Algorithmic Cost Allocation Games: Theory and Applications

vorgelegt von

Dipl.-Math. Nam Dũng Hoàng

aus Hanoi, Vietnam

Von der Fakultät II – Mathematik und Naturwissenschaften
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften
– Dr. rer. nat. –

genehmigte Dissertation

Promotionsausschuss

Vorsitzender:	Prof. Dr. Fredi Tröltzsch
Berichter:	Prof. Dr. Dr. h.c. mult. Martin Grötschel
	Dr. habil. Ralf Borndörfer

Tag der wissenschaftlichen Aussprache: 6. Oktober 2010

Berlin 2010
D 83

For my parents

Abstract

Due to economy of scale, it is suggested that individual users, in order to save costs, should join a cooperation rather than acting on their own. However, a challenge for individuals when cooperating with others is that every member of the cooperation has to agree on how to allocate the common costs among members, otherwise the cooperation cannot be realised. Taken this issue into account, we set the objective of our thesis in investigating the issue of fair allocations of common costs among users in a cooperation. This thesis combines cooperative game theory and state-of-the-art algorithms from linear and integer programming in order to define fair cost allocations and calculate them numerically for large real-world applications. Our approaches outclasse traditional cost allocation methods in terms of fairness and users' satisfaction.

Cooperative game theory analyzes the possible grouping of individuals to form their coalitions. It provides mathematical tools to understand fair prices in the sense that a fair price prevents the collapse of the grand coalition and increases the stability of the cooperation. The current definition of cost allocation game does not allow us to restrict the set of possible coalitions of players and to set conditions on the output prices, which often occur in real-world applications. Our generalization bring the cost allocation game model a step closer to practice. Based on our definition, we present and discuss in the thesis several mathematical concepts, which model fairness.

This thesis also considers the question of whether there exists a "best" cost allocation, which people naturally like to have. It is well-known that multicriteria optimization problems often do not have "the optimal solution" that simultaneously optimizes each objective to its optimal value. There is also no "perfect" voting-system which can satisfy all the five simple, essential social choice procedures presented in the book "Mathematics and Politics. Strategy, Voting, Power and Proof" of Taylor et al. Similarly, the cost allocation problem is shown to experience the same problem. In

particular, there is no cost allocation which can satisfy all of our desired properties, which are coherent and seem to be reasonable or even indispensable. Our game theoretical concepts try to minimize the degree of axiomatic violation while the validity of some most important properties is kept.

From the complexity point of view, it is NP-hard to calculate the allocations which are based on the considered game theoretical concepts. The hardest challenge is that we must take into account the exponential number of the possible coalitions. However, this difficulty can be overcome by using constraint generation approaches. Several primal and dual heuristics are constructed in order to decrease the solving time of the separation problem. Based on these techniques, we are able to solve our applications, whose sizes vary from small with 4 players, to medium with 18 players, and even large with 85 players and $2^{85} - 1$ possible coalitions. Via computational results, we show the unfairness of traditional cost allocations. For example, for the ticket pricing problem of the Dutch IC railway network, the current distance tariff results in a situation where the passengers in the central Randstad region of the country pay over 25% more than the costs they incur and these excess payments subsidize operations elsewhere, which is absolutely not fair. In contrast, our game theory based prices decrease this unfairness and increase the incentive to stay in the grand coalition for players.

Acknowledgments

I would like to express my sincere gratitude to my advisor Prof. Dr. Dr. h.c. mult. Martin Grötschel for the interesting research theme and for his supervision. I am grateful to Dr. Ralf Borndörfer for his valuable supports and suggestions.

I would like to express my thank to the Zuse Institute Berlin (ZIB) for providing me a Konrad-Zuse Scholarship. I also want to thank to all my friends and colleagues at ZIB for the wonderful working atmosphere.

Moreover, I am appreciative to my proof readers Carlos Cardonha, Dr. Benjamin Hiller, and Dr. Thorsten Koch for their precious comments.

And last but not least, I would like to thank my parents for their care and continual supports.

Contents

Chapter 1

Introduction

There has been an endless controversy on a fair price system for train slots in Germany for years. The current one of the German railway infrastructure provider DB Netz AG is accused of being incommensurate with the real cost. The train path charges per train path kilometer are composed of route category, train path product, service-dependent component, regional factor, and several other components. DB Netz AG has subdivided its routes in 12 categories based on the maximal allowed speed, which reflects the investment cost. The higher the maximal allowed speed of a route, the more expensive is the basic price for one transport kilometer on this route. There are 9 different train path products with 5 products for passenger transport and 4 products for freight transport. Each of them is assigned to a factor from 0.5 to 1.8. This classification is based on the customer's demands, e.g., direct connection, priority in terms of operations management, and/or frequency. DB Netz AG claims that the role of the service-dependent component is to provide "an incentive to reduce disturbances and improve the efficiency of the rail network" by applying some penalty factors on "very busy routes" and for trains where a minimum speed of 50 km/h is not achieved. As revealed from its name, regional factor differs locally depending on the regional network concerned. They represent a supplement on the top of the train path price. The train path price is then the product of these components. However, it is unclear how DB Netz AG came to these numbers. In the article "Schienennetz wird für Deutsche Bahn zum blendenden Geschäft" from 18.07.2009 in Wirtschaftswoche, a German weekly business news magazine, Christian Schlesiger wrote that:

> The path prices are often incommensurate to the costs caused by the trains. Example Hamburg-Berlin: On this route local

> and long distance trains pay the same amount of 6.95 Euros per kilometer, although the load on the rails and railroad switches is different. The ICE requires the triple of the capacity of the regional trains, needs numerous expensive extra facilities, a special trolley, and high-speed switches. According to a calculation of KCW, a management consulting firm in public transport, the charges of ICE ought to be 10.43 Euros per kilometer, while a regional train would pay only 3.48 Euros. For Deutsche Bahn that would be very unattractive, since it would have to transport 37 passengers more per ICE in order to obtain the same profit.

The "Interessengemeinschaft" Bibertbahn, which has striven for years for the reactivation of the railway line Nuremberg-Stein-Altenberg-Zirndorf-Leichendorf, also complained on the high path price for the 5.6 kilometer route between Stein and the Nuremberg main station. DB Netz AG asked for about one million euros per year, while the community claimed that according to its estimation 200 thousands euros should be a reasonable price since the extra cost for DB Netz AG is very small. These are only two of many complaints on the DB Netz AG price system. The lack of transparency and equity in the price calculation leads to this eternal strife. In order to solve it, we have to answer the question: What is a fair price? Finding fair prices for every party is the goal of a cost allocation problem.

This thesis deals with the cost allocation problem, i.e., the question of fair allocations of common costs among users. It combines concepts of cooperative game theory and state-of-the-art algorithms from linear and integer programming in order to solve large real world applications. Cost allocation problems (see [65] for a survey/an introduction) are widespread. They appear whenever it is necessary or desirable to divide a common cost between several users or items. Some parts of the cost are direct costs, which are easy to allocate. However, it is not the case for the remainder. And then a fair allocation method is needed. A cost allocation problem arises whenever a cooperation exists. This cooperation can be forced or voluntary because of financial benefit. In many cases, because of economy of scale, it is cheaper for an individual user to join in a larger cooperation than acting itself. The goal of a cost allocation problem is to find a "fair" price for every user. Unfortunately, fairness has a vague meaning and may be understood subjectively among people. However, we want to present and discuss in this thesis several mathematical concepts, which model fairness in some way. The analysis bases on the fact that several groups of users can form real coalitions or theoretical coalitions. The cost of each coalition is the minimal cost to fulfil the demand of every user in this coalition. If real coalitions are allowed, then the users have an alternative to accepting a given price, namely forming more favourable coalitions. Otherwise, each given price for which there is a better theoretical alternative for some users, will cause dissatisfaction and unrest among these users. A fair price can be considered as a price which is accepted by every user or at least the best possible solution in some sense.

History of cost allocation

The history of the cost allocation problem goes back to the early 20th century. One of the first examples that we have found in the literature is the cost allocation problem for the Tennessee Valley Authority [50, 52]. The Tennessee Valley Authority was a major regional development project created by an act of the congress in the 1930's to stimulate economic activity in the mid-southern United States. The goal was to construct a series of dams and reservoirs along the Tennessee River in order to generate hydroelectric power, control flooding, and improve navigational and recreational uses of the waterway. Economists who analyzed the costs and benefits of this project observed that there is no completely obvious way to attribute costs to these purposes, because the system is designed to satisfy all of them simultaneously. The concepts that they devised to deal with this problem foreshadowed modern ideas in game theory [65]. This problem was reconsidered later on as a cooperative game in [61, 65].

Game theory experienced a flurry of activity in the 1950's and 1960's, during which time the concepts of the core [23, 24], the Shapley value [54], and the nucleolus [53, 45] were developed. These concepts have been then applied to the cost allocation problem. One of the first applications of game theory to cost allocation is presented in Martin Shubik's paper [57]. In this paper, Shubik argued that the Shapley value could be used to provide a mean of devising incentive-compatible cost assignments and internal pricing in a firm with decentralised decision making. Since then, several applications of the cost allocation problem using cooperative game theory as a mathematical tool for analysis have been published. The allocation of joint overhead costs of a firm among its different divisions is studied in [31]. Four cost allocation schemes were evaluated using the core criteria in this paper. Littlechild and Thompson [42, 44] considered the problem of aircraft landing fees. They use the nucleolus, along with the core and

Shapley value, to calculate fair and efficient landing and take-off fees for Birmingham airport during the time 1968-1969. There were 11 different aircraft types at that time. The total cost are the sum of annual runway operating costs, which depend on the number of movements of all aircraft types, and annual runway capital costs, which depend upon the largest type for which the runway is designed. Young et al. [67] presented several cost allocation methods and applied them to a municipal cost allocation problem in Sweden. Their case study consists of 18 municipalities in the Skåne region of southern Sweden. At that time most of the municipal water supply was drawn from three sources: local groundwater and water from two lakes, Vombsjön and Ringsjön. Water from the lakes was distributed to the municipalities via two pipeline systems. The question was how to allocate fairly the common cost. Engevall et al. [13, 14, 15] studied the cost allocation problem arisen in a distribution planning situation at the logistics department at Norsk Hydro Olje AB, a sales and marketing company for gas and gas-oil in Sweden. The logistics department is responsible for the transportation of different qualities of gas and gas-oil to the customers. The distribution planning problem was modeled as a vehicle routing problem and the transportation cost was defined as the corresponding optimal value. Based on that they analyzed how to allocate the transportation cost among Norsk Hydro's customers.

Though one can name many more examples of the cost allocation problem than the ones listed above and in our bibliography, the number of real applications in the mathematical literature is small. One reason may be that price determination is a complicated task, which is influenced by political and/or psychological factors. Prices often result from negotiation or monopoly decision, which are based on other facts than fairness. However, fairness is immensely important. An unfair price will cause dissatisfaction and instability. In our opinion, the necessity of equitable cost allocation methods will increase. The regulation policy of the European Union in telecommunication, transportation, and energy requires equitable tariffs. Even if cost allocation methods are not used directly, the prices obtained thereby can be used as basis for price determination or negotiation and the mathematical concept of fairness still can be used to justify some given tariff. Another reason for the small number of published applications may be the size of the problem itself. Many cost allocation problems consist of just a few users and have a simple structure, which can be solved easily without involved analysis. The users can agree on some simple pricing scheme in that case. On the other hand, for problems with many users, it is not easy to calculate the price since we have to take into account the exponential number of possible groupings of users. All the problems listed above are small except for the one studied by Engevall et al., which has a medium size of 21 users and about two millions of possible coalitions. In [67], the authors assort the 18 municipalities into six groups and solve the problem of six users since in their opinion "to develop the costs for each of the $2^{18} - 1$ possible groupings of the 18 municipalities would be impractical and unrealistic". This was the situation in the 1980's. However, it is not necessary to evaluate the cost of every possible coalition. Using the constraint generation technique [30, 15, 11], we only have to calculate the costs of several coalitions and therefore are able to solve large applications of the cost allocation problem in practice.

Thesis outline

The thesis is organized into two parts. The first part, which contains Chapters 2, 3, 4, and 5, deals with the theory of the cost allocation problem, while the second part is devoted to applications. In Chapter 2, we represent the cost allocation problem as a cost allocation game. The definition of the cost allocation game in this thesis is a generalization of the common one, which allows us to restrict the set of possible coalitions of players and to set conditions on the output prices. This generalization is practically relevant and gives us not only the possibility to merge several game theoretical concepts in a single one, but also a better understanding of the relationship between different concepts. Several desired properties of the output of the cost allocation game are considered. They are coherent and seem to be reasonable or even indispensable. Unfortunately, against all hope in the existence of "the best" allocation, there is no allocation method which fulfils all these properties in general. It is the task of the decision maker to choose the most important ones and sacrifice the others. We also investigate in this chapter several game theoretical concepts like the Shapley value, the core, the least core, the nucleolus, and their generalizations. These concepts only focus on the possible grouping of each player with the other ones. Their outputs reflect the strategic position of each player and are fair in that sense. The major drawback of these concepts is that they do not take into account the realizability of their outputs. The prices obtained by these concepts are coalitionally fair, but may favour several players extremely over some others. The new concept, called (f, r)-least core, with some given weight function f and reference vector r, is a compromise between the coalitional fairness and the individual satisfaction. In the end of this chapter, we present some theoretical results on the non-emptiness of the core and the so called f-least core radius.

Chapter 3 analyses the computational complexity of cost allocations which are based on the game theoretical concepts for cost allocation games whose cost functions are given by minimization problems. In general, these problems are NP-hard, even for simple games like the so called minimum cost spanning tree game, whose cost function is polynomial time evaluable. However, we can prove that there exist oracle-polynomial-time algorithms based on the ellipsoid method for submodular cost allocation games.

Unfortunately, a real world application often does not have a submodular cost function. This means that the polynomial-time solvability is not guaranteed. Apart from that, the ellipsoid method sufferes from numerical instability and poor performance in practice. The goal of Chapter 4 is to study state-of-the-art algorithms that work well for large real-world applications. The hardest challenge is that we must take into account the exponential number of the possible coalitions. To overcome this difficulty, several constraint generation approaches and their separation problems as mixed integer programs are presented. Computational results show that these approaches work very well for our applications. We also discuss the choice of a good starting set for the constraint generation approaches and several primal and dual heuristics for the separation problem, which are very effective in practice.

Chapter 5 shows how one can evaluate the fairness of a given price vector visually. For games with many players, it is impossible to calculate the cost and the profit (the satisfaction) of every coalition with a given price vector. However, via a so called fairness distribution diagram, we can show how the curve representing the profits of some essential coalitions looks like and are able to compare different price vectors.

In Chapter 6, we demonstrate how different are the results when we apply different cost allocation methods to the same cost allocation problem. For this we consider a very simple real example with only four players.

We study in Chapter 7 the problem in which a good is produced in some places and then transported to customers via a network. Example of this problem are water supply system, irrigation system, or gas transportation. The cost can be modeled by a non-linear multi-commodity flow problem. However, in order to solve large cost allocation problems, it is linearized using piecewise linear function as in [10]. Based on this model, we solve a cost allocation problem in water resources development in Sweden.

The subject of Chapter 8 is the ticket pricing problem in public transport, where passengers share a common infrastructure. The question of ticket fee ist not new, but it has been considered from another perspective. Thereby, the decision maker either is only interested in maximizing the revenue or just uses the common distance price. The distance price of each passenger depends only on his traveling distance and the used means of transport. These approaches do not take fairness into account. Consequently, by using prices based on these approaches, there exist coalitions which have to pay much more than their costs. We will show that for the example of the Dutch IC railway network the current distance tariff results in a situation where the passengers in the central Randstad region of the country pay over 25% more than the costs they incur and these excess payments subsidize operations elsewhere, which is absolutly not fair. Our approach models the ticket pricing problem as a cost allocation game, where the set of all passengers traveling between each pair of an origin and a destination is considered as a player, and calculates fair prices based on the (f, r)-least core.

Chapter 2

The Cost Allocation Problem

In economics, cost allocation means "an assignment of common costs to cost objects (jobs or tasks)". These assignments usually result from simple forms or rules or just negotiation. However, the prices obtained this way often do not reflect the real cost caused by each user (cost object) and therefore raise discontent among them, like our example of the path price system of the German railway infrastructure provider DB Netz AG in the introduction. In order to avoid such kind of conflict, an equitable cost allocation scheme is needed. This is exactly the goal of the cost allocation problem. The *cost allocation problem* is the problem of allocating fairly the common costs among the users. Whenever it is necessary or desirable to divide a common cost between several users or items, a cost allocation method is needed. Let us start with an illustrative example. There are three cities A, B, and C. The distances in kilometers between these cities are

$$AB = 800,\ BC = 100,\ \text{and } AC = 850.$$

Person 1 wants to travel from A to C while person 2 want to go from B to C. The cheapest solution for person 1 to travel from A to C is with a car at the cost of 90 euros. For person 2 traveling with the train from B to C at the cost of 20 euros is the cheapest option. However, if the two persons cooperate, i.e., instead of driving directly from A to C, person 1 firstly drives to B, picks up person 2 there, and then goes to C, then it will cost them only 100 euros. In other words, this alternative will give them a benefit of 10 euros. There are several ways to allocate the cost:

- Allocating the common cost proportionally to the driving distances: The total driving distance of the two persons is $800 + 2 \cdot 100 = 1000$

kilometers. The price for each driving kilometer is then 10 cents. This means that the prices for person 1 and person 2 are 90 and 10 euros, respectively.

- Allocating the benefit equally: Person 1 pays 85 euros and person 2 pays 15 euros.

- Allocating the common cost proportionally to the individual cost: Person 1 pays $100\frac{90}{90+20} \approx 81.82$ euros and person 2 pays $100\frac{20}{90+20} \approx 18.18$ euros.

The question is, which allocation is the most suitable solution? The first one is a common solution. It is considered to treat the people equally since each person has to pay the same amount of money for one unit (one kilometer). However, this solution gives no incentive to person 1, since the price that he has to pay is the same as the individual cost, while person 2 gets the whole benefit. The cooperation will be hardly realized with this price. With the second and third allocations, each player has some profits, and therefore cooperation is preferred. But these two allocations differ from each other by the amount of the profits. With the second one, each person has a benefit of 5 euros, i.e., person 1 saves only 5.56 percents, while person 2 has to pay 25 percents less. It may make person 1 discontented, since he would think that person 2 pays too few. By the third allocation, this will not happen because each person has the same relative profit, namely, 9.1 percents. The third allocation is perhaps the best solution for our example. However, such simple analysis cannot be applied to more complicated real world applications, though the main idea is still useful. A general mathematical tool is then needed to determine fair allocations or justify some given prices.

The first goal of this chapter is to present and study some desired properties of the cost allocation. The second goal is to consider several concepts of fairness based on cooperative game theory. Cooperative game theory analyzes optimal strategies for groups of individuals, presuming that they can enforce agreements between them about proper strategies. The allocation obtained in this way reflects the position and the real cost of each grouping of individuals including each individual itself. The last part of this chapter is devoted to a theoretical study, which helps to improve the computational task.

2.1 The Cost Allocation Game

The *cost allocation game* deals with price determination and can be defined as follows. Given is a finite set of players $N = \{1, 2, \ldots, n\}$. These players can form real coalitions or theoretical coalitions. The set of possible coalitions is given by a family $\Sigma \subseteq 2^N$. Assume that the family Σ contains the *grand coalition* N and the set $\Sigma \backslash \{\emptyset, N\}$ is non-empty. In addition, given are also a cost function $c : \Sigma \to \mathbb{R}$ satisfying

$$c(S) > 0, \ \forall S \in \Sigma \backslash \{\emptyset\},$$

and a polyhedron P,

$$P = \{x \in \mathbb{R}^N_+ \mid Ax \leq b\},$$

which gives conditions on the prices x that the players are asked to pay. We denote $\Sigma^+ := \Sigma \backslash \{\emptyset\}$. The tuple $\Gamma := (N, c, P, \Sigma)$ is called *data* of a cost allocation game. The corresponding *cost allocation game* is specified by the requirement that the output of this game is a "fair" allocation of the common cost $c(N)$ among the players in N. We use the notation (N, c, P, Σ) for both the cost allocation game as well as its data. Each cost allocation problem can be modeled as a cost allocation game, where each player may be an individual user or a group of users. Each price vector in P is called a *valid price*. The cost allocation problem only makes sense if there exists a valid price that covers exactly the cost. Therefore, we require that the following set

$$\mathcal{X}(\Gamma) := \Big\{x \in P \mid \sum_{i \in N} x_i = c(N)\Big\}$$

is non-empty. Each vector in $\mathcal{X}(\Gamma)$ is called an *imputation* of the game and the set $\mathcal{X}(\Gamma)$ is called the *imputation set* of Γ. Each cost allocation game $(N, c, \mathbb{R}^N_+, 2^N)$ is called a *simple cost allocation game*.

In the common definition of the cost allocation game, there are no other requirements on the prices than the non-negativity and each subset of N is a possible coalition, i.e., $P = \mathbb{R}^N_+$ and $\Sigma = 2^N$. However, both kinds of restrictions on prices and allowed coalitions occur in practice. Therefore, a study of the cost allocation game in the general form above is desirable. Moreover, with our definition of cost allocation game, we can on one hand generalize several concepts from game theory into a single definition and on the other hand describe easily some new concepts.

If we remove the restrictions in the definition of the cost allocation game that the prices have to be non-negative and the cost function has to be positive, then we obtain the so called *general cost allocation game*. Each cost allocation game is a general cost allocation game as well.

For each general cost allocation game $\Gamma := (N, c, P, \Sigma)$ and each possible coalition $S \in \Sigma^+$, we denote

$$\Sigma_S := \{T \in \Sigma \,|\, T \subseteq S\}$$

and $c_S := c|_{\Sigma_S}$. For real applications, there are rules or conditions on prices, which can be formulated mathematically with each given set of players. We assume that for each general cost allocation game (N, c, P, Σ) there exists an oracle that gives us the set P_S of valid prices for players in each given coalition S in Σ^+ when they leave the grand coalition N and form a cooperation themselves. The restriction $\Gamma_S = (S, c_S, P_S, \Sigma_S)$ of Γ to S is called *a subgame of* Γ. This subgame is nothing else than the general cost allocation subproblem among the players in the coalition S.

For a given weight function $f : \Sigma \backslash \{\emptyset\} \to \mathbb{R}_{>0}$, each price vector $x \in \mathbb{R}^N$, and each coalition $S \in \Sigma^+$, we define the *f-excess* of S at x as

$$e_f(S, x) := \frac{c(S) - x(S)}{f(S)}.$$

In order to avoid any possible misunderstanding regarding the notation, we explicitly point out that, for functions f and c, $f(S)$ and $c(S)$ denote the values of these functions for coalition S, while for the price vector x, $x(S)$ denotes the sum $\sum_{i \in S} x_i$, i.e., the price of the coalition S. The f-excess represents the f-weighted gain (or loss, if it is negative) of the coalition S, if its members accept to pay $x(S)$ instead of operating some service themselves at cost $c(S)$. The excess measures price acceptability. The smaller $e_f(S, x)$, the less favorable is the price x for coalition S; and the larger $e_f(S, x)$, the more favorable is the price x. For $e_f(S, x) < 0$, i.e., in case of a loss, x will be seen as unfair by the members of S. If the players are allowed to leave the grand coalition, then for the coalition S it is cheaper to create its own cooperation than accepting the price x. Conversely, if $e_f(S, x) > 0$, then for the players in S accepting the price x is a better alternative than acting themselves.

A *cost allocation method* is a function that maps each general cost allocation game $\Gamma := (N, c, P, \Sigma)$ in its domain to a vector in $\mathbb{R}^N$. We require that the domain of each cost allocation method contains every simple cost allocation game. A cost allocation method is called *well-defined* for a general cost allocation game if this game and all of its subgames belong to the domain of the cost allocation method. For a general cost allocation game $\Gamma := (N, c, P, \Sigma)$, the price vector obtained by applying a given cost allocation method Φ is $\Phi(\Gamma)$. The price for each player i in N is then $\Phi(\Gamma)_i$. The sum $\sum_{i \in N} \Phi(\Gamma)_i$ is the total price that players are asked to pay.

2.2 Desired Properties and Conflicts

In this section, we present and study some desired properties of cost allocation methods. Each of them seems to be indispensable, but unfortunately they cannot be fulfilled simultaneously. Depending on the problem, one has to decide which one is unalterable and which one should be approximated as well as possible. But before doing that, we want to list several cost allocation methods, which will be used later as examples for verifying the properties.

Non-cooperative cost allocation method: Each player is required to pay the same amount as its individual cost. This means that the price of each player i in N is

$$\Phi^1(N, c, P, \Sigma)_i := c(\{i\}).$$

Proportional cost allocation method: The price of each player is proportional to its individual cost and the total price of all players is equal to the common cost. That means the price of each player i in N is

$$\Phi^2(N, c, P, \Sigma)_i := \frac{c(\{i\})}{\sum_{j \in N} c(\{j\})} c(N).$$

Minimum subsidy cost allocation method: The prices of players are allocated in such a way that the price of each coalition (including the grand coalition) is not larger than its cost by doing the job itself and the total price is as large as possible. This means that the minimum subsidy cost allocation method applied on a general cost allocation game $\Gamma = (N, c, P, \Sigma)$ gives us an optimal solution of the following linear program

$$\begin{aligned} &\max \ x(N) \\ s.t.\ &x(S) \leq c(S), \ \forall S \in \Sigma^+ \\ &x \in P \end{aligned}$$

provided that the linear program is feasible.

Proportional least core cost allocation method: Least core is a concept of game theory, which will be defined in the next section. A least core price is a price such that the common cost is exactly covered and the maximum ratio $\frac{x(S)}{c(S)}$ over every possible coalition S is kept as small as possible. The idea behind this allocation method is that, for each coalition S the smaller the ratio $\frac{x(S)}{c(S)}$, the more favorable is the price x for this coalition. Mathematically, this allocation method, denoted by Φ^4, is defined as follows. For each general cost allocation game $\Gamma = (N, c, P, \Sigma)$ which satisfies that the following linear program is feasible and bounded

$$\begin{aligned} &\max_{(x,\varepsilon)} \ \varepsilon \\ s.t.\ &x(S) \leq (1-\varepsilon)c(S),\ \forall S \in \Sigma\backslash\{\emptyset, N\} \\ &x(N) = c(N) \\ &x \in P, \end{aligned}$$

let ε^* be the optimal value, then $\left(\Phi^4(\Gamma), \varepsilon^*\right)$ is an optimal solution of the above linear program.

2.2.1 Desired Properties

In order to avoid repeating properties' descriptions, we use the following definition. A cost allocation method is said to have a property X if it holds for every general cost allocation game in its domain.

Validity

The most important property is validity. A cost allocation is useless if it does not fulfil the conditions of the game. A price vector x is called *valid* for a general cost allocation game $\Gamma = (N, c, P, \Sigma)$ if x belongs to P. A cost allocation method Φ is called *valid* for a general cost allocation game Γ in the domain of Φ if $\Phi(\Gamma)$ is valid for Γ. For an arbitrary general cost allocation game $\Gamma = (N, c, P, \Sigma)$, we have that

- The non-cooperative cost allocation method and the proportional cost allocation method are valid for Γ if either $P = \mathbb{R}^N$ or $P = \mathbb{R}^N_+$ and the cost of each individual and the common cost $c(N)$ are positive.

- The minimum subsidy cost allocation method and the proportional least core cost allocation method are valid.

Efficiency

In the case that there is no subsidy from outside, the cooperation can only be realized if the common cost is covered. On the other hand, it is irrational to pay more than the common cost. Therefore, the total price should equal the common cost. A price vector x is called *efficient* for a general cost allocation game $\Gamma = (N, c, P, \Sigma)$ if the total price $x(N)$ equals the common cost $c(N)$. A cost allocation method Φ is called *efficient* for a general cost allocation game Γ in the domain of Φ if $\Phi(\Gamma)$ is efficient for this game.

- The proportional cost allocation method is efficient.
- The proportional least core cost allocation method is efficient.

Coalitional stability

We consider a cost allocation method Φ and a general cost allocation game $\Gamma = (N, c, P, \Sigma)$ which satisfy that Φ is well-defined for Γ, i.e., Γ and all its subgames belong to the domain of Φ. For each coalition S in Σ^+, we denote the corresponding subgame by Γ_S. Joining N is more favourable for S than acting itself only if its price does not increase thereby, i.e.,

$$\sum_{i \in S} \Phi(\Gamma)_i \leq \sum_{i \in S} \Phi(\Gamma_S)_i. \tag{2.1}$$

This will guarantee the stability of the grand cooperation. A cost allocation method Φ is called *coalitionally stable* for a general cost allocation game $\Gamma = (N, c, P, \Sigma)$, for which Φ is well-defined, if (2.1) holds for each coalition S in Σ^+. A cost allocation method Φ is called *coalitionally stable* if it is coalitionally stable for every general cost allocation game, for which Φ is well-defined.

- The non-cooperative cost allocation method is coalitionally stable.
- The minimum subsidy cost allocation method is coalitionally stable for every general cost allocation game $\Gamma = (N, c, P, \Sigma)$ in its domain which satisfies that

  $$x|_S \in P_S, \ \forall x \in P, \ \forall S \in \Sigma^+, \tag{2.2}$$

  where P_S is the set of valid prices of the subgame Γ_S. For example, each simple cost allocation game fulfills (2.2).

User friendliness

A price vector x is called *user friendly* for a general cost allocation game $\Gamma = (N, c, P, \Sigma)$ in the domain of Φ if the price of each coalition in Σ^+ does not exceed its cost, i.e.,

$$x(S) \leq c(S),\ \forall S \in \Sigma^+.$$

A cost allocation method Φ is called *user friendly* for a general cost allocation game Γ in the domain of Φ if $\Phi(\Gamma)$ is user friendly for Γ.

- The minimum subsidy cost allocation method is user friendly.

Core cost allocation

The core is a well-known concept from game theory. The core of a general cost allocation game is the set of all valid, user friendly, and efficient prices. We will see in the next section that the core may be empty. However, if it is non-empty, the allocated price should belong to it. A cost allocation method Φ is called a *core cost allocation* method if for every general cost allocation game Γ in the domain of Φ whose core is non-empty, the price $\Phi(\Gamma)$ belongs to the core of Γ.

- The minimum subsidy cost allocation method is a core cost allocation method.

- The proportional least core cost allocation method is a core cost allocation method.

Monotonicity

A desired property of the allocation method is that, if an arbitrary player agrees to join some subcoalition of the grand coalition then his price does not decrease compared to his allocated price in the grand coalition. If this property holds, then there is no incentive for any player to leave the grand coalition. A cost allocation method Φ is called *monotonic* for a general cost allocation game $\Gamma = (N, c, P, \Sigma)$ in the domain of Φ if for every coalition S in Σ^+ and its corresponding subgame Γ_S there holds

$$\Phi(\Gamma)_i \leq \Phi(\Gamma_S)_i,\ \forall i \in S.$$

A monotonic cost allocation method is clearly coalitionally stable. Another property of a monotonic cost allocation method is that, if a new player joins a cooperation such that the original game is a subgame of the new one and the cost allocation method is well-defined for the new game, then the price for each present player does not increase.

- The non-cooperative cost allocation method is monotonic.

Equality

Assume that there is some measure to evaluate the resource utilization of each player based on certain unit. E.g., for the ticket pricing game in public transport we can consider the resource utilization of each passenger as its traveling distance and a unit is a traveling kilometer. A cost allocation is called *equal* for the players if each player has to pay the same amount of money for a unit. This allocation is widely used and is often claimed to be "fair".

Bounded variation

The cost function of real applications often can only be approximated and/or can change over time. For example, transportation cost depends on energy prices, which vary over time. Therefore, cost allocation method should be insensitive to small changes of the cost function. A cost allocation method Φ is said to have *bounded variation* for a general cost allocation game $\Gamma := (N, c, P, \Sigma)$ if there exist $\lambda > 0$ and $K > 0$ such that for every number $\alpha \in [0, \lambda]$ and general cost allocation game $\tilde{\Gamma} := (N, \tilde{c}, P, \Sigma)$ satisfying

$$|\tilde{c}(S) - c(S)| \leq \alpha |c(S)|, \ \forall S \in \Sigma^+, \tag{2.3}$$

there holds

$$|\Phi(\tilde{\Gamma})_i - \Phi(\Gamma)_i| \leq K\alpha|\Phi(\Gamma)_i|, \ \forall i \in N : \Phi(\Gamma)_i \neq 0. \tag{2.4}$$

In the above definition, K may vary depending on Γ. In the following one, we require that K is independent on the general cost allocation games. Φ is said to have *uniformly bounded variation* if there exist $\lambda > 0$ and $K > 0$ such that for every number $\alpha \in [0, \lambda]$ and general cost allocation games $\Gamma := (N, c, P, \Sigma)$ and $\tilde{\Gamma} := (N, \tilde{c}, P, \Sigma)$ satisfying (2.3) there holds (2.4).

- The non-cooperative cost allocation method has uniformly bounded variation.

- The proportional cost allocation method has uniformly bounded variation considering only general cost allocation games with positive cost functions.

The following table summarizes the properties of the four considered cost allocation methods, namely, non-cooperative (Non-coop), proportional (Prop), minimum subsidy (MinSub), and proportional least core (Prop LC) cost allocation methods:

	Non-coop	Prop	MinSub	Prop LC
Validity	No*	No*	Yes	Yes
Efficiency		Yes		Yes
Coalitional stability	Yes		No*	
User friendliness			Yes	
Core cost allocation			Yes	Yes
Monotonicity	Yes			
Bounded variation	Yes	Yes*		

Table 2.1: Properties of cost allocation methods

Here, Yes* means that the property holds for cost allocation games, while No* means that the property fails in general but holds for some large classes of general cost allocation games that appear in practice.

2.2.2 Conflicts

The properties presented in the previous subsection are reasonable or even indispensable. The question is whether there exists a cost allocation method which has all these properties. As you could guess, the answer is no. For each general cost allocation game whose core is non-empty, each efficient core cost allocation method is also coalitionally stable and user friendly. And clearly, a valid, efficient cost allocation method which is coalitionally stable (or user friendly) for every cost allocation game in its domain having a non-empty core is a core cost allocation method. However, neither equality nor uniformly bounded variation can hold simultaneously with the core cost allocation property as we will see in the following. We also will show that, in the general case, if the efficiency and core cost allocation property are essential, then we have to sacrifice the other properties except validity.

It is well-known that the core of a simple cost allocation game may be empty. In other words, the following result holds.

Proposition 2.2.1. *There is no efficient, coalitionally stable cost allocation method, even if we only consider cost allocation games which have monotone, subadditive cost functions.*

Proof. Let $N = \{1, 2, 3\}$ and the cost function c be defined as follows

$$c(S) = \begin{cases} |S|, & |S| \leq 1, \\ |S| - 1, & |S| \geq 2. \end{cases}$$

Clearly, c is monotone and subadditive. Denote $S_i := N \backslash \{i\}$ for $i = 1, 2, 3$. For each efficient allocation method Φ, there holds

$$\sum_{i=1}^{3} \Phi(S_i, c_{S_i}, \mathbb{R}^2_+, 2^{S_i})(S_i) = \sum_{i=1}^{3} c(S_i) < 2c(N) = \sum_{i=1}^{3} \Phi(N, c, \mathbb{R}^3_+, 2^N)(S_i),$$

i.e., Φ is not coalitionally stable. □

Proposition 2.2.2. *There is no efficient, user friendly cost allocation method.*

Proof. This proposition is a corollary of Proposition 2.2.1 because each efficient cost allocation method is user friendly if and only if it is coalitionally stable. □

Proposition 2.2.3. *There is no efficient, monotonic cost allocation method.*

Proof. This proposition is a corollary of Proposition 2.2.1 since each monotonic cost allocation method is coalitionally stable. □

Proposition 2.2.4. *Each equal cost allocation method is not a core cost allocation method.*

Proof. We give a cost allocation game with a non-empty core for which the price obtained by applying an arbitrary equal cost allocation does not belongs to its core. If the allocated price vector is not efficient, then it does not belong to the core. Therefore, we only have to consider equal cost allocation methods which are efficient for our cost allocation game. There are three cities A, B, and C. The distances in kilometers between these cities are:

$$AB = 800, BC = 100, \text{ and } AC = 820.$$

Person 1 wants to travel from A to C while person 2 wants to go from B to C. The cheapest solution for person 1 to travel from A to C is with a car at the cost of 88 Euros. For person 2, traveling by train from B to C at the cost of 20 Euros is the cheapest option. However, if instead of driving directly from A to C, person 1 firstly drives to B, picks up person 2 there, and then goes to C, then it will cost them only 100 Euros. In other words, this alternative will give them a benefit of 8 Euros. The core of this game is non-empty. The total traveling distance of the two persons is $800 + 2 \cdot 100 = 1000$ kilometers. By using an equal allocation method that cover exactly the cost of 100 Euros, the price for each traveling kilometer is 10 cents and the prices for person 1 and person 2 are 90 and 10 euros, respectively. This means that the price for person 1 is larger than his individual cost. $\square$

Proposition 2.2.5. *Core cost allocation methods do not have uniformly bounded variation, even if we only consider cost allocation games having monotone, subadditive cost functions.*

Proof. For arbitrary $k > 0$ and $0 \leq \alpha \leq \frac{1}{k}$, let $N = \{1, 2, 3\}$ and the cost functions c and $\tilde{c}$ be defined as follows

S	$c(S)$	$\tilde{c}(S)$
$\{1\}$	2	2
$\{2\}$	$3+k$	$3+k$
$\{3\}$	$4+k$	$4+k$
$\{1,2\}$	$3+k$	$3+(1+\alpha)k$
$\{2,3\}$	$5+2k$	$5+2k$
$\{3,1\}$	$4+k$	$4+(1+\alpha)k$
$\{1,2,3\}$	$6+2k$	$6+(2+\alpha)k$

The cost functions c and $\tilde{c}$ are monotone and subadditive and satisfy

$$0 \leq \tilde{c}(S) - c(S) \leq \alpha c(S), \ \forall \emptyset \neq S \subseteq N.$$

Denote $\Gamma_k := (N, c, \mathbb{R}^3_+, 2^N)$ and $\tilde{\Gamma}_k := (N, \tilde{c}, \mathbb{R}^3_+, 2^N)$. Their cores $\mathcal{C}(\Gamma_k)$ and $\mathcal{C}(\tilde{\Gamma}_k)$ are

$$\mathcal{C}(\Gamma_k) = \{(1, 2+k, 3+k)\}$$
$$\mathcal{C}(\tilde{\Gamma}_k) = \{(1+\alpha k, 2+k, 3+k)\}.$$

Hence, for each core cost allocation method Φ, there holds

$$\Phi(\tilde{\Gamma}_k)_1 - \Phi(\Gamma_k)_1 = \alpha k = \alpha k \Phi(\Gamma_k)_1,$$

i.e., Φ does not have uniformly bounded variation. □

Due to the above Propotitions and Table 2.1, we have the following table, which indicates which properties fail/hold for which cost allocation methods:

	Non-coop	Prop	MinSub	Prop LC
Validity	No*	No*	Yes	Yes
Efficiency	No	Yes	No	Yes
Coalitional stability	Yes	No	No*	No
User friendliness	No	No	Yes	No
Core cost allocation	No	No	Yes	Yes
Monotonicity	Yes	No	No	No
Bounded variation	Yes	Yes*	No	No

Table 2.2: Properties of cost allocation methods

Here, Yes* means that the property holds for cost allocation games, while No* means that the property fails in general but holds for some large classes of general cost allocation games that appear in practice (see Subsection 2.2.1). User friendliness often fails for the non-cooperative cost allocation method because of economy of scale. The minimum subsidy cost allocation method is also not monotonic in general. These statements can be easily proved by constructing counter examples.

2.2.3 Some Other Desired Properties

Symmetry

Given a general cost allocation game $\Gamma = (N, c, P, \Sigma)$, two players i and j in N, are called *equivalent* if there hold

(i) Equivalence regarding Σ

$$\forall S \subseteq N \backslash \{i, j\} : S \cup \{i\} \in \Sigma \iff S \cup \{j\} \in \Sigma,$$

$$\exists S \in \Sigma : S \ni i \vee S \ni j.$$

(ii) Equivalence regarding c

$$c(S \cup \{i\}) = c(S \cup \{j\}),\ \forall S \subseteq N \backslash \{i, j\} : S \cup \{i\} \in \Sigma.$$

(iii) Equivalence regarding P

$$\forall x \in \mathbb{R}^N : x \in P \Longrightarrow x^{ij} \in P,$$

where

$$x_k^{ij} = \begin{cases} x_j & \text{if } k = i \\ x_i & \text{if } k = j \\ x_k & \text{otherwise.} \end{cases}$$

A cost allocation method should then allocate the same amount to i and j. A cost allocation method Φ is called *symmetric for a general cost allocation game* $\Gamma = (N, c, P, \Sigma)$ in the domain of Φ if it allocates the same amount to equivalent players. A cost allocation method Φ is called *symmetric* if it is symmetric for every general cost allocation game in the domain of Φ. Symmetry is an indispensable requirement for a discrimination-free cost allocation method.

- The non-cooperative cost allocation method is symmetric for general cost allocation games whose sets of possible coalitions contain (the coalition of) each single player.
- The proportional cost allocation method is symmetric for general cost allocation games whose sets of possible coalitions contain (the coalition of) each single player.

Additivity

A cost allocation method Φ is called *additive* if for any general cost allocation games $\Gamma = (N, c, P, \Sigma)$, $\Gamma_1 = (N, c_1, P_1, \Sigma)$, and $\Gamma_2 = (N, c_2, P_2, \Sigma)$ in the domain of Φ satisfying

$$c = c_1 + c_2 \quad \text{and} \quad P = P_1 + P_2,$$

there holds

$$\Phi(\Gamma) = \Phi(\Gamma_1) + \Phi(\Gamma_2).$$

With the additivity property, we can allocate different parts of the cost, e.g., operating cost and fixed cost, separately without changing the total allocated cost of each player.

- The non-cooperative cost allocation method is additive.

Scalar multiplicativity

For each general cost allocation game $\Gamma = (N, c, P, \Sigma)$ and each number λ, the *scalar multiplication* of λ and Γ is defined by

$$\lambda\Gamma = (N, \lambda c, \lambda P, \Sigma).$$

A cost allocation method Φ is called *scalar multiplicative* if for any positive real number λ and any general cost allocation game $\Gamma = (N, c, P, \Sigma)$ in the domain of Φ which satisfy that the general cost allocation game $\lambda\Gamma$ belongs also to the domain of Φ, there holds

$$\Phi(\lambda\Gamma) = \lambda\Phi(\Gamma).$$

- The non-cooperative cost allocation method is scalar multiplicative.
- The proportional cost allocation method is scalar multiplicative.

Dummy player

Given a general cost allocation game $\Gamma = (N, c, P, \Sigma)$ and a player $i \in N$ that satisfy $\Sigma \ni \{i\}$,

$$\forall S \subseteq N\backslash\{i\} : S \cup \{i\} \in \Sigma \Leftrightarrow S \in \Sigma,$$

$$c(S \cup \{i\}) = c(S) + c(\{i\}),\ \forall S \in \Sigma, S \not\ni i,$$

and

$$P = \left\{x \in \mathbb{R}^N \,\middle|\, x|_{N\backslash\{i\}} \in Q_i, x_i \in I_i\right\}$$

for some polyhedron Q_i and interval $I_i \ni c(\{i\})$. The player i is called a *dummy player* of Γ. A dummy player with zero cost, i.e., $c(\{i\}) = 0$, is called a *zero player* of Γ. Since a dummy player contributes nothing to the cooperation, it should pay his own cost, namely, $c(\{i\})$.

- The non-cooperative cost allocation method charges each dummy player exactly the same amount as his individual cost.

- The proportional cost allocation method charges zero players nothing.

In general, a cost allocation method may not charge a dummy player exactly his individual cost. However, we may modify a given cost allocation method, while we firstly find all dummy players of the given game, charge each of them the amount equaling his individual cost, and finally apply the cost allocation method to the subgame which contains all the remaining non-dummy players. In the following, we define the above described operator, denoted by $\mathcal{D}$. To do it we need some other definitions. For an arbitrary general cost allocation game $\Gamma = (N, c, P, \Sigma)$, let $d(\Gamma)$ denote the set of all its dummy players. One can easily prove that there exist a polyhedron Q and intervals I_i for $i \in d(\Gamma)$ such that

$$P = \left\{x \in \mathbb{R}^N \,\middle|\, x|_{N \setminus d(\Gamma)} \in Q, x_i \in I_i, \forall i \in d(\Gamma)\right\} \tag{2.5}$$

and I_i contains $c(\{i\})$ for every i in $d(\Gamma)$. Denote $M := N \setminus d(\Gamma)$, $c_M := c|_M$, and

$$\Sigma_M := \{S \in \Sigma \,|\, S \subseteq M\}.$$

Due to the definition of dummy players, since $\Sigma \ni N$, we have that M belongs to Σ_M. If M and $\Sigma_M^+ \setminus \{M\}$ are not empty, then $\Gamma_D := (M, c_M, Q, \Sigma_M)$ forms a general cost allocation game, called the *dummy-free subgame* of Γ.

Definition 2.2.6. *For each cost allocation method Φ, the cost allocation method $\mathcal{D} \circ \Phi$, called the* dummy-friendly version *of Φ, is defined for any general cost allocation game Γ which satisfies that Γ and its dummy-free subgame Γ_D belong to the domain of Φ as follows*

$$\mathcal{D} \circ \Phi(\Gamma)_i := \begin{cases} c(\{i\}) & \text{if } i \in d(\Gamma) \\ \Phi(\Gamma_D)_i & \text{otherwise,} \end{cases} \quad \forall i \in N.$$

The operator $\mathcal{D}$ is called the dummy operator.

Directly from the definition, we have that $\mathcal{D} \circ \Phi(\Gamma) = \Phi(\Gamma)$ if the dummy players set $d(\Gamma)$ is empty and $\mathcal{D} \circ \Phi(\Gamma)_i = c(\{i\})$ for all $i \in N$ if $d(\Gamma) = N$.

Proposition 2.2.7. *The dummy operator $\mathcal{D}$ preserves validity, efficiency, user friendliness, core cost allocation, symmetry, and scalar multiplicativity.*

Proof. Let Φ be an arbitrary cost allocation method and $\Gamma = (N, c, P, \Sigma)$ be some general cost allocation game, for which Φ and $\mathcal{D} \circ \Phi$ are defined. If Γ does not have any dummy player, then there is nothing to prove. We assume that the set of all dummy players $d(\Gamma)$ is non-empty. Let Γ_D be the dummy-free subgame of Γ. Denote $M := N \backslash d(\Gamma)$ and $x := \mathcal{D} \circ \Phi(\Gamma)$. We have then

$$x_i = c(\{i\}),\ \forall i \in d(\Gamma).$$

If Φ is valid, then $\Phi(\Gamma_D)$ is a valid price of Γ_D. From this and (2.5) it follows that $\mathcal{D} \circ \Phi(\Gamma)$ is a valid price of Γ as well.

If Φ is efficient, then we have

$$x(M) = \Phi(\Gamma_D)(M) = c(M).$$

On the other hand, there holds

$$c(N) = c(M) + \sum_{i \in d(\Gamma)} c(\{i\}).$$

Combining them yields

$$x(N) = c(N),$$

i.e., x is an efficient price of Γ.

If Φ is user friendly, then there holds

$$x(T) = \Phi(\Gamma_D)(T) \leq c(T),\ \forall T \in \Sigma_M^+. \tag{2.6}$$

Moreover, for each set $S \in \Sigma^+$, the set $S \backslash d(\Gamma)$ either is empty or belongs to Σ_M^+. If the set $S \backslash d(\Gamma)$ is empty, then S is a subset of $d(\Gamma)$ and there holds

$$x(S) = \sum_{i \in S} x_i = \sum_{i \in S} c(\{i\}) = c(S).$$

On the other hand, if $S \backslash d(\Gamma) \in \Sigma_M^+$, then because of (2.6) we have

$$\begin{aligned} c(S) &= c\big(S \backslash d(\Gamma)\big) + \sum_{i \in S \cap d(\Gamma)} c(\{i\}) \qquad (2.7)\\ &\geq x\big(S \backslash d(\Gamma)\big) + \sum_{i \in S \cap d(\Gamma)} c(\{i\}) \\ &= x\big(S \backslash d(\Gamma)\big) + \sum_{i \in S \cap d(\Gamma)} x_i \\ &= x(S). \end{aligned}$$

Hence, $\mathcal{D} \circ \Phi(\Gamma)$ is an user friendly price of Γ.

If Φ is a core cost allocation method and the core of Γ is non-empty, then $y := \Phi(\Gamma)$ belongs to $\mathcal{C}(\Gamma)$. We want to prove that x belongs to $\mathcal{C}(\Gamma)$ as well. We have $y \in P$,

$$y(N) = c(N), \tag{2.8}$$

and

$$y(S) \leq c(S),\ \forall S \in \Sigma^+ \backslash \{N\}. \tag{2.9}$$

From these it follows that there hold for all $i \in d(\Gamma)$

$$\begin{aligned} c(\{i\}) &\geq y_i \\ &= y(N) - y(N\backslash\{i\}) \\ &= c(N) - y(N\backslash\{i\}) \\ &\geq c(N) - c(N\backslash\{i\}) \\ &= c(\{i\}), \end{aligned}$$

i.e., $y_i = c(\{i\})$. From this, (2.7), (2.8), and (2.9) it follows that

$$y(M) = c(M)$$

and

$$y(T) \leq c(T), \forall T \in \Sigma_M^+ \backslash \{M\}.$$

We also have that $y|_M$ belongs to Q. Hence, the core of Γ_D contains $y|_M$ and therefore is not empty. Since Φ is a core cost allocation method, $x|_M = \Phi(\Gamma_D)$ belongs to $\mathcal{C}(\Gamma_D)$. One can then easily prove that x belongs to $\mathcal{C}(\Gamma)$.

If two players i and j are equivalent regarding Γ, then either both of them are dummy players or none of them is a dummy player and they are equivalent regarding Γ_D. In the first case, we have that $\{i\}, \{j\} \in \Sigma$,

$$x_i = c(\{i\}) \text{ and } x_j = c(\{j\}).$$

Since $\{i\}, \{j\} \in \Sigma$ and i and j are equivalent regarding Γ, we have $c(\{i\}) = c(\{j\})$ and therefore $x_i = x_j$. In the second case, we have that i and j are equivalent regarding Γ_D and

$$x_i = \Phi(\Gamma_D)_i \text{ and } x_j = \Phi(\Gamma_D)_j.$$

If Φ is symmetric, then $x_i = x_j$. Therefore, if Φ is symmetric, then so is $\mathcal{D} \circ \Phi$.

Let λ be an arbitrary positive number, denote $x^\lambda := \Phi(\lambda\Gamma)$. Clearly, $d(\Gamma)$ is also the set of all dummy players of $\lambda\Gamma$. Hence, we have

$$x_i^\lambda = \lambda c(\{i\}) = \lambda x_i, \ \forall i \in d(\Gamma) = d(\lambda\Gamma).$$

On the other hand, let Γ_D^λ be the dummy-free subgame of $\lambda\Gamma$, we have $\Gamma_D^\lambda = \lambda\Gamma_D$. If Φ is scalar multiplicative, then there holds

$$x^\lambda|_M = \Phi(\Gamma_D^\lambda) = \lambda\Phi(\Gamma_D) = \lambda x|_M.$$

Therefore, it holds $x^\lambda = \lambda x$. □

The following table summarizes the properties of the four considered cost allocation methods:

	Non-coop	Prop	MinSub	Prop LC
Validity	No*	No*	Yes	Yes
Efficiency	No	Yes	No	Yes
Coalitional stability	Yes	No	No*	No
User friendliness	No	No	Yes	No
Core cost allocation	No	No	Yes	Yes
Monotonicity	Yes	No	No	No
Bounded variation	Yes	Yes*	No	No
Symmetry	Yes**	Yes**		
Additiviy	Yes	No	No	No
Scalar multiplicativity	Yes	Yes		
Dummy player	Yes	No	No	No

Table 2.3: Properties of cost allocation methods

Yes* means that the property holds for cost allocation games; Yes** means that the property holds for general cost allocation games whose sets of possible coalitions contain (the coalition of) each single player; while No* means that the property fails in general but holds for some large classes of general cost allocation games that appear in practice (see Subsection

2.2.1). Minimum subsidy and proportional least core cost allocation methods are actually two classes of cost allocation methods, since they are defined based on the sets of the optimal solutions of two optimization problems which may have more than one optimal solutions. Because of this non-uniqueness, symmetry and scalar multiplicativity may fail in general for these cost allocation methods. However, these properties hold for some of proportional least core cost allocation methods which will be considered in the next section. Proportional, minimum subsidy, and proportional least core cost allocation methods may not charge a dummy player exactly his individual cost, but their dummy-friendly versions do. Moreover, their dummy-friendly versions have all of their other properties due to Proposition 2.2.7.

2.3 Game Theoretical Concepts

Due to the previous section, one cannot construct in general an efficient, coalitionally stable core allocation method which has bounded variation. Even worse, at most two of these four properties can be simultaneously fulfilled. There are two way to proceed: One way is to consider more specific families of cost allocation games which could have better properties, another is to minimize the degree of axiomatic violation. Since specific cost allocation games occur seldom in practice, we choose the latter approach. We consider several game theoretical concepts, where the unfairness in the sense of coalitional instability is minimized.

A cooperation can only be realized if the common cost is allocated in a fair way and accepted by the users. Cooperative game theory analyzes the possible grouping of individuals to form their own real or theoretical coalitions. It provides mathematical tools to understand fair prices in the sense that a fair price prevents the collapse of the grand coalition and increases the stability of the cooperation. In this section, we present some generalizations of several well-known concepts from game theory and a new concept, namely, the (f, r)-least core.

2.3.1 The Core and the f-Least Core

The core is the most attractive solution concept in cooperative game theory. It is the set of valid, efficient allocations that cannot be improved upon by a subset (a coalition) of the economy's consumers.

Definition 2.3.1. *The core of the cost allocation game $\Gamma = (N, c, P, \Sigma)$, denoted by $\mathcal{C}(\Gamma)$, is the set of all valid, efficient, and user friendly prices of Γ*

$$\mathcal{C}(\Gamma) = \{x \in \mathcal{X}(\Gamma) \mid x(S) \le c(S),\ \forall S \in \Sigma \backslash \{\emptyset, N\}\}. \tag{2.10}$$

If the core of Γ is non-empty, then by using a price in the core there is no incentive for each subcoalition S of N to leave the grand coalition N and create its own subgame $\Gamma_S = (S, c_S, P_S, \Sigma_S)$ since the total price that S has to pay will not decrease thereby. Unfortunately, the core in general may be empty, e.g., the ticket-pricing problem that we consider in Chapter 8. Because of this reason, Shapley and Shubik generalized the core concept to the strong ε-core and weak ε-core [56], which are non-empty with suitable parameter ε. Following this line of reasoning, Maschler, Peleg, and Shapley [45] introduced in 1979 the least core, which is the intersection of all non-empty strong ε-cores. The least core is non-empty under the assumption that the imputation set $\mathcal{X}(\Gamma)$ is non-empty and it is a subset of the core in the case the core is non-empty. These concepts appear as special cases in the following definition.

Definition 2.3.2. *Given a cost allocation game $\Gamma = (N, c, P, \Sigma)$ and a weight function $f : \Sigma^+ \to \mathbb{R}_{>0}$. The set*

$$\mathcal{C}_{\varepsilon,f}(\Gamma) := \left\{x \in \mathcal{X}(\Gamma) \mid e_f(S, x) \ge \varepsilon,\ \forall S \in \Sigma^+ \backslash \{N\}\right\}$$

is called the (ε, f)-core *of Γ. In particular, $\mathcal{C}_{0,f}(\Gamma)$ is the* core *of Γ, $\mathcal{C}_{\varepsilon,1}(\Gamma)$ is the strong ε-core, and $\mathcal{C}_{\varepsilon,|\cdot|}(\Gamma)$ is the weak ε-core.*

Let $\varepsilon_f(\Gamma)$ be the optimal value of the following linear program

$$\begin{aligned} \max_{(x,\varepsilon)}\ & \varepsilon \\ s.t.\ \ & x(S) + \varepsilon f(S) \le c(S),\ \forall S \in \Sigma^+ \backslash \{N\} \\ & x \in \mathcal{X}(\Gamma). \end{aligned} \tag{2.11}$$

*The number $\varepsilon_f(\Gamma)$ is called the f-*least core radius *of Γ. The f-*least core *of the game Γ, denoted $\mathcal{LC}_f(\Gamma)$, is the set of all vectors $x \in \mathcal{X}(\Gamma)$ such that $(x, \varepsilon_f(\Gamma))$ is an optimal solution of the above linear program, i.e.,*

$$\mathcal{LC}_f(\Gamma) = \mathcal{C}_{\varepsilon_f(\Gamma),f}(\Gamma).$$

Equivalently, we have

$$\varepsilon_f(\Gamma) = \max_{x \in \mathcal{X}(\Gamma)} \min_{S \in \Sigma^+ \setminus \{N\}} e_f(S, x),$$

i.e., $\varepsilon_f(\Gamma)$ is the largest ε such that $\mathcal{C}_{\varepsilon,f}(\Gamma)$ is non-empty. In other words, the f-least core is the set of all vectors in $\mathcal{X}(\Gamma)$ that maximize the minimum f-excess of coalitions in $\Sigma^+ \setminus \{N\}$.

For each cost allocation game whose core is empty, there exists for every valid, efficient price at least one coalition such that its price exceeds its cost. In that case, what we only can do is to minimize the loss of such coalitions and thereby decrease the dissatisfaction of them. This is exactly the idea behind the f-least core. The f-least core can also be used for cost allocation games having non-empty core to determine fair prices. In that case, for each price vector in the core, no coalition has to pay more than its cost. However, the coalitions have different amounts of benefit. A price where most coalitions have a small or zero benefit and only a few coalitions have a significant cost saving will also be seen as unfair. The minimum f-excess $\varepsilon_f(x; \Gamma)$ of coalitions in $\Sigma^+ \setminus \{N\}$ at a price x,

$$\varepsilon_f(x; \Gamma) := \min_{S \in \Sigma^+ \setminus \{N\}} \frac{c(S) - x(S)}{f(S)},$$

is a global coalitions' gain indicator. An allocation x with a large $\varepsilon_f(x; \Gamma)$ makes the cooperation more attractive and gives each coalition the feeling that they are taken seriously by the decision maker. Maximizing $\varepsilon_f(x; \Gamma)$ prevents unrest among the users. Choosing a suitable weight function f is the job of the decision maker. Typically one can choose f equal to 1, or the cardinal function, or the cost function itself. The f-least core is called *least core* with $f = 1$, *weak least core* with $f = |\cdot|$, and *proportional least core* with $f = c$.

As we said, the original reason for the study of the f-least core is the existence of cost allocation games with empty core. The reader may wonder why we should calculate prices for such games, since for every valid, efficient price there exists at least a coalition such that its price exceeds its cost? For each of these coalitions creating its own cooperation would be a better alternative. Why don't we just consider the corresponding subgames and the game consisting of the remaining players? The first reason is that, in many games, the players are not allowed to form their real coalitions.

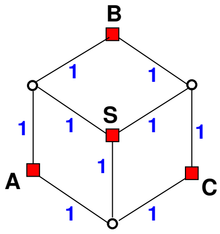

Figure 2.1: Steiner tree game

Considering theoretical coalitions provides alternatives in theory. For players, if they can find a better alternative even only in theory, then they will feel discriminated. However, if the decision maker can show that it is the best solution due to mathematical reasons, then the players may accept the price. The second reason is that, even if real coalitions are allowed, for some games the coalitions, for which creating its own cooperation is a better alternative than accepting a given price, overlap each other and therefore one cannot decide which coalitions among them should be realized. Let us consider a Steiner tree game of 3 players A, B, and C as in Figure 2.1. Each player has to be connected to the source S and the cost of each arc is as in the figure. The cost function is then

$$c(\{A\}) = c(\{B\}) = c(\{C\}) = 2,\ c(\{A, B, C\}) = 5,$$

$$c(\{A, B\}) = c(\{B, C\}) = c(\{C, A\}) = 3.$$

Since the players are symmetric, each of them should pay the same amount of 5/3. That mean the price for each coalition of two players is 10/3, which is larger than its cost of 3. However, one cannot decide which two players should leave the grand coalition to build their own cooperation, because the three players are identical. This game is totally unstable and the only solution for the three players is staying in the grand coalition. With the above price each player still has a benefit of 1/3 in comparison with acting alone.

In the following, we prove that the f-least core is well-defined.

Proposition 2.3.3. *For each cost allocation game* $\Gamma = (N, c, P, \Sigma)$*, if* $\mathcal{X}(\Gamma)$ *is non-empty then the* f*-least core of* Γ *is well-defined and non-empty.*

Proof. Due to the definition of a cost allocation game we have $\Sigma^+ \backslash \{N\} \neq \emptyset$. The polyhedron defined by the constraints of (2.11) is non-empty because with each $x \in \mathcal{X}(\Gamma)$, since $f(S) > 0$ for all S in Σ^+, we can choose ε sufficiently small such that

$$x(S) + \varepsilon f(S) \leq c(S), \ \forall S \in \Sigma^+ \backslash \{N\}.$$

On the other hand, since $x \in \mathcal{X}(\Gamma) \subseteq \mathbb{R}_+^N$, the objective value of (2.11) is bounded for each feasible solution (x, ε),

$$\varepsilon \leq \frac{c(S) - x(S)}{f(S)} \leq \frac{c(S)}{f(S)}, \ \forall S \in \Sigma^+ \backslash \{N\}.$$

Therefore, the linear program (2.11) has at least one optimal solution. Let ε^* be the optimal value, then the f-least core of Γ is

$$\{x \in \mathcal{X}(\Gamma) \,|\, (x, \varepsilon^*) \text{ is a feasible solution of (2.11)}\},$$

which is non-empty. □

Proposition 2.3.4. *Given a cost allocation game $\Gamma = (N, c, P, \Sigma)$ with a non-empty imputation set $\mathcal{X}(\Gamma)$. For $f = \alpha g + \beta c$, with $\alpha, \beta \geq 0$, $\alpha + \beta > 0$, and some function $g : \Sigma \to \mathbb{R}$ satisfying $g|_{\Sigma^+} > 0$, let ε^* be the f-least core radius of Γ. Then there holds $\beta\varepsilon^* \leq 1$.*

Proof. If ε^* is negative, then clearly the proposition holds. We consider the case that ε^* is non-negative. The f-least core of Γ is non-empty due to Proposition 2.3.3. For any $x^* \in \mathcal{LC}_f(\Gamma) \subseteq \mathbb{R}_+^N$, there holds for every set S in $\Sigma^+ \backslash \{N\}$

$$0 \leq x^*(S) \leq c(S) - \varepsilon^* f(S) \leq c(S) - \varepsilon^* \beta c(S).$$

Since the cost function is positive, from this it follows that $\beta\varepsilon^* \leq 1$. □

The core and the f-least core of general cost allocation games are defined in the same way as the ones of cost allocation games. However, the non-emptiness of the imputation set is not sufficient to guarantee that the f-least core is well-defined. The following property is needed.

Definition 2.3.5. *A general cost allocation game $\Gamma := (N, c, P, \Sigma)$ is called* bounded *if there exists a finite number M such that for each vector $x \in \mathcal{X}(\Gamma)$ there exists a set $S \in \Sigma^+ \backslash \{N\}$ satisfying $x(S) \geq M$. Clearly, each cost allocation game is bounded with $M = 0$.*

Remark 2.3.6. *As we will see in Proposition 2.3.8, each general cost allocation game* $\Gamma := (N, c, P, \Sigma)$ *with* $\Sigma = 2^N$ *is bounded. If* $\Sigma \neq 2^N$*, then it does not hold in general. We consider a general cost allocation game with* $N = \{1, 2\}$*,* $P = \mathbb{R}^N$*, and* $\Sigma = \big\{\emptyset, \{1\}, N\big\}$*. We have* $\Sigma^+ \backslash \{N\} = \big\{\{1\}\big\}$ *and* $x(\{1\})$ *is not bounded from below for* $x \in \mathcal{X}(\Gamma)$*. Therefore, the linear program* (2.11) *is unbounded for this game. Consequently, the* f*-least core of* Γ *is not well-defined.*

Definition 2.3.7. *Let* N *be a given finite set. For* $S \subseteq N$*, let* $\chi_S \in \{0, 1\}^N$ *denote the incidence vector of* S*, i.e.,* χ_S^i *is 1 if* $i \in S$ *and 0 else. A family* $\Sigma \subseteq 2^N$ *is called a* partitioning family *of* N *if there exist* $S_i \in \Sigma^+ \backslash \{N\}$ *and* $\lambda_i \in \mathbb{R}_{>0}$ *for* $i = 1, 2, \ldots, k$ *such that*

$$\sum_{i=1}^{k} \lambda_i \chi_{S_i} = \chi_N.$$

Clearly, the power set 2^N *of* N *is a partitioning family of* N*.*

Proposition 2.3.8. *For each general cost allocation game* $\Gamma = (N, c, P, \Sigma)$*, if* Σ *is a partitioning family of* N*, then* Γ *is bounded.*

Proof. Let x be an arbitrary vector in $\mathcal{X}(\Gamma)$. We have that $x(N) = c(N)$. On the other hand, since Σ is a partitioning family of N, there exist $S_i \in \Sigma^+ \backslash \{N\}$ and $\lambda_i \in \mathbb{R}_{>0}$ for $i = 1, 2, \ldots, k$ such that

$$\sum_{i=1}^{k} \lambda_i \chi_{S_i} = \chi_N.$$

From this it follows that

$$c(N) = x(N) = \sum_{i=1}^{k} \lambda_i x(S_i).$$

Clearly, there exists $j \in \{1, 2, \ldots, k\}$ such that

$$x(S_j) \geq \min \left\{0, \frac{c(N)}{\lambda_1}\right\},$$

i.e., Γ is bounded. □

Proposition 2.3.9. *The f-least core of a general cost allocation game $\Gamma = (N, c, P, \Sigma)$ is well-defined and non-empty iff $\mathcal{X}(\Gamma)$ is non-empty and Γ is bounded.*

Proof. If the f-least core of $\Gamma = (N, c, P, \Sigma)$ is well-defined, then the following linear program has optimal solutions with the optimal value $\varepsilon_f(\Gamma)$

$$\begin{aligned} &\max_{(x,\varepsilon)} \ \varepsilon \\ s.t. \ \ &x(S) + \varepsilon f(S) \leq c(S), \ \forall S \in \Sigma^+ \backslash \{N\} \\ &x \in \mathcal{X}(\Gamma). \end{aligned} \tag{2.12}$$

Therefore, $\mathcal{X}(\Gamma)$ is non-empty and for each vector $x \in \mathcal{X}(\Gamma)$ there exists a set $T \in \Sigma^+ \backslash \{N\}$ such that

$$x(T) + \varepsilon_f(\Gamma) f(T) = c(T),$$

i.e.,

$$x(T) = c(T) - \varepsilon_f(\Gamma) f(T) \geq \min_{S \in \Sigma^+ \backslash \{N\}} \big(c(S) - \varepsilon_f(\Gamma) f(S)\big) > -\infty.$$

This means that Γ is bounded.

With $\mathcal{X}(\Gamma)$ is non-empty and Γ is bounded, we want to prove that the linear program (2.12) has optimal solutions. The linear program (2.12) is feasible because with each $x \in \mathcal{X}(\Gamma)$, since $f(S) > 0$ for all S in Σ^+, we can choose ε sufficiently small such that

$$x(S) + \varepsilon f(S) \leq c(S), \ \forall S \in \Sigma^+ \backslash \{N\}.$$

Since Γ is bounded, there exists a finite number M such that

$$\forall x \in \mathcal{X}(\Gamma), \exists S \in \Sigma^+ \backslash \{N\} : \ x(S) \geq M.$$

Let (x, ε) be an arbitrary feasible solution of (2.12), then there exists a set $T \in \Sigma^+ \backslash \{N\}$ such that $x(T) \geq M$. From this it follows that

$$\varepsilon \leq \frac{c(T) - x(T)}{f(T)} \leq \frac{c(T) - M}{f(T)} \leq \max_{S \in \Sigma^+ \backslash \{N\}} \frac{c(S) - M}{f(S)}.$$

This means that the linear program (2.12) is bounded from above. And therefore it has optimal solutions and the f-least core of Γ is well-defined and non-empty. $\square$

The following proposition shows that the f-least core of a general cost allocation game $\Gamma = (N, c, \mathbb{R}^N, \Sigma)$ is well-defined if and only if Σ is a partitioning family of N.

Proposition 2.3.10. *Given a finite set N, a family $\Sigma \subseteq 2^N$ satisfying $\Sigma^+\backslash\{N\} \neq \emptyset$, a function $u : \Sigma \to \mathbb{R}$, and a weight function $f : \Sigma^+ \to \mathbb{R}_{>0}$. The following linear program*

$$\begin{aligned} &\max_{(x,\varepsilon)} \ \varepsilon \\ s.t. \ & x(S) + \varepsilon f(S) \leq u(S), \ \forall S \in \Sigma^+\backslash\{N\} \\ & x(N) = u(N) \end{aligned} \tag{2.13}$$

is bounded if and only if Σ is a partitioning family of N.

Proof. The linear program (2.13) is feasible because with each $x \in \mathbb{R}^N$ satisfying $x(N) = u(N)$, since $f(S) > 0$ for all S in Σ^+, we can choose ε sufficiently small such that

$$x(S) + \varepsilon f(S) \leq u(S), \ \forall S \in \Sigma^+\backslash\{N\}.$$

Therefore, if it is bounded, then it has optimal solutions. That means, due to the duality theorem, the dual problem

$$\begin{aligned} &\min_{(\lambda,\mu)} \sum_{S \in \Sigma^+\backslash\{N\}} \nu_S u(S) + \mu u(N) \\ s.t. \ & \sum_{\substack{S \in \Sigma^+\backslash\{N\} \\ S \ni i}} \nu_S + \mu = 0, \ \forall i \in N \\ & \sum_{S \in \Sigma^+\backslash\{N\}} \nu_S f(S) = 1 \\ & \nu_S \geq 0, \ \forall S \in \Sigma^+\backslash\{N\} \end{aligned}$$

is feasible. Let (λ, μ) be a feasible solution of the dual problem. Since ν is non-negative,

$$\sum_{S \in \Sigma^+\backslash\{N\}} \nu_S f(S) = 1 > 0,$$

and

$$f(S) > 0, \ \forall S \in \Sigma^+,$$

the set Ω defined by

$$\Omega := \{S \in \Sigma^+ \backslash \{N\} \mid \nu_S > 0\}$$

is non-empty. Let T be a set in Ω and j be an element of T, we have then

$$\mu = -\sum_{\substack{S \in \Sigma^+ \backslash \{N\} \\ S \ni j}} \nu_S \leq -\nu_T < 0.$$

Define for each $S \in \Omega$

$$\lambda_S = -\frac{\nu_S}{\mu},$$

then we have that $(\lambda_S)_{S \in \Omega}$ is positive and

$$\sum_{S \in \Omega} \lambda_S \chi_S = \chi_N,$$

i.e., Σ is a partitioning family of N.

Assume now that Σ is a partitioning family of N. The linear program (2.13) is bounded due to Proposition 2.3.8 and the proof of Proposition 2.3.9. □

Definition 2.3.11. *A cost allocation method defined for every bounded general cost allocation game with a non-empty imputation set whose output belongs to the f-least core for some weight function f is called a* f-least core cost allocation method.

In the following, we present some properties of the f-least core and f-least core cost allocation methods beside the two obvious ones, namely, validity and efficiency.

Core cost allocation

Proposition 2.3.12. *For any bounded general cost allocation game $\Gamma = (N, c, P, \Sigma)$, if the core of Γ is non-empty, then there holds*

$$\mathcal{LC}_f(\Gamma) \subseteq \mathcal{C}(\Gamma).$$

Proof. If the core is non-empty then the f-least core radius is non-negative and therefore each vector in the f-least core belongs also to the core. □

Scalar multiplicativity

Proposition 2.3.13. *For any bounded general cost allocation game* $\Gamma = (N, c, P, \Sigma)$ *with a non-empty imputation set* $\mathcal{X}(\Gamma)$*, any weight function* $f : \Sigma^+ \to \mathbb{R}_{>0}$*, and any positive number* λ*, there holds*

$$\mathcal{LC}_f(\lambda\Gamma) = \lambda\mathcal{LC}_f(\Gamma).$$

Proof. Let ε^* be the f-least core radius of Γ and δ^* be the f-least core radius of $\lambda\Gamma$. Due to the definition, we have that δ^* is the optimal value of the following linear program

$$\begin{aligned} &\max_{(y,\delta)} \ \delta && (2.14)\\ s.t.\ \ & y(S) + \delta f(S) \leq \lambda c(S),\ \forall S \in \Sigma^+ \backslash \{N\} \\ & y \in \mathcal{X}(\lambda\Gamma). \end{aligned}$$

Since $\mathcal{X}(\lambda\Gamma) = \lambda\mathcal{X}(\Gamma)$, by setting $\varepsilon = \frac{\delta}{\lambda}$ and $x = \frac{1}{\lambda}y$, the linear program (2.14) is equivalent to the following one

$$\begin{aligned} &\lambda \max_{(x,\varepsilon)} \ \varepsilon && (2.15)\\ s.t.\ \ & x(S) + \varepsilon f(S) \leq c(S),\ \forall S \in \Sigma^+ \backslash \{N\} \\ & x \in \mathcal{X}(\Gamma). \end{aligned}$$

From this it follows that

$$\lambda\varepsilon^* = \delta^*$$

and

$$\mathcal{LC}_f(\lambda\Gamma) = \lambda\mathcal{LC}_f(\Gamma).$$

□

Proposition 2.3.14. *For any bounded general cost allocation game* $\Gamma = (N, c, P, \Sigma)$ *with a non-empty imputation set* $\mathcal{X}(\Gamma)$*, any weight function* $f : \Sigma^+ \to \mathbb{R}_{>0}$*, and any positive number* μ*, there holds*

$$\mathcal{LC}_{\mu f}(\Gamma) = \mathcal{LC}_f(\Gamma).$$

Proof. Let ε^* be the f-least core radius of Γ and $\tilde{\varepsilon}$ be the μf-least core radius of Γ. Similarly to the proof of Proposition 2.3.13, one can prove that $\mu\tilde{\varepsilon} = \varepsilon^*$ and

$$\mathcal{LC}_{\mu f}(\Gamma) = \mathcal{LC}_f(\Gamma). \qquad \square$$

Combining Proposition 2.3.13 and Proposition 2.3.14 gives us the following result.

Proposition 2.3.15. *For any bounded general cost allocation game $\Gamma = (N, c, P, \Sigma)$ with a non-empty imputation set $\mathcal{X}(\Gamma)$, any weight function $f : \Sigma^+ \to \mathbb{R}_{>0}$, and any positive numbers λ and μ, there holds*

$$\mathcal{LC}_{\mu f}(\lambda\Gamma) = \lambda\mathcal{LC}_f(\Gamma).$$

Using the scalar multiplicativity, we may scale the cost function and the weight function in order to improve the numerical stability when calculating the f-least core.

Stability

Due to Proposition 2.3.12, each f-least core cost allocation method is a core cost allocation method. As we showed before in Section 2.2.2, in general, no core cost allocation method has uniformly bounded variation. However, the f-least core has another stability property in the case $f = c$. Let $\Gamma = (N, c, P, \Sigma)$ be a cost allocation game and $\tilde{\Gamma} = (N, \tilde{c}, \tilde{P}, \Sigma)$ with $\tilde{P} = \frac{\tilde{c}(N)}{c(N)} P$ be a perturbed cost allocation game of Γ. Assume that the imputation sets of Γ and $\tilde{\Gamma}$ are non-empty. Let $\alpha_2 \geq \alpha_1 > -1$ be real numbers that satisfy

$$(1 + \alpha_1)c(S) \leq \tilde{c}(S) \leq (1 + \alpha_2)c(S),\ \forall S \in \Sigma^+ \backslash \{N\}.$$

Denote $\beta := \frac{\tilde{c}(N)}{c(N)} - 1$. Since $c(N) > 0$ and $\tilde{c}(N) > 0$, we have that $\beta > -1$. Let ε be the c-least core radius of Γ, $\tilde{\varepsilon}$ be the $\tilde{c}$-least core radius of $\tilde{\Gamma}$, x be an arbitrary point in $\mathcal{LC}_c(\Gamma)$, and $\tilde{x}$ be an arbitrary point in $\mathcal{LC}_{\tilde{c}}(\tilde{\Gamma})$. We have then $x \in \mathcal{X}(\Gamma)$, $\tilde{x} \in \mathcal{X}(\tilde{\Gamma})$,

$$\min_{S \in \Sigma^+ \backslash \{N\}} \frac{c(S) - x(S)}{c(S)} = \varepsilon$$

and

$$\min_{S \in \Sigma^+ \backslash \{N\}} \frac{\tilde{c}(S) - \tilde{x}(S)}{\tilde{c}(S)} = \tilde{\varepsilon}.$$

Because of the definition of β and since $\tilde{P} = (1+\beta)P$, we have that

$$(1+\beta)x \in \mathcal{X}(\tilde{\Gamma}) \quad \text{and} \quad \frac{1}{1+\beta}\tilde{x} \in \mathcal{X}(\Gamma).$$

Let $\tilde{\delta}(x)$ be the largest number such that $(1+\beta)x \in \mathcal{C}_{\tilde{\delta}(x),\tilde{c}}(\tilde{\Gamma})$, i.e.,

$$\tilde{\delta}(x) = \min_{S \in \Sigma^+ \setminus \{N\}} \frac{\tilde{c}(S) - (1+\beta)x(S)}{\tilde{c}(S)},$$

and $\delta(\tilde{x})$ be the largest number such that $\frac{1}{1+\beta}\tilde{x} \in \mathcal{C}_{\delta(\tilde{x}),c}(\Gamma)$, i.e.,

$$\delta(\tilde{x}) = \min_{S \in \Sigma^+ \setminus \{N\}} \frac{c(S) - \frac{1}{1+\beta}\tilde{x}(S)}{c(S)}.$$

In general, $(1+\beta)x$ and $\frac{1}{1+\beta}\tilde{x}$ may lie outside of $\mathcal{LC}_{\tilde{c}}(\tilde{\Gamma})$ and $\mathcal{LC}_c(\Gamma)$, respectively, i.e.,

$$\tilde{\delta}(x) < \tilde{\varepsilon} \text{ and } \delta(\tilde{x}) < \varepsilon.$$

However, the following result shows that the gaps $\tilde{\varepsilon} - \tilde{\delta}(x)$ and $\varepsilon - \delta(\tilde{x})$ are bounded by an order of $\alpha_2 - \alpha_1$. Therefore, these gaps are small if the difference between α_2 and α_1 is sufficiently small. In other words, in that case, $(1+\beta)\mathcal{LC}_c(\Gamma)$ is a "good" approximation of $\mathcal{LC}_{\tilde{c}}(\tilde{\Gamma})$ and vice versa.

Proposition 2.3.16. *For any cost allocation games $\Gamma = (N, c, P, \Sigma)$ and $\tilde{\Gamma} = (N, \tilde{c}, \tilde{P}, \Sigma)$ whose imputation sets are non-empty and numbers $\alpha_2 \geq \alpha_1 > -1$ that satisfy*

$$\tilde{P} = \frac{\tilde{c}(N)}{c(N)} P$$

and

$$(1+\alpha_1)c(S) \leq \tilde{c}(S) \leq (1+\alpha_2)c(S), \ \forall S \in \Sigma^+ \setminus \{N\},$$

let ε, $\tilde{\varepsilon}$, δ, and $\tilde{\delta}$ be defined as above, then there hold

$$0 \leq \tilde{\varepsilon} - \tilde{\delta}(x) \leq (\alpha_2 - \alpha_1)\frac{(1+\beta)(1-\varepsilon)}{(1+\alpha_1)(1+\alpha_2)} \leq (\alpha_2 - \alpha_1)\frac{1-\tilde{\varepsilon}}{1+\alpha_1}, \ \forall x \in \mathcal{LC}_c(\Gamma)$$

and

$$0 \leq \varepsilon - \delta(\tilde{x}) \leq (\alpha_2 - \alpha_1)\frac{1-\tilde{\varepsilon}}{1+\beta} \leq (\alpha_2 - \alpha_1)\frac{1-\varepsilon}{1+\alpha_1}, \ \forall \tilde{x} \in \mathcal{LC}_{\tilde{c}}(\tilde{\Gamma}).$$

Proof. Denote $\beta := \frac{\tilde{c}(N)}{c(N)} - 1$. Since $c(N) > 0$ and $\tilde{c}(N) > 0$, we have that $\beta > -1$. Let x be an arbitrary vector in $\mathcal{LC}_c(\Gamma)$ and $\tilde{x}$ be an arbitrary point in $\mathcal{LC}_{\tilde{c}}(\tilde{\Gamma})$. Because $\tilde{P} = (1+\beta)P$,

$$\frac{1}{1+\beta}\tilde{x}(N) = \frac{1}{1+\beta}\tilde{c}(N) = c(N),$$

and

$$(1+\beta)x(N) = (1+\beta)c(N) = \tilde{c}(N),$$

we have that $\frac{1}{1+\beta}\tilde{x} \in \mathcal{X}(\Gamma)$ and $(1+\beta)x \in \mathcal{X}(\tilde{\Gamma})$. Hence, there hold

$$\min_{S\in\Sigma^+\setminus\{N\}} \frac{c(S) - \frac{1}{1+\beta}\tilde{x}(S)}{c(S)} \leq \max_{y\in\mathcal{X}(\Gamma)} \min_{S\in\Sigma^+\setminus\{N\}} \frac{c(S)-y(S)}{c(S)} = \varepsilon \tag{2.16}$$

and

$$\tilde{\delta}(x) = \min_{S\in\Sigma^+\setminus\{N\}} \frac{\tilde{c}(S) - (1+\beta)x(S)}{\tilde{c}(S)} \leq \max_{y\in\mathcal{X}(\tilde{\Gamma})} \min_{S\in\Sigma^+\setminus\{N\}} \frac{\tilde{c}(S)-y(S)}{\tilde{c}(S)} = \tilde{\varepsilon}. \tag{2.17}$$

From (2.16), $1+\beta > 0$, and the non-negativity of $\tilde{x}$, it follows that

$$\begin{aligned}
(1+\beta)\varepsilon &\geq \min_{S\in\Sigma^+\setminus\{N\}} \frac{(1+\beta)c(S) - \tilde{x}(S)}{c(S)} \\
&= \min_{S\in\Sigma^+\setminus\{N\}} \left(1+\beta - \frac{\tilde{x}(S)}{c(S)}\right) \\
&\geq \min_{S\in\Sigma^+\setminus\{N\}} \left(1+\beta - \frac{\tilde{x}(S)}{\frac{1}{1+\alpha_2}\tilde{c}(S)}\right) \\
&= \beta - \alpha_2 + (1+\alpha_2) \min_{S\in\Sigma^+\setminus\{N\}} \left(1 - \frac{\tilde{x}(S)}{\tilde{c}(S)}\right) \\
&= \beta - \alpha_2 + (1+\alpha_2)\tilde{\varepsilon},
\end{aligned}$$

i.e.,

$$\tilde{\varepsilon} \leq \frac{\alpha_2 - \beta}{1+\alpha_2} + \frac{1+\beta}{1+\alpha_2}\varepsilon. \tag{2.18}$$

On the other hand, we have

$$\begin{aligned}
\tilde{\delta}(x) &= \min_{S\in\Sigma^+\setminus\{N\}} \frac{\tilde{c}(S)-(1+\beta)x(S)}{\tilde{c}(S)} \\
&= \min_{S\in\Sigma^+\setminus\{N\}} \left(1-\frac{(1+\beta)x(S)}{\tilde{c}(S)}\right) \\
&\geq \min_{S\in\Sigma^+\setminus\{N\}} \left(1-\frac{(1+\beta)x(S)}{(1+\alpha_1)c(S)}\right) \\
&= \frac{\alpha_1-\beta}{1+\alpha_1}+\frac{1+\beta}{1+\alpha_1}\min_{S\in\Sigma^+\setminus\{N\}}\left(1-\frac{x(S)}{c(S)}\right) \\
&= \frac{\alpha_1-\beta}{1+\alpha_1}+\frac{1+\beta}{1+\alpha_1}\varepsilon. \qquad (2.19)
\end{aligned}$$

Combining (2.17)–(2.19) yields

$$\frac{\alpha_1-\beta}{1+\alpha_1}+\frac{1+\beta}{1+\alpha_1}\varepsilon \leq \tilde{\delta}(x) \leq \tilde{\varepsilon} \leq \frac{\alpha_2-\beta}{1+\alpha_2}+\frac{1+\beta}{1+\alpha_2}\varepsilon. \qquad (2.20)$$

Denote

$$\kappa := \frac{1}{1+\alpha_1}-\frac{1}{1+\alpha_2}=\frac{\alpha_2-\alpha_1}{(1+\alpha_1)(1+\alpha_2)},$$

we have

$$\tilde{\varepsilon}-\tilde{\delta}(x)\geq 0$$

and

$$\begin{aligned}
\tilde{\varepsilon}-\tilde{\delta}(x) &\leq \left(\frac{\alpha_2-\beta}{1+\alpha_2}+\frac{1+\beta}{1+\alpha_2}\varepsilon\right)-\left(\frac{\alpha_1-\beta}{1+\alpha_1}+\frac{1+\beta}{1+\alpha_1}\varepsilon\right) \\
&= \frac{\alpha_2}{1+\alpha_2}-\frac{\alpha_1}{1+\alpha_1}+\beta\kappa-(1+\beta)\kappa\varepsilon \\
&= \left(1-\frac{1}{1+\alpha_2}\right)-\left(1-\frac{1}{1+\alpha_1}\right)+\beta\kappa-(1+\beta)\kappa\varepsilon \\
&= (1+\beta)\kappa(1-\varepsilon) \\
&= (\alpha_2-\alpha_1)\frac{(1+\beta)(1-\varepsilon)}{(1+\alpha_1)(1+\alpha_2)}. \qquad (2.21)
\end{aligned}$$

To estimate $\varepsilon-\delta(\tilde{x})$ we just have to swap the role of Γ and $\tilde{\Gamma}$ in the above proof. Denote

$$\tilde{\alpha}_1 := -\frac{\alpha_1}{1+\alpha_1}, \quad \tilde{\alpha}_2 := -\frac{\alpha_2}{1+\alpha_2}, \quad \text{and} \quad \tilde{\beta} := -\frac{\beta}{1+\beta},$$

we have then

$$(1+\tilde{\alpha}_2)\tilde{c}(S) \leq c(S) \leq (1+\tilde{\alpha}_1)\tilde{c}(S),\ \forall S \in \Sigma^+\backslash\{N\},$$

$$\tilde{\beta} = \frac{c(N)}{\tilde{c}(N)} - 1,$$

and

$$\delta(\tilde{x}) = \min_{S\in\Sigma^+\backslash\{N\}} \frac{c(S) - (1+\tilde{\beta})\tilde{x}(S)}{c(S)},$$

Due to (2.20), we have

$$\frac{\tilde{\alpha}_2 - \tilde{\beta}}{1+\tilde{\alpha}_2} + \frac{1+\tilde{\beta}}{1+\tilde{\alpha}_2}\tilde{\varepsilon} \leq \delta(\tilde{x}) \leq \varepsilon \leq \frac{\tilde{\alpha}_1 - \tilde{\beta}}{1+\tilde{\alpha}_1} + \frac{1+\tilde{\beta}}{1+\tilde{\alpha}_1}\tilde{\varepsilon}. \tag{2.22}$$

Since

$$1+\tilde{\alpha}_i = \frac{1}{1+\alpha_i},\ i \in \{1,2\}$$

and

$$1+\tilde{\beta} = \frac{1}{1+\beta},$$

there hold for $i \in \{1,2\}$

$$\frac{\tilde{\alpha}_i - \tilde{\beta}}{1+\tilde{\alpha}_i} = \frac{(1+\tilde{\alpha}_i) - (1+\tilde{\beta})}{1+\tilde{\alpha}_i} = (1+\alpha_i)\left(\frac{1}{1+\alpha_i} - \frac{1}{1+\beta}\right) = \frac{\beta - \alpha_i}{1+\beta},$$

and

$$\frac{1+\tilde{\beta}}{1+\tilde{\alpha}_i} = \frac{1+\alpha_i}{1+\beta}.$$

Therefore, the inequalities (2.22) are equivalent to

$$\frac{\beta - \alpha_2}{1+\beta} + \frac{1+\alpha_2}{1+\beta}\tilde{\varepsilon} \leq \delta(\tilde{x}) \leq \varepsilon \leq \frac{\beta - \alpha_1}{1+\beta} + \frac{1+\alpha_1}{1+\beta}\tilde{\varepsilon}. \tag{2.23}$$

From this it follows that

$$0 \leq \varepsilon - \delta(\tilde{x}) \leq \left(\frac{\beta - \alpha_1}{1+\beta} + \frac{1+\alpha_1}{1+\beta}\tilde{\varepsilon}\right) - \left(\frac{\beta - \alpha_2}{1+\beta} + \frac{1+\alpha_2}{1+\beta}\tilde{\varepsilon}\right),$$

i.e.,

$$0 \leq \varepsilon - \delta(\tilde{x}) \leq (\alpha_2 - \alpha_1)\frac{1-\tilde{\varepsilon}}{1+\beta}. \tag{2.24}$$

From (2.23) we have that

$$1-\varepsilon \le 1-\frac{\beta-\alpha_2}{1+\beta}-\frac{1+\alpha_2}{1+\beta}\tilde{\varepsilon}=\frac{1+\alpha_2}{1+\beta}(1-\tilde{\varepsilon}).$$

From this and (2.21) it follows that

$$\tilde{\varepsilon}-\tilde{\delta}(x) \le (\alpha_2-\alpha_1)\frac{(1+\beta)(1-\varepsilon)}{(1+\alpha_1)(1+\alpha_2)} \le (\alpha_2-\alpha_1)\frac{1-\tilde{\varepsilon}}{1+\alpha_1}.$$

Similarly, from (2.20) we have

$$1-\tilde{\varepsilon} \le 1-\frac{\alpha_1-\beta}{1+\alpha_1}-\frac{1+\beta}{1+\alpha_1}\varepsilon=\frac{1+\beta}{1+\alpha_1}(1-\varepsilon).$$

From this and (2.24) it follows that

$$0 \le \varepsilon-\delta(\tilde{x}) \le (\alpha_2-\alpha_1)\frac{1-\tilde{\varepsilon}}{1+\beta} \le (\alpha_2-\alpha_1)\frac{1-\varepsilon}{1+\alpha_1}.$$

□

Symmetry

Proposition 2.3.17. *For any bounded general cost allocation game* $\Gamma = (N, c, P, \Sigma)$ *with a non-empty imputation set* $\mathcal{X}(\Gamma)$ *and two equivalent players* i *and* j *and any weight function* $f : \Sigma^+ \to \mathbb{R}_{>0}$ *satisfying*

$$f(S \cup \{i\}) = f(S \cup \{j\}),\ \forall S \subseteq N \backslash \{i,j\} : S \cup \{i\} \in \Sigma,$$

if a vector x *belongs to the* f*-least core of* Γ*, then so do* x^{ij} *and* $\frac{1}{2}(x + x^{ij})$*, where*

$$x_k^{ij} = \begin{cases} x_j & \text{if } k = i \\ x_i & \text{if } k = j \\ x_k & \text{otherwise.} \end{cases}$$

Proof. Since x belongs to $\mathcal{LC}_f(\Gamma)$, x belongs to P and $x(N) = c(N)$. For each set $S \subseteq N$ and player $k \in N$, denote $S_k := S \cup \{k\}$. Since x belongs to P and i and j are equivalent, x^{ij} belongs to P as well. On the other hand, there holds

$$x^{ij}(N) = x(N) = c(N)$$

and hence x^{ij} belongs to $\mathcal{X}(\Gamma)$. Moreover, since i and j are equivalent, we have

$$\forall S \subseteq N \backslash \{i,j\} : S_i \in \Sigma \Leftrightarrow S_j \in \Sigma$$

and

$$c(S_i) = c(S_j),\ \forall S \subseteq N \backslash \{i,j\} : S_i \in \Sigma.$$

From these and the definition of x^{ij} it follows that

$$\frac{c(S) - x^{ij}(S)}{f(S)} = \frac{c(S) - x(S)}{f(S)},\ \forall S \in \Sigma^+ : \{i,j\} \subseteq S \text{ or } \{i,j\} \cap S = \emptyset,$$

$$\frac{c(S_i) - x^{ij}(S_i)}{f(S_i)} = \frac{c(S_j) - x(S_j)}{f(S_j)},\ \forall S \subseteq N \backslash \{i,j\} : S_i \in \Sigma,$$

and

$$\frac{c(S_j) - x^{ij}(S_j)}{f(S_j)} = \frac{c(S_i) - x(S_i)}{f(S_i)},\ \forall S \subseteq N \backslash \{i,j\} : S_j \in \Sigma.$$

Combining the above equalities yields

$$\min_{S \in \Sigma^+ \backslash \{N\}} \frac{c(S) - x^{ij}(S)}{f(S)} = \min_{S \in \Sigma^+ \backslash \{N\}} \frac{c(S) - x(S)}{f(S)},$$

i.e., x^{ij} belongs to $\mathcal{LC}_f(\Gamma)$.

Let ε be the f-least core radius of Γ, i.e.,

$$\varepsilon = \min_{S \in \Sigma^+ \backslash \{N\}} \frac{c(S) - x(S)}{f(S)} = \max_{z \in \mathcal{X}(\Gamma)} \min_{S \in \Sigma^+ \backslash \{N\}} \frac{c(S) - z(S)}{f(S)}.$$

Denote $y := \frac{1}{2}(x + x^{ij})$. As x and x^{ij} belong to $\mathcal{X}(\Gamma)$, so does y. Hence, there holds

$$\min_{S \in \Sigma^+ \backslash \{N\}} \frac{c(S) - y(S)}{f(S)} \leq \varepsilon.$$

On the other hand, we have

$$\begin{aligned}
\min_{S \in \Sigma^+ \backslash \{N\}} \frac{c(S) - y(S)}{f(S)} &= \frac{1}{2} \min_{S \in \Sigma^+ \backslash \{N\}} \left(\frac{c(S) - x(S)}{f(S)} + \frac{c(S) - x^{ij}(S)}{f(S)} \right) \\
&\geq \frac{1}{2} \left(\min_{S \in \Sigma^+ \backslash \{N\}} \frac{c(S) - x(S)}{f(S)} + \min_{S \in \Sigma^+ \backslash \{N\}} \frac{c(S) - x^{ij}(S)}{f(S)} \right) \\
&= \min_{S \in \Sigma^+ \backslash \{N\}} \frac{c(S) - x(S)}{f(S)} \\
&= \varepsilon.
\end{aligned}$$

From these it follows that

$$\min_{S \in \Sigma^+ \setminus \{N\}} \frac{c(S) - y(S)}{f(S)} = \varepsilon,$$

i.e., y belongs to $\mathcal{LC}_f(\Gamma)$. $\square$

Dummy player

Proposition 2.3.18. *For any general cost allocation game* $\Gamma = (N, c, P, \Sigma)$ *with a non-empty core, each core cost allocation method defined for* Γ *charges dummy players exactly the amount of their individual costs.*

Proof. Let k be a dummy player of Γ, we have then

$$\forall S \subseteq N \setminus \{k\} : S \cup \{k\} \in \Sigma \Leftrightarrow S \in \Sigma$$

and

$$c(S \cup \{k\}) = c(S) + c(\{k\}),\ \forall S \in \Sigma, S \not\ni k.$$

Since $N \in \Sigma$, the set $N \setminus \{k\}$ belongs also to Σ. Let Φ be a core cost allocation method. Since the core $\mathcal{C}(\Gamma)$ of Γ is non-empty, we have $\Phi(\Gamma)$ belongs to $\mathcal{C}(\Gamma)$. Hence, there holds

$$\Phi(\Gamma)_k \leq c(\{k\}),$$

$$\sum_{i \in N \setminus \{k\}} \Phi(\Gamma)_i \leq c(N \setminus \{k\}),$$

and

$$\sum_{i \in N} \Phi(\Gamma)_i = c(N).$$

Therefore, we have

$$\begin{aligned} c(\{k\}) \geq \Phi(\Gamma)_k &= \sum_{i \in N} \Phi(\Gamma)_i - \sum_{i \in N \setminus \{k\}} \Phi(\Gamma)_i \\ &\geq c(N) - c(N \setminus \{k\}) \\ &= c(\{k\}), \end{aligned}$$

i.e.,

$$\Phi(\Gamma)_k = c(\{k\}).$$

$\square$

The above proposition shows that each f-least core allocation method charges the dummy players of any bounded general cost allocation game with a non-empty core exactly their costs. It remains to consider games with empty core.

Proposition 2.3.19. *For any given bounded general cost allocation game $\Gamma = (N, c, \mathbb{R}^N, \Sigma)$ which has a non-empty imputation set $\mathcal{X}(\Gamma)$ and an empty core, any dummy player k of Γ, and any weight function $f : \Sigma^+ \to \mathbb{R}_{>0}$ satisfying*

$$f(S \cup \{k\}) = f(S),\ \forall S \in \Sigma^+ : S \not\ni k,$$

each f-least core cost allocation method charges the dummy player k exactly $c(\{k\})$.

Proof. If $|N| = 1$, then there is nothing to prove. Assume that $|N| \geq 2$. Let Φ be an arbitrary f-least core cost allocation method and ε be the f-least core radius of Γ. Denote $x := \Phi(\Gamma)$, we have

$$\varepsilon = \min_{S \in \Sigma^+ \backslash \{N\}} \frac{c(S) - x(S)}{f(S)} = \max_{y \in \mathcal{X}(\Gamma)} \min_{S \in \Sigma^+ \backslash \{N\}} \frac{c(S) - y(S)}{f(S)}. \tag{2.25}$$

Since the core of Γ is empty, the number ε is negative. Denote

$$\Omega := \left\{ T \in \Sigma^+ \backslash \{N\} \,\middle|\, \frac{c(T) - x(T)}{f(T)} = \varepsilon \right\}.$$

For each set $S \in \Sigma^+$ which does not contain k, there holds

$$\frac{c(S \cup \{k\}) - x(S \cup \{k\})}{f(S \cup \{k\})} = \frac{c(S) + c(\{k\}) - x(S) - x_k}{f(S)}. \tag{2.26}$$

We want to prove $x_k = c(\{k\})$ by contradiction. Suppose that $x_k \neq c(\{k\})$. If $x_k < c(\{k\})$, due to (2.26) there holds

$$\frac{c(S \cup \{k\}) - x(S \cup \{k\})}{f(S \cup \{k\})} > \frac{c(S) - x(S)}{f(S)},\ \forall S \in \Sigma^+ : S \not\ni k$$

and

$$\frac{c(\{k\}) - x_k}{f(\{k\})} > 0 > \varepsilon.$$

Therefore, any set T in Ω does not contain k. Denote

$$\nu := \min_{\substack{S \in \Sigma^+ \setminus \{N\} \\ S \ni k}} \frac{c(S) - x(S)}{f(S)}.$$

We have then $\nu > \varepsilon$. Denote

$$\delta := \frac{1}{2}(\nu - \varepsilon) \min_{\substack{S \in \Sigma^+ \setminus \{N\} \\ S \ni k}} f(S) > 0$$

and

$$x_i^\delta := \begin{cases} x_i + \delta & \text{if } i = k \\ x_i - \frac{\delta}{|N|-1} & \text{otherwise,} \end{cases} \quad \forall i \in N.$$

Clearly, $x^\delta(N) = x(N) = c(N)$ and there hold

$$\frac{c(S) - x^\delta(S)}{f(S)} > \frac{c(S) - x(S)}{f(S)} \geq \varepsilon, \ \forall S \in \Sigma^+ \setminus \{N\} : S \not\ni k.$$

On the other hand, for each set $S \in \Sigma^+ \setminus \{N\}$ containing k, we have

$$\begin{aligned} \frac{c(S) - x^\delta(S)}{f(S)} &\geq \frac{c(S) - x(S) - \delta}{f(S)} \\ &\geq \frac{c(S) - x(S)}{f(S)} - \frac{1}{2}(\nu - \varepsilon) \\ &\geq \nu - \frac{1}{2}(\nu - \varepsilon) \\ &> \varepsilon. \end{aligned}$$

Therefore, it holds

$$\min_{S \in \Sigma^+ \setminus \{N\}} \frac{c(S) - x^\delta(S)}{f(S)} > \varepsilon$$

contradicting (2.25). Hence $x_k > c(\{k\})$ and due to (2.26) there holds

$$\frac{c(S \cup \{k\}) - x(S \cup \{k\})}{f(S \cup \{k\})} < \frac{c(S) - x(S)}{f(S)}, \ \forall S \in \Sigma^+ : S \not\ni k.$$

Therefore, any set T in Ω contains k. Denote

$$\mu := \min_{\substack{S \in \Sigma^+ \setminus \{N\} \\ S \not\ni k}} \frac{c(S) - x(S)}{f(S)}.$$

We have then $\mu > \varepsilon$. Denote

$$\theta := \frac{1}{2}(\mu - \varepsilon) \min_{\substack{S \in \Sigma^+ \setminus \{N\} \\ S \not\ni k}} f(S) > 0$$

and

$$x_i^\theta := \begin{cases} x_i - \theta & \text{if } i = k \\ x_i + \frac{\theta}{|N|-1} & \text{otherwise,} \end{cases} \quad \forall i \in N.$$

Similarly as above, one can prove that $x^\theta(N) = c(N)$ and

$$\min_{S \in \Sigma^+ \setminus \{N\}} \frac{c(S) - x^\theta(S)}{f(S)} > \varepsilon$$

contradicting (2.25). □

Similarly, one can prove the following result.

Proposition 2.3.20. *For any given cost allocation game $\Gamma = (N, c, \mathbb{R}^N_+, \Sigma)$ which has a non-empty imputation set $\mathcal{X}(\Gamma)$ and an empty core, any dummy player k of Γ, and any weight function $f : \Sigma^+ \to \mathbb{R}_{>0}$ satisfying*

$$f(S \cup \{k\}) = f(S), \ \forall S \in \Sigma^+ : S \not\ni k,$$

each f-least core cost allocation method charges the dummy player k exactly $c(\{k\})$.

Since the weight function $f = 1$ satisfies the conditions of Proposition 2.3.19 and Proposition 2.3.20, from these propositions and Proposition 2.3.18 we have the following results.

Corollary 2.3.21. *Each least core cost allocation method charges the dummy players of any bounded general cost allocation game $(N, c, \mathbb{R}^N, \Sigma)$ exactly their individual costs.*

Corollary 2.3.22. *Each least core cost allocation method charges the dummy players of any cost allocation game $(N, c, \mathbb{R}^N_+, \Sigma)$ exactly their individual costs.*

Unfortunately, if the weight function f does not fulfil the assumption of Proposition 2.3.19, then the dummy-property does not hold in general.

Proposition 2.3.23. *For any given bounded general cost allocation game $\Gamma = (N, c, \mathbb{R}^N, \Sigma)$ which has a non-empty imputation set $\mathcal{X}(\Gamma)$ and an empty core, any dummy player k of Γ, and any weight function $f : \Sigma^+ \to \mathbb{R}_{>0}$ satisfying*

$$f(S \cup \{k\}) > f(S), \ \forall S \in \Sigma^+ : S \not\ni k,$$

each f-least core cost allocation method charges the dummy player k more than $c(\{k\})$.

Proof. Let Φ be an arbitrary f-least core cost allocation method and ε be the f-least core radius of Γ. Denote $x := \Phi(\Gamma)$, we have

$$\varepsilon = \min_{S \in \Sigma^+ \backslash \{N\}} \frac{c(S) - x(S)}{f(S)} = \max_{y \in \mathcal{X}(\Gamma)} \min_{S \in \Sigma^+ \backslash \{N\}} \frac{c(S) - y(S)}{f(S)}. \tag{2.27}$$

Since the core of Γ is empty, the number ε is negative. We want to prove $x_k > c(\{k\})$ by contradiction. Suppose that $x_k \le c(\{k\})$. Denote

$$\Omega := \left\{ T \in \Sigma^+ \backslash \{N\} \ \middle| \ \frac{c(T) - x(T)}{f(T)} = \varepsilon \right\}.$$

We want to prove that any set T in Ω does not contain k. Suppose that it were not the case and there exists a set $S \in \Omega$ which contains k. Since

$$\frac{c(\{k\}) - x_k}{f(\{k\})} \ge 0 > \varepsilon,$$

the set $S \backslash \{k\}$ is non-empty and hence belongs to Σ^+. Since

$$c(S) - x(S) = \varepsilon f(S) < 0$$

and

$$f(S) > f(S \backslash \{k\}) > 0$$

there holds

$$\begin{aligned}
\varepsilon &= \frac{c(S) - x(S)}{f(S)} \\
&> \frac{c(S) - x(S)}{f(S \backslash \{k\})} \\
&= \frac{c(S \backslash \{k\}) + c(\{k\}) - x(S \backslash \{k\}) - x_k}{f(S \backslash \{k\})} \\
&\ge \frac{c(S \backslash \{k\}) - x(S \backslash \{k\})}{f(S \backslash \{k\})} \\
&\ge \varepsilon,
\end{aligned}$$

contradiction. Hence, any set T in Ω does not contain k. Due to the proof of Proposition 2.3.19, one can construct a vector $x^\delta \in \mathcal{X}(\Gamma)$ satisfying

$$\min_{S \in \Sigma^+ \setminus \{N\}} \frac{c(S) - x^\delta(S)}{f(S)} > \varepsilon,$$

which contradicts (2.27). □

For any f-least core cost allocation method which does not have the dummy-property we still can apply the dummy operator $\mathcal{D}$. One can easily prove that if a general cost allocation game Γ is bounded and satisfies

$$\{x \in \mathcal{X}(\Gamma) \,|\, x_i = c(\{i\}),\ \forall i \in d(\Gamma)\} \neq \emptyset,$$

then its dummy-free subgame Γ_D is also bounded and $\mathcal{X}(\Gamma_D)$ is non-empty. With other words, each f-least core cost allocation method is defined for both Γ and Γ_D in that case. Because of Proposition 2.2.7, since each f-least core cost allocation method Φ is a valid, efficient core cost allocation method, so is its dummy-friendly version $\mathcal{D} \circ \Phi$, which charges every dummy player exactly its individual cost.

Summary. Each f-least core cost allocation method has the following properties:

- Validity
- Efficiency
- Core cost allocation method (Proposition 2.3.12)
- Scalar multiplicativity for bounded general cost allocation games whose f-least core contains a unique vector (Proposition 2.3.15)
- Stability (Proposition 2.3.16)
- Symmetry for bounded general cost allocation games whose f-least core contains a unique vector (Proposition 2.3.17)
- Dummy player property for some general cost allocation games (Propositions 2.3.18–2.3.20).

2.3.2 The f-Nucleolus

Vectors in the f-least core maximize the minimal f-excess (minimal weighted gain) among all possible coalitions. We can also go further and find a price that maximizes the f-excesses of all possible coalitions with respect to the lexicographic ordering. Such a price keeps the benefit of each coalition as high as possible (or keeps the loss of each coalition as low as possible) and reflects the position and the cost of each coalition. Keeping in mind that a coalition which has many good alternatives to form cooperations with other players has a better strategic position than another one which does not have such an advantage. That is the idea behind the f-nucleolus.

In the following, we define the f-nucleolus formally. It is well-defined for general cost allocation games with the following property.

Definition 2.3.24. *A general cost allocation game* $\Gamma = (N, c, P, \Sigma)$ *is called* strongly bounded *if for each family* $\Omega \subsetneq \Sigma \backslash \{\emptyset, N\}$ *and each finite vector* $\alpha \in \mathbb{R}^{\Omega}$ *which satisfy that the set*

$$\mathcal{X}(\Gamma, \Omega, \alpha) := \{x \in \mathcal{X}(\Gamma) \,|\, x(S) = \alpha_S,\ \forall S \in \Omega\}$$

is non-empty, there exists a finite number M *such that there holds*

$$\forall x \in \mathcal{X}(\Gamma, \Omega, \alpha),\ \exists S \in \Sigma^{+} \backslash (\Omega \cup \{N\}) : x(S) \geq M.$$

A strongly bounded cost allocation game is bounded as well.

One can easily prove the following result.

Proposition 2.3.25. *Each general cost allocation game* $\Gamma = (N, c, P, \Sigma)$ *which satisfies at least one of the following two conditions*
(i) x *is bounded below for all* $x \in \mathcal{X}(\Gamma)$,

$$\exists m \in \mathbb{R} : x_i \geq m,\ \forall i \in N, \forall x \in \mathcal{X}(\Gamma),$$

(ii) *for each* $S_0 \in \Sigma^{+} \backslash \{N\}$ *there exist* $S_1, S_2, \ldots, S_{k(S_0)} \in \Sigma^{+} \backslash \{N\}$ *and positive real numbers* $\lambda_0, \lambda_1, \ldots, \lambda_{k(S_0)}$ *such that*

$$\sum_{i=0}^{k(S_0)} \lambda_i \chi_{S_i} = \chi_N$$

is strongly bounded.

Remark 2.3.26. *Each cost allocation game satisfies the condition* (i) *in Proposition 2.3.25 with $m = 0$. The condition* (ii) *is fulfilled if, e.g., for each non-empty set $S \subsetneq N$ there holds*

$$S \in \Sigma \Leftrightarrow N \backslash S \in \Sigma$$

or Σ contains the coalition of each single player, i.e.,

$$\{i\} \in \Sigma, \ \forall i \in N.$$

Let $\Gamma = (N, c, P, \Sigma)$ be a strongly bounded general cost allocation game and $f : \Sigma^+ \to \mathbb{R}_{>0}$ be a weight function. Due to Proposition 2.3.9, the f-least core of Γ is well-defined and non-empty if the imputation set $\mathcal{X}(\Gamma)$ is non-empty. In that case, the f-least core of Γ may contain, in general, more than one point. However, if the family Σ is *full dimensional*, i.e.,

$$\dim\bigl(\operatorname{span}(\chi_\Sigma)\bigr) = |N|,$$

then uniqueness can be enforced by imposing a lexicographic order. Here, we use the notation χ_Ω to represent the set $\{\chi_S \mid S \in \Omega\}$ for each family Ω of coalitions in N. A general cost allocation game with a full dimensional family of possible coalitions is called *full dimensional*.

For each $x \in \mathcal{X}(\Gamma)$, let $\theta_{f,\Gamma}(x)$ be the vector in $\mathbb{R}^{|\Sigma^+|-1}$ whose components are the f-excesses $e_f(S, x)$ of $S \in \Sigma^+ \backslash \{N\}$ arranged in increasing order, i.e.,

$$\theta^i_{f,\Gamma}(x) \leq \theta^j_{f,\Gamma}(x), \ \ \forall 1 \leq i < j \leq |\Sigma^+| - 1.$$

The vector $\theta_{f,\Gamma}(x)$ is called the *f-excess vector of Γ at x*. The function $\theta_{f,\Gamma}$ is called the *f-excess function of Γ*. For $x, y \in \mathcal{X}(\Gamma)$, $\theta_{f,\Gamma}(x)$ is *lexicographically greater* than $\theta_{f,\Gamma}(y)$, denoted $\theta_{f,\Gamma}(x) \succ \theta_{f,\Gamma}(y)$, if there exists an index i_0 such that

$$\theta^{i_0}_{f,\Gamma}(x) > \theta^{i_0}_{f,\Gamma}(y) \ \text{ and } \ \theta^i_{f,\Gamma}(x) = \theta^i_{f,\Gamma}(y), \ \forall i < i_0.$$

We say x is more acceptable than y.

Definition 2.3.27. *The* f-nucleolus *of a strongly bounded general cost allocation game Γ, denoted by $\mathcal{N}_f(\Gamma)$, is the set of all vectors in $\mathcal{X}(\Gamma)$ that maximizes $\theta_{f,\Gamma}$ in $\mathcal{X}(\Gamma)$ with respect to the lexicographic ordering. The f-nucleolus is called* nucleolus, weak nucleolus, *and* proportional nucleolus *for $f = 1$, $f = |\cdot|$, and $f = c$, respectively.*

Remark 2.3.28. *Our definition of the f-nucleolus here is different than the common one. In the common definition, there is no requirement on the set of prices, the family of possible allocations is the power set of N, and our f-nucleolus is actually "f-prenucleolus" in that sense. The common f-nucleolus is the set of vectors in Π,*

$$\Pi := \{x \in \mathbb{R}^N \mid x(N) = c(N) \;\; \text{and} \;\; x_i \leq c(\{i\}), \forall i \in N\},$$

that maximizes $\theta_{f,\Gamma}$ in Π with respect to the lexicographic ordering. However, in our definition of the cost allocation game, it is allowed to set some requirements on the prices. That means we can add the conditions

$$x_i \leq c(\{i\}), \; \forall i \in N$$

to the requirements on the prices. We thereby unite the f-nucleolus and the f-prenucleolus in the common sense into a single definition.

The following algorithm computes points in the f-least core after each step and terminates with the f-nucleolus. It is a generalization of the algorithm calculating the nucleolus in [45], i.e., the f-nucleolus with $f = 1$ for cost allocation games with 2^N as the family of possible coalitions. Here a different update step is used, which guarantees that the number of linear programs that have to be solved is bounded by the number of players. Theorem 2.3.30 shows that the f-nucleolus is well defined and Algorithm 2.3.29 works correctly. Algorithm 2.3.29 allocates step by step prices to coalitions and terminates when the price of each coalition is uniquely determined. By calculating the f-nucleolus of a strongly bounded general cost allocation game $\Gamma = (N, c, P, \Sigma)$, the algorithm starts with finding the f-least core radius and the f-least core $\mathcal{LC}_f(\Gamma)$ of Γ. Each price vector in the f-least core maximizes the minimal weighted benefit among the possible coalitions. There exists coalitions in $\Sigma^+ \backslash \{N\}$ whose prices do not change by choosing different price vector in $\mathcal{LC}_f(\Gamma)$. That means their prices are already optimal and cannot be improved. Hence these coalitions should accept the prices obtained by an arbitrary price vector in $\mathcal{LC}_f(\Gamma)$. By considering any dual optimal solution and taking the linear combination of all coalitions whose corresponding dual optimal variables are positive, the algorithm can find some of these coalitions. Denote the family of these found coalitions as Ω. Let $p_\Omega \in \mathbb{R}^\Omega$ be the allocated prices of the coalitions in Ω. Since the price of every coalition in Ω is fixed, we only have to consider the coalitions in $\tilde{\Sigma} := \Sigma \backslash \Omega$ and the set of possible prices shrinks from P to $\tilde{P}$,

$$\tilde{P} := \{x \in P \mid x(S) = p_S, \ \forall S \in \Omega\}.$$

In the next step, the algorithm again calculates the f-least core radius and the f-least core of the new strongly bounded general cost allocation game $\tilde{\Gamma} = (N, c|_{\tilde{\Sigma}}, \tilde{P}, \tilde{\Sigma})$. This process will be repeated until the price of each coalition is uniquely determined. The algorithm thereby maximizes gradually the attractiveness of a cooperation for all coalitions by improving their prices.

Algorithm 2.3.29. *Computing the f-nucleolus of a strongly bounded general cost allocation game $\Gamma = (N, c, P, \Sigma)$ with non-empty imputation set $\mathcal{X}(\Gamma)$.*

1. *Set $k := 0$, $\mathcal{A}_1 := \{\chi_N\}$, and $P_1 := \mathcal{X}(\Gamma)$.*

2. *Set $k := k + 1$, define*

$$\mathcal{S}_k := \Sigma^+ \backslash \{S \in \Sigma^+ \mid \chi_S \in \operatorname{span}\mathcal{A}_k\},$$

and solve the linear program

$$\max_{(x,\varepsilon)} \ \varepsilon \tag{2.28}$$

$$s.t. \ \ x(S) + \varepsilon f(S) \leq c(S), \ \forall S \in \mathcal{S}_k \tag{2.29}$$

$$x \in P_k. \tag{2.30}$$

Let (x^k, ε^k) and (λ^k, μ^k) be primal and dual optimal solutions of (2.28), where λ^k corresponds to the constraint (2.29).

3. *Define*

$$\begin{aligned}
\Pi_{k+1} :=& \{S \in \mathcal{S}_k \mid \lambda_S^k > 0\}, \\
\mathcal{B}_{k+1} \subseteq \Pi_{k+1} :& \ \chi_{\mathcal{B}_{k+1}} \ \textit{is a basis of} \operatorname{span}\chi_{\Pi_{k+1}} \\
P_{k+1} :=& \{x \in P_k \mid x(S) = c(S) - \varepsilon^k f(S), \ \forall S \in \mathcal{B}_{k+1}\}, \\
\mathcal{A}_{k+1} :=& \mathcal{A}_k \cup \chi_{\mathcal{B}_{k+1}}.
\end{aligned}$$

4. If $|\mathcal{A}_{k+1}| < \dim(\text{span}\chi_\Sigma)$ *then goto 2, else* P_{k+1} *is the* f*-nucleolus of* Γ *and* x^k *is a point in it.*

We denote in the following the linear program (2.28) for $k = i$ by (2.28_i) and the set of its optimal solutions by O_i.

Theorem 2.3.30. *Given a strongly bounded general cost allocation game* $\Gamma = (N, c, P, \Sigma)$. *If* $\mathcal{X}(\Gamma)$ *is non-empty then the* f*-nucleolus of* Γ *is non-empty. In that case, Algorithm 2.3.29 gives a point in the* f*-least core of* Γ *after each step and terminates after at most* $\dim(\text{span}\chi_\Sigma) - 1$ *steps. Moreover, if the family* Σ *is full dimensional, i.e.,* $\dim(\text{span}\chi_\Sigma) = |N|$, *then the* f*-nucleolus contains a unique point.*

Proof. By induction, we can easily prove that the linear program (2.28_k) has an optimal solution for every $k \geq 1$ satisfying $\mathcal{S}_k \neq \emptyset$, i.e., $|\mathcal{A}_k| < \dim(\text{span}\chi_\Sigma)$. Due to Proposition 2.3.9, it holds for $k = 1$. Assume that the linear program (2.28_k) has an optimal solution for every $k \leq \bar{k}$. If $\mathcal{S}_{\bar{k}+1} = \emptyset$, then there is nothing to prove. We consider the case $\mathcal{S}_{\bar{k}+1} \neq \emptyset$. Let $(y^{\bar{k}}, \varepsilon^{\bar{k}})$ be an optimal solution of ($2.28_{\bar{k}}$), then we have due to the complementary slackness theorem

$$y^{\bar{k}}(S) = c(S) - \varepsilon^{\bar{k}} f(S),\ \forall S \in \mathcal{B}_{\bar{k}+1}.$$

Combining this with $y^{\bar{k}} \in P_{\bar{k}}$ yields $y^{\bar{k}} \in P_{\bar{k}+1}$, i.e., $(y^{\bar{k}}, \varepsilon^{\bar{k}})$ is a feasible solution of ($2.28_{\bar{k}+1}$). Therefore, if we can prove that the objective function is bounded above, then the linear program ($2.28_{\bar{k}+1}$) has an optimal solution. Since Γ is strongly bounded, there exists a finite number M such that there holds

$$\forall x \in P_{\bar{k}+1},\ \exists T \in \mathcal{S}_{\bar{k}+1} : x(T) \geq M.$$

Therefore, for each feasible solution (x, ε) of the linear program ($2.28_{\bar{k}+1}$) there exists a set $T \in \mathcal{S}_{\bar{k}+1}$ satisfying $x(T) \geq M$. From this it follows that

$$\begin{aligned} c(T) &\geq x(T) + \varepsilon f(T) \\ &\geq M + \varepsilon f(T), \end{aligned}$$

i.e.,

$$\varepsilon \leq \frac{c(T)-M}{f(T)} \leq \max_{S\in\mathcal{S}_{\bar{k}+1}} \frac{c(S)-M}{f(S)}.$$

That means the linear program ($2.28_{\bar{k}+1}$) is bounded.

As next we consider $z^1, z^2 \in P_l$ for some integer $l \geq 1$ satisfying $\mathcal{S}_l \neq \emptyset$. We prove that

$$e_f(S, z^1) = e_f(S, z^2),\ \forall S \in \Sigma^+ \backslash \mathcal{S}_l. \tag{2.31}$$

Due to the definition of P_l, there holds

$$z^1(S) = c(S) - \varepsilon^i f(S) = z^2(S),\ \forall S \in \mathcal{B}_{i+1}, \forall 1 \leq i \leq l-1,$$

i.e.,

$$z^1(S) = z^2(S),\ \forall S \in \cup_{i=2}^{l} \mathcal{B}_i = \{T \in \Sigma^+ \backslash \{N\} \,|\, \chi_T \in \mathcal{A}_l\}.$$

From this and since

$$z^1(N) = c(N) = z^2(N)$$

it follows that

$$z^1(S) = z^2(S),\ \forall S \in \{T \in \Sigma^+ \,|\, \chi_T \in \operatorname{span}\mathcal{A}_l\} = \Sigma^+ \backslash \mathcal{S}_l, \tag{2.32}$$

i.e., (2.31) holds.

For each $i \geq 1$ satisfying $\mathcal{S}_i \neq \emptyset$ and each optimal solution (y^i, ε^i) of (2.28_i), due to the complementary slackness theorem, we have

$$y^i(S) = c(S) - \varepsilon^i f(S),\ \forall S \in \mathcal{B}_{i+1}.$$

On the other hand, we have $y^i \in P_i$. Therefore, there holds

$$y^i \in P_{i+1},\ \forall (y^i, \varepsilon^i) \in O_i, \forall i \geq 1 : \mathcal{S}_i \neq \emptyset. \tag{2.33}$$

If $\mathcal{S}_{i+1} \neq \emptyset$, then (y^i, ε^i) is a feasible solution of (2.28_{i+1}). Hence, there holds

$$\varepsilon^i \leq \varepsilon^{i+1},\ \forall i \geq 1 :\ \mathcal{S}_{i+1} \neq \emptyset. \tag{2.34}$$

Let $(y^{i+1}, \varepsilon^{i+1})$ be an arbitrary optimal solution of (2.28_{i+1}). Because of (2.34) and $f(S) > 0$ for every coalition S in Σ^+, we have

$$y^{i+1}(S) + \varepsilon^i f(S) \leq y^{i+1}(S) + \varepsilon^{i+1} f(S) \leq c(S),\ \forall S \in \mathcal{S}_{i+1}. \tag{2.35}$$

On the other hand, since $y^i \in P_{i+1}$ and $y^{i+1} \in P_{i+1}$, there holds due to (2.32)

$$y^{i+1}(S) + \varepsilon^i f(S) = y^i(S) + \varepsilon^i f(S) \leq c(S), \ \forall S \in \mathcal{S}_i \backslash \mathcal{S}_{i+1}. \tag{2.36}$$

From (2.35), (2.36), and $y^{i+1} \in P_{i+1} \subseteq P_i$ it follows that (y^{i+1}, ε^i) is a feasible solution and therefore an optimal solution of (2.28$_i$). By induction we have that for an arbitrary integer $k \geq 1$ satisfying $\mathcal{S}_k \neq \emptyset$ and an arbitrary optimal solution (y^k, ε^k) of (2.28$_k$), (y^k, ε^i) is an optimal solution of (2.28$_i$) for each $1 \leq i \leq k$. Especially, since (x^k, ε^1) is an optimal solution of (2.28$_1$), we have $x^k \in \mathcal{LC}_f(\Gamma)$ for every $k \geq 1$.

We now consider an arbitrary k-th. step of Algorithm 2.3.29 with $|\mathcal{A}_k| < \dim(\mathrm{span}\chi_\Sigma)$, i.e., $\mathcal{S}_k \neq \emptyset$. Let (λ^k, μ^k) be a dual optimal solution of (2.28$_k$). Then we have

$$\sum_{S \in \mathcal{S}_k} f(S) \lambda_S^k = 1 > 0.$$

Therefore, since $f(S) > 0$ for all $S \in \mathcal{S}_k$, the set Π_{k+1} is non-empty. Hence, $\mathcal{B}_{k+1}$ is non-empty and

$$|\mathcal{A}_{k+1}| - |\mathcal{A}_k| \geq 1.$$

On the other hand, we have $\mathcal{A}_1 = \{\chi_N\}$ and $\dim(\mathrm{span}\chi_\Sigma) > 1$. Therefore

$$|\mathcal{A}_{k+1}| \geq k + 1, \ \forall k \geq 1 : |\mathcal{A}_k| < \dim(\mathrm{span}\chi_\Sigma).$$

So there exists $1 \leq l \leq \dim(\mathrm{span}\chi_\Sigma)$ such that

$$|\mathcal{A}_l| \geq \dim(\mathrm{span}\chi_\Sigma). \tag{2.37}$$

Let $k^* + 1 \leq \dim(\mathrm{span}\chi_\Sigma)$ be the smallest number l satisfying (2.37), then $\mathcal{S}_k$ is non-empty for every $k \leq k^*$. The algorithm stops after k^* steps.

It remains to prove that P_{k^*+1} is the f-nucleolus of Γ and x^{k^*} belongs to it. Clearly, x^{k^*} belongs to P_{k^*+1} due to (2.33). Now let y be an arbitrary vector in $P_1 = \mathcal{X}(\Gamma)$. If $y \in P_{k^*+1}$ then there holds

$$x^{k^*}(S) = y(S), \ \forall S \in \Sigma^+ : \chi_S \in \mathcal{A}_{k^*+1}.$$

Since $\mathcal{A}_{k^*+1} \subseteq \chi_\Sigma$, $|\mathcal{A}_{k^*+1}| \geq \dim(\mathrm{span}\chi_\Sigma)$, and $\mathcal{A}_{k^*+1}$ is independent, from this it follows that

$$x^{k^*}(S) = y(S), \ \forall S \in \Sigma^+,$$

i.e.,

$$\theta_{f,\Gamma}(x^{k^*}) = \theta_{f,\Gamma}(y),\ \forall y \in P_{k^*+1}. \tag{2.38}$$

Hence, it remains to consider the case $y \in P_1 \backslash P_{k^*+1}$. Since

$$P_1 \backslash P_{k^*+1} = \cup_{i=1}^{k^*}(P_i \backslash P_{i+1}),$$

there exists $1 \leq l \leq k^*$ satisfying

$$y \in P_l \backslash P_{l+1}. \tag{2.39}$$

Clearly, it holds

$$\min_{S \in \mathcal{S}_l} e_f(S, y) < \varepsilon^l. \tag{2.40}$$

If this were not so, then, since $y \in P_l$, (y, ε^l) is a feasible solution and therefore an optimal solution of (2.28_l). Due to (2.33) there holds

$$y \in P_{l+1},$$

which contradicts (2.39). On the other hand, since $y \in P_l$ and $x^{k^*} \in P_{k^*+1} \subseteq P_l$, we have due to (2.31)

$$e_f(S, x^{k^*}) = e_f(S, y),\ \forall S \in \mathcal{S}_1 \backslash \mathcal{S}_l. \tag{2.41}$$

Moreover, as we showed before in this proof, for each $k \geq 1$ satisfying $\mathcal{S}_k \neq \emptyset$ and each optimal solution (y^k, ε^k) of (2.28_k), (y^k, ε^i) is also an optimal solution of (2.28_i) for every $1 \leq i \leq k$. Therefore, since $(x^{k^*}, \varepsilon^{k^*})$ is an optimal solution of (2.28_{k^*}), (x^{k^*}, ε^l) is an optimal solution of (2.28_l) and we have

$$\min_{S \in \mathcal{S}_l} e_f(S, x^{k^*}) = \varepsilon^l. \tag{2.42}$$

From (2.40), (2.41), and (2.42) it follows that

$$\theta_{f,\Gamma}(x^{k^*}) \succ \theta_{f,\Gamma}(y),\ \forall y \in P_1 \backslash P_{k^*+1}. \tag{2.43}$$

From (2.38) and (2.43), we have that P_{k^*+1} is the f-nucleolus of Γ.

If Σ is full dimensional, then there holds

$$|\mathcal{A}_{k^*+1}| \geq \dim(\text{span}\chi_\Sigma) = |N|.$$

Hence, since $\mathcal{A}_{k^*+1}$ is independent, P_{k^*+1} contains at most a point. On the other hand, since x^{k^*} belongs to P_{k^*+1}, we have $P_{k^*+1} = \{x^{k^*}\}$. That means the f-nucleolus of Γ contains a unique point, namely, x^{k^*}. □

The following algorithm is similar to Algorithm 2.3.29 but with a difference, namely, the set P_k,

$$P_k = \{x \in P_{k-1} \,|\, x(S) = c(S) - \varepsilon^k f(S)\},$$

is replaced by the set Q_k defined by

$$Q_k := \{x \in Q_{k-1} \,|\, x(S) \leq c(S) - \varepsilon^k f(S)\}.$$

It also gives us the f-nucleolus but may be less vulnerable to numerical error than Algorithm 2.3.29.

Algorithm 2.3.31. *Computing the f-nucleolus of a cost allocation game $\Gamma = (N, c, P, \Sigma)$ with non-empty imputation set $\mathcal{X}(\Gamma)$.*

1. *Set $k := 0$, $\mathcal{A}_1 := \{\chi_N\}$, $Q_1 := \mathcal{X}(\Gamma)$, and $\bar{Q}_1 := \mathcal{X}(\Gamma)$.*

2. *Set $k := k + 1$, define*

$$\mathcal{S}_k := \Sigma^+ \backslash \{S \in \Sigma^+ \,|\, \chi_S \in \text{span}\mathcal{A}_k\},$$

and solve the linear program

$$\max_{(x,\varepsilon)} \; \varepsilon \tag{2.44}$$

$$s.t. \;\; x(S) + \varepsilon f(S) \leq c(S), \; \forall S \in \mathcal{S}_k \tag{2.45}$$

$$x \in Q_k. \tag{2.46}$$

Let (x^k, ε^k) and (λ^k, μ^k) be primal and dual optimal solutions of (2.44), where λ^k corresponds to the constraint (2.45).

3. *Define*

$$\begin{aligned}
\Pi_{k+1} &:= \{S \in \mathcal{S}_k \,|\, \lambda^k_S > 0\}, \\
\mathcal{B}_{k+1} \subseteq \Pi_{k+1} &: \chi_{\mathcal{B}_{k+1}} \textit{ is a basis of } \text{span}\chi_{\Pi_{k+1}} \\
Q_{k+1} &:= \{x \in Q_k \,|\, x(S) \leq c(S) - \varepsilon^k f(S), \; \forall S \in \mathcal{B}_{k+1}\}, \\
\bar{Q}_{k+1} &:= \{x \in \bar{Q}_k \,|\, x(S) = c(S) - \varepsilon^k f(S), \; \forall S \in \mathcal{B}_{k+1}\}, \\
\mathcal{A}_{k+1} &:= \mathcal{A}_k \cup \chi_{\mathcal{B}_{k+1}}.
\end{aligned}$$

4. *If* $|\mathcal{A}_{k+1}| < \dim(\text{span}\chi_\Sigma)$ *then goto 2, else* $\bar{Q}_{k+1}$ *is the* f*-nucleolus of* Γ *and* x^k *is a point in it.*

Theorem 2.3.32. *For any cost allocation game* Γ *with non-empty imputation set* $\mathcal{X}(\Gamma)$*, Algorithm 2.3.31 gives a point in the* f*-least core of* Γ *after each step and terminates with the* f*-nucleolus after at most* $\dim(\text{span}\chi_\Sigma)-1$ *steps.*

Proof. Consider the linear program

$$\begin{aligned} & \max_{(x,\varepsilon)} \ \varepsilon \\ s.t. \ & x(S) + \varepsilon f(S) \leq c(S), \ \forall S \in \mathcal{S}_k \\ & x \in \bar{Q}_k. \end{aligned} \tag{2.47}$$

We denote in the following the linear program (2.44) for $k = i$ by (2.44_i) and the linear program (2.47) for $k = i$ by (2.47_i). We want to prove by induction that for every step k satisfying $\mathcal{S}_k \neq \emptyset$, each primal and dual optimal solutions of (2.44_k) are also primal and dual optimal solution of (2.47_k). Then Theorem 2.3.32 follows this and Theorem 2.3.30. With $k = 1$ the above statement is true since $Q_1 = \mathcal{X}(\Gamma) = \bar{Q}_1$. Assume that the statement holds for every $k \leq i$, we want to show that it is true for $k = i+1$ as well. If $\mathcal{S}_{i+1} = \emptyset$, then there is nothing to prove. We consider the case $\mathcal{S}_{i+1} \neq \emptyset$. Clearly, each optimal solution of (2.44_i) is a feasible solution of (2.44_{i+1}). Therefore, (2.44_{i+1}) is feasible. For each feasible solution (x, ε) of (2.44_{i+1}), since f is positive, $x \in Q_{i+1} \subseteq \mathbb{R}^N_+$, and

$$x(S) + \varepsilon f(S) \leq c(S), \ \forall S \in \mathcal{S}_{i+1},$$

we have

$$\varepsilon \leq \frac{c(S)}{f(S)}, \ \forall S \in \mathcal{S}_{i+1},$$

i.e., (2.44_{i+1}) is bounded above. Therefore, (2.44_{i+1}) has optimal solutions. Let $(y^{i+1}, \varepsilon^{i+1})$ be an arbitrary optimal solution of (2.44_{i+1}). Since each optimal solution of (2.44_i) is feasible for (2.44_{i+1}), there holds $\varepsilon^{i+1} \geq \varepsilon^i$. From this it follows that (y^{i+1}, ε^i) is feasible and hence optimal for (2.44_i).

Then, due to the induction hypothesis, (y^{i+1}, ε^i) is an optimal solution of (2.47_i). Hence, we have

$$y^{i+1} \in \bar{Q}_i. \tag{2.48}$$

On the other hand, since (y^{i+1}, ε^i) is an optimal solution of (2.44_i), due to the complementary slackness theorem, there holds

$$y^{i+1}(S) = c(S) - \varepsilon^i f(S),\ \forall S \in \mathcal{B}_{i+1}. \tag{2.49}$$

Combining (2.48) and (2.49) yields $y^{i+1} \in \bar{Q}_{i+1}$, i.e., $(y^{i+1}, \varepsilon^{i+1})$ is feasible for (2.47_{i+1}). On the other hand, since each feasible solution of (2.47_{i+1}) is feasible for (2.44_{i+1}), the optimal value ε^{i+1} of (2.44_{i+1}) is an upper bound of (2.47_{i+1}). Therefore, $(y^{i+1}, \varepsilon^{i+1})$ is optimal for (2.47_{i+1}) with the optimal value ε^{i+1} as well. Due to the duality theorem, from this it follows that the dual problems of (2.44_{i+1}) and (2.47_{i+1}) have the same optimal value, namely ε^{i+1}. Moreover, each dual feasible solution of (2.44_{i+1}) is also dual feasible for (2.47_{i+1}). Combining these results, we have that each dual optimal solutions of (2.44_{i+1}) is a dual optimal solution of (2.47_{i+1}) as well. □

We have directly from the definition of the f-least core and the f-nucleolus the following result.

Proposition 2.3.33. *For each strongly bounded general cost allocation game $\Gamma = (N, c, P, \Sigma)$ with a non-empty imputation set $\mathcal{X}(\Gamma)$ and each weight function $f : \Sigma^+ \to \mathbb{R}_{>0}$, there holds*

$$\mathcal{N}_f(\Gamma) \subseteq \mathcal{LC}_f(\Gamma).$$

Definition 2.3.34. *A cost allocation method defined for every strongly bounded general cost allocation game with a non-empty imputation set whose output belongs to the f-nucleolus for some weight function f is called a f-*nucleolus cost allocation method.

The next part of this section is devoted to the properties of f-nucleolus cost allocation methods. Due to Proposition 2.3.33, each f-nucleolus cost allocation method is a f-least core cost allocation method and therefore it has all properties of a f-least core cost allocation method, e.g., validity, efficiency, and core cost allocation. Moreover, it has the following properties.

Scalar multiplicativity

Proposition 2.3.35. *For each strongly bounded general cost allocation game $\Gamma = (N, c, P, \Sigma)$ with a non-empty imputation set $\mathcal{X}(\Gamma)$, each weight function $f : \Sigma^+ \to \mathbb{R}_{>0}$, and each positive numbers λ and μ, there holds*

$$\mathcal{N}_{\mu f}(\lambda\Gamma) = \lambda\mathcal{N}_f(\Gamma).$$

Proof. Let $\theta_{f,\Gamma}$ be the f-excess function of Γ and $\theta_{f,\lambda\Gamma}$ be the f-excess function of $\lambda\Gamma$. Let x^* be an arbitrary vector in $\mathcal{N}_f(\Gamma)$ and y^* be an arbitrary vector in $\mathcal{N}_f(\lambda\Gamma)$. We have then

$$\theta_{f,\Gamma}(x^*) \succeq \theta_{f,\Gamma}(x),\ \forall x \in \mathcal{X}(\Gamma)$$

and

$$\theta_{f,\lambda\Gamma}(y^*) \succeq \theta_{f,\lambda\Gamma}(y),\ \forall y \in \mathcal{X}(\lambda\Gamma).$$

On the other hand, since $\lambda x^* \in \mathcal{X}(\lambda\Gamma)$ and $\frac{1}{\lambda}y^* \in \mathcal{X}(\Gamma)$, from these it follows that

$$\theta_{f,\Gamma}(x^*) \succeq \theta_{f,\Gamma}\left(\frac{1}{\lambda}y^*\right)$$

and

$$\theta_{f,\lambda\Gamma}(y^*) \succeq \theta_{f,\lambda\Gamma}(\lambda x^*).$$

Moreover, because there holds

$$\theta_{f,\lambda\Gamma}(\lambda x) = \lambda\theta_f(x),\ \forall x \in \mathbb{R}^N,$$

the above two inequalities are equivalent to the following ones

$$\theta_{f,\lambda\Gamma}(\lambda x^*) \succeq \theta_{f,\lambda\Gamma}(y^*)$$

and

$$\theta_{f,\Gamma}\left(\frac{1}{\lambda}y^*\right) \succeq \theta_{f,\Gamma}(x^*).$$

That means, λx^* belongs to $\mathcal{N}_f(\lambda\Gamma)$ and $\frac{1}{\lambda}y^*$ belongs to $\mathcal{N}_f(\Gamma)$. Hence, we have finally

$$\mathcal{N}_f(\lambda\Gamma) = \lambda\mathcal{N}_f(\Gamma).$$

With a similar proof, we have

$$\mathcal{N}_{\mu f}(\Gamma) = \mathcal{N}_f(\Gamma)$$

and therefore

$$\mathcal{N}_{\mu f}(\lambda\Gamma) = \mathcal{N}_f(\lambda\Gamma) = \lambda\mathcal{N}_f(\Gamma).$$

□

Since the f-nucleolus of each full dimensional, strongly bounded general cost allocation game whose imputation set is non-empty contains a unique vector, from the above proposition we have the following result.

Corollary 2.3.36. *For each full dimensional, strongly bounded general cost allocation game Γ with a non-empty imputation set $\mathcal{X}(\Gamma)$, each weight function $f : \Sigma^+ \to \mathbb{R}_{>0}$, and each positive numbers λ, there holds for any f-nucleolus cost allocation method Φ*

$$\Phi(\lambda\Gamma) = \lambda\Phi(\Gamma).$$

Symmetry

Proposition 2.3.37. *For any strongly bounded general cost allocation game $\Gamma = (N, c, P, \Sigma)$ with a non-empty imputation set $\mathcal{X}(\Gamma)$ and two equivalent players i and j and any weight function $f : \Sigma^+ \to \mathbb{R}_{>0}$ satisfying*

$$f(S \cup \{i\}) = f(S \cup \{j\}),\ \forall S \subseteq N\backslash\{i, j\} : S \cup \{i\} \in \Sigma,$$

there holds for each f-nucleolus cost allocation method Φ

$$\Phi(\Gamma)_i = \Phi(\Gamma)_j.$$

Proof. Denote $x = \Phi(\Gamma)$ and define y as follows

$$y_k = \begin{cases} \frac{1}{2}(x_i + x_j) & \text{if } k \in \{i, j\} \\ x_k & \text{otherwise.} \end{cases}$$

We have $x \in \mathcal{N}_f(\Gamma)$. Suppose that $x_i \neq x_j$. We want to show that $y \in \mathcal{X}(\Gamma)$ and

$$\theta_{f,\Gamma}(y) \succ \theta_{f,\Gamma}(x),$$

which contradicts $x \in \mathcal{N}_f(\Gamma)$. Due to the proof of Proposition 2.3.17, we have that y belongs to $\mathcal{X}(\Gamma)$. Without loss of generality, we assume that $x_i > x_j$. For each set $S \subseteq N$ and player $k \in N$, denote $S_k := S \cup \{k\}$. Recalling that the f-excess of each set $S \in \Sigma^+$ at $\xi \in \mathbb{R}^N$ is

$$e_f(S, \xi) = \frac{c(S) - \xi(S)}{f(S)}.$$

Since $x_i > x_j$, we have

$$x_i > y_i = y_j > x_j.$$

From this and the definition of y it follows that

$$e_f(S, y) = e_f(S, x), \ \forall S \in \Sigma^+ : \{i, j\} \subseteq S \text{ or } \{i, j\} \cap S = \emptyset$$

and

$$e_f(S_i, x) < e_f(S_i, y) = e_f(S_j, y) < e_f(S_j, x), \ \forall S \subseteq N \backslash \{i, j\} : S_i \in \Sigma.$$

On the other hand, due to the definition of equivalent players, the set

$$\{S \subseteq N \backslash \{i, j\} \,|\, S_i \in \Sigma\}$$

is non-empty and

$$\forall S \subseteq N \backslash \{i, j\} : S_i \in \Sigma \Longleftrightarrow S_j \in \Sigma.$$

Hence, there holds

$$\theta_{f,\Gamma}(y) \succ \theta_{f,\Gamma}(x),$$

which contradicts $x \in \mathcal{N}_f(\Gamma)$. Therefore, $x_i = x_j$. □

Dummy property

Since each f-nucleolus cost allocation method is a f-least core cost allocation method, there hold the results on dummy player presented in section 2.3.1 for f-nucleolus cost allocation methods as well. We have then each f-nucleolus cost allocation method charges every dummy player of any strongly bounded general cost allocation game with a non-empty core exactly its cost. Each nucleolus cost allocation method charges every dummy player of any strongly bounded general cost allocation game $(N_1, c_1, \mathbb{R}^{N_1}, \Sigma_1)$ and any cost allocation game $(N_2, c_2, \mathbb{R}_+^{N_2}, \Sigma_2)$ with a non-empty imputation set exactly its cost. Unfortunately, this does not hold in general. But for any f-nucleolus cost allocation method which does not have the dummy-property, we still can apply the dummy operator $\mathcal{D}$. One can easily prove that if a general cost allocation game $\Gamma = (N, c, P, \Sigma)$ is strongly bounded and satisfies

$$\{x \in \mathcal{X}(\Gamma) \,|\, x_i = c(\{i\}),\ \forall i \in d(\Gamma)\} \neq \emptyset, \tag{2.50}$$

then its dummy-free subgame Γ_D is also strongly bounded and $\mathcal{X}(\Gamma_D)$ is non-empty. With other words, each f-nucleolus cost allocation method is defined for both Γ and Γ_D in that case. Because of Proposition 2.2.7, since each f-nucleolus cost allocation method Φ is a valid, efficient, and symmetric core cost allocation method, so is its dummy-friendly version $\mathcal{D} \circ \Phi$, which charges every dummy player exactly its cost.

Scalar multiplicativity may not hold in general for each f-nucleolus cost allocation method, since the f-nucleolus may contain more than one vector. However, for full dimensional games we have the following result.

Proposition 2.3.38. *For any full dimensional, strongly bounded general cost allocation game* $\Gamma = (N, c, P, \Sigma)$ *with a non-empty imputation set* $\mathcal{X}(\Gamma)$ *satisfying* (2.50)*, any weight function* $f : \Sigma^+ \to \mathbb{R}_{>0}$*, any positive number* λ*, and any* f*-nucleolus cost allocation method* Φ*, there holds*

$$\mathcal{D} \circ \Phi(\lambda\Gamma) = \lambda\mathcal{D} \circ \Phi(\Gamma).$$

Proof. With Γ is full dimensional, its dummy-free subgame Γ_D is full dimensional as well. Let λ be an arbitrary positive number. Due to Corollary 2.3.36, there holds for any f-nucleolus cost allocation method Φ

$$\Phi(\lambda\Gamma_D) = \lambda\Phi(\Gamma_D).$$

From this and Proposition 2.2.7 it follows that

$$\mathcal{D} \circ \Phi(\lambda\Gamma) = \lambda\mathcal{D} \circ \Phi(\Gamma). \qquad \square$$

Summary. Each f-nucleolus cost allocation method is a f-least core cost allocation method and has the following properties:

- Validity
- Efficiency
- Core cost allocation method (Proposition 2.3.33)

- Scalar multiplicativity for full dimensional, strongly bounded general cost allocation games (Theorem 2.3.30 and Proposition 2.3.35)
- Stability (Proposition 2.3.16)
- Symmetry (Proposition 2.3.37)
- Dummy player property for some general cost allocation games (Propositions 2.3.18–2.3.20).

2.3.3 The (f, r)-Least Core

The f-nucleolus is a nice concept from the cooperative game theory. But in this concept the coalitions of one player have the same priority as the other coalitions. The roll of each individual player is however more important. Imagine you have a choice between earning \$50,000 a year while other people make \$25,000 or earning \$100,000 a year while other people get \$250,000. Prices of goods and services are the same. Which would you prefer? Surprisingly, studies show that the majority of people select the first option. The social critic HL Mencken once quipped that, "a wealthy man is one who earns \$100 a year more than his wife's sister's husband." That means human's objective is quite local. It does not rely on some absolute measure but is relative to what other people have. It is very hard to convince someone that his price is fair, while somebody else has to pay just a fractional of his price for one unit. Game theoretical fairness (or coalitional fairness) means that the price should reflect the position of each coalition and its cost by considering all possible of grouping. While individual fairness tends to equality. Designing a cost allocation method should take into account the both types of fairness, since if an allocation is fair in the sense of the cooperative game theory, but some players are not happy with this solution, then it is hardly realized. The (f, r)-least core is a compromise between coalitional fairness and individual satisfaction. Due to Proposition 2.2.4, equality and core cost allocation cannot hold simultaneously. While the core cost allocation property is indispensable, we try to approximate the equality as well as possible.

Given a cost allocation game $\Gamma = (N, c, P, \Sigma)$ with a non-empty imputation set $\mathcal{X}(\Gamma)$ and a weight function $f : \Sigma^+ \to \mathbb{R}_{>0}$. Let $r \in \mathbb{R}^N_{>0}$ be a vector that satisfies

$$\sum_{i \in N} r_i = c(N).$$

r is called a *reference price-vector* of Γ. For example for the ticket pricing problem we can choose r as the distance-price vector. The distance-price of a passenger is the product of the traveling distance and some base-price for a passenger for a distance unit. The ration $\frac{x_i}{r_i}$ in this case is nothing else than the ratio between the price that the player i asked to pay for a distance unit and the base-price. Each individual player i prefers a small ratio $\frac{x_i}{r_i}$. A price x_i with a big ratio $\frac{x_i}{r_i}$ will be seen as unfair by the player i, since in this case there exists a player j with much smaller ratio $\frac{x_j}{r_j}$. Our goal is to find a price vector in the f-least core of Γ that is as "near" as possible to r. That means from the point of view of the cooperative game theory our price is fair since it belongs to the f-least core and hence the minimal weighted benefit