}}(N_n) = v(N_n) \leq x^{\mathsf{OPT}}(N_n) \leq x^{\mathsf{OPT}}(N)$$

Hence we get $x^{\mathsf{ALG}}(N) = x_n^{\mathsf{ALG}} + x^{\mathsf{ALG}}(N_n) \leq 2x^{\mathsf{OPT}}(N)$.

To see that the performance bound 2 is tight for Algorithm 3, consider the instance in Figure 5.1.

Here, Algorithm 3 computes the solution $x^{\mathsf{ALG}} = (0, 3, 3)$ with value 6, as the order in which players get assigned their cost shares is $1, 2, 3$, and in line 8 of the algorithm we get $x_3^{\mathsf{ALG}} = v(\{1, 3\}) - x_1 = 3$. An almost core optimum would be $x^{\mathsf{OPT}} = (3, 0, 0)$ with value 3. □

In fact, for a cost sharing game $\langle N, c\rangle$, when the characteristic function of the corresponding value game $\langle N, v\rangle$ is given by: $v(S) = \sum_{i\in S} c_i - c(S)$, given a feasible allocation x of cost sharing game, we can get a feasible allocation y of value game by:

$$y_i = c_i - x_i$$

It is feasible because $y(S) = \sum_{i\in S} c_i - x(S) \geq \sum_{i\in S} c_i - c(S) = v(S)$. However, the performance guarantee of the algorithm by this transformation may change.

Chapter 6

Conclusions

We collected some problems which we believe are interesting for further research. First, we would like to gain more insight into the computational complexity for the almost core problem (2.1) for other classes of games. Moreover, we could give a 2-approximation for MST cost games under the additional assumption that subsidies are not allowed, so $x \geq 0$, but do not have such a result when it is allowed that players are subsidized.

Furthermore, we studied the games when the grand coalition could not cooperate. Similarly, cooperative games with restricted cooperation are worth considering for further research.

Most of the results in this thesis were based on joint work with M. Uetz, M. Walter and R. Zou [40] (In preparation). While the joint work focus on cost sharing games, this thesis also studied the almost core concept of value games.

Bibliography

[1] O. Morgenstern and J. Von Neumann, *Theory of games and economic behavior*. Princeton university press, 1953.

[2] D. B. Gillies, "Solutions to general non-zero-sum games," in *Contributions to the Theory of Games, Volume IV*, ser. Annals of Mathematics Studies, A. W. Tucker and R. D. Luce, Eds. Princeton University Press, 1959, vol. 40, pp. 47–85.

[3] L. S. Shapley and M. Shubik, "Quasi-cores in a monetary economy with nonconvex preferences," *Econometrica: Journal of the Econometric Society*, pp. 805–827, 1966.

[4] M. Maschler, B. Peleg, and L. S. Shapley, "Geometric properties of the kernel, nucleolus, and related solution concepts," *Mathematics of operations research*, vol. 4, no. 4, pp. 303–338, 1979.

[5] U. Faigle and W. Kern, "On some approximately balanced combinatorial cooperative games," *Zeitschrift für Operations Research*, vol. 38, no. 2, pp. 141–152, 1993.

[6] K. Jain and M. Mahdian, "Cost sharing," in *Algorithmic game theory*, N. Nisan, T. Roughgarden, É. Tardos, and V. V. Vazirani, Eds. Cambridge University Press New York, 2007, ch. 15, pp. 385–410.

[7] Y. Bachrach, E. Elkind, R. Meir, D. Pasechnik, M. Zuckerman, J. Rothe, and J. S. Rosenschein, "The cost of stability in coalitional games," in *International Symposium on Algorithmic Game Theory*. Springer, 2009, pp. 122–134.

[8] R. Meir, Y. Bachrach, and J. S. Rosenschein, "Minimal subsidies in expense sharing games," in *International Symposium on Algorithmic Game Theory*. Springer, 2010, pp. 347–358.

[9] C. Bejan and J. C. Gómez, "Core extensions for non-balanced TU-games," *International Journal of Game Theory*, vol. 38, no. 1, pp. 3–16, 2009.

[10] Y. Zick, M. Polukarov, and N. R. Jennings, "Taxation and stability in cooperative games," in *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, 2013, pp. 523–530.

[11] C. G. Bird, "On cost allocation for a spanning tree: a game theoretic approach," *Networks*, vol. 6, no. 4, pp. 335–350, 1976.

[12] D. Granot and G. Huberman, "Minimum cost spanning tree games," *Mathematical programming*, vol. 21, no. 1, pp. 1–18, 1981.

[13] D. Granot and G. Huberman, "The relationship between convex games and minimum cost spanning tree games: A case for permutationally convex games," *SIAM Journal on Algebraic Discrete Methods*, vol. 3, no. 3, pp. 288–292, 1982.

[14] D. Granot and G. Huberman, "On the core and nucleolus of minimum cost spanning tree games," *Mathematical programming*, vol. 29, no. 3, pp. 323–347, 1984.

[15] U. Faigle, W. Kern, S. P. Fekete, and W. Hochstättler, "On the complexity of testing membership in the core of min-cost spanning tree games," *International Journal of Game Theory*, vol. 26, no. 3, pp. 361–366, 1997.

[16] U. Faigle, W. Kern, and D. Paulusma, "Note on the computational complexity of least core concepts for min-cost spanning tree games," *Mathematical Methods of Operations Research*, vol. 52, pp. 23–38, 2000.

[17] R. Meir, J. S. Rosenschein, and E. Malizia, "Subsidies, stability, and restricted cooperation in coalitional games," in *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011, pp. 301–306.

[18] Y. Bachrach, E. Elkind, E. Malizia, R. Meir, D. Pasechnik, J. S. Rosenschein, J. Rothe, and M. Zuckerman, "Bounds on the cost of stabilizing a cooperative game," *Journal of Artificial Intelligence Research*, vol. 63, pp. 987–1023, 2018.

[19] R. Meir, Y. Zick, E. Elkind, and J. Rosenschein, "Bounding the cost of stability in games over interaction networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2013, pp. 690–696.

[20] N. Bousquet, Z. Li, and A. Vetta, "Coalition games on interaction graphs: a horticultural perspective," in *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, 2015, pp. 95–112.

[21] E. Resnick, Y. Bachrach, R. Meir, and J. S. Rosenschein, "The cost of stability in network flow games," in *International Symposium on Mathematical Foundations of Computer Science*. Springer, 2009, pp. 636–650.

[22] H. Aziz, F. Brandt, and P. Harrenstein, "Monotone cooperative games and their threshold versions." in *AAMAS*, vol. 10. Citeseer, 2010, pp. 1017–1024.

[23] G. Chalkiadakis, G. Greco, and E. Markakis, "Characteristic function games with restricted agent interactions: Core-stability and coalition structures," *Artificial Intelligence*, vol. 232, pp. 76–113, 2016.

[24] U. Faigle, S. P. Fekete, W. Hochstättler, and W. Kern, "On approximately fair cost allocation in euclidean tsp games," *Operations-Research-Spektrum*, vol. 20, no. 1, pp. 29–37, 1998.

[25] M. Bläser and L. Shankar Ram, "Approximately fair cost allocation in metric traveling salesman games," *Theory of Computing Systems*, vol. 43, no. 1, pp. 19–37, 2008.

[26] A. Caprara and A. N. Letchford, "New techniques for cost sharing in combinatorial optimization games," *Mathematical programming*, vol. 124, no. 1, pp. 93–118, 2010.

[27] L. Liu, X. Qi, and Z. Xu, "Computing near-optimal stable cost allocations for cooperative games by lagrangian relaxation," *INFORMS Journal on Computing*, vol. 28, no. 4, pp. 687–702, 2016.

[28] X. Deng, "Combinatorial optimization and coalition games," in *Handbook of Combinatorial Optimization*, D.-Z. Du and P. Pardalos, Eds. Dordrecht: Kluwer Academic Publishers, 1959, vol. 2, pp. 77–103.

[29] A. S. Schulz and N. A. Uhan, "Sharing supermodular costs," *Operations Research*, vol. 58, no. 4, pp. 1051–1056, 2010.

[30] M. Grötschel, L. Lovász, and A. Schrijver, "The ellipsoid method and its consequences in combinatorial optimization," *Combinatorica*, vol. 1, no. 2, pp. 169–197, 1981.

[31] R. M. Karp and C. H. Papadimitriou, "On linear characterizations of combinatorial optimization problems," in *Foundations of Computer Science, 1980., 21st Annual Symposium on*. IEEE, 1980, pp. 1–9.

[32] M. W. Padberg and M. R. Rao, "The russian method for linear inequalities iii: Bounded integer programming," Ph.D. dissertation, INRIA, 1981.

[33] J. Drechsel and A. Kimms, "The subcoalition-perfect core of cooperative games," *Annals of Operations Research*, vol. 181, no. 1, pp. 591–601, 2010.

[34] J. Edmonds, “Submodular functions, matroids, and certain polyhedra,” in *Combinatorial Structures and Their Applications*, R. Guy, Ed. New York: Gordon and Breach, 1970, pp. 69–87.

[35] X. Deng and C. H. Papadimitriou, “On the complexity of cooperative solution concepts,” *Mathematics of operations research*, vol. 19, no. 2, pp. 257–266, 1994.

[36] W. Kern and D. Paulusma, “Matching games: The least core and the nucleolus,” *Mathematics of Operations Research*, vol. 28, no. 2, pp. 294–308, 2003.

[37] M. R. Garey and D. S. Johnson, “Computers and intractability,” *A Guide to the*, 1979.

[38] R. C. Prim, “Shortest connection networks and some generalizations,” *The Bell System Technical Journal*, vol. 36, no. 6, pp. 1389–1401, 1957.

[39] A. Claus and D. Kleitman, “Cost-allocation for a spanning tree,” *Networks*, vol. 3, pp. 289–304, 1973.

[40] R. Zou, B. Lin, M. Uetz, and M. Walter, “Algorithmic solutions for maximizing shareable costs,” Submitted, 2022.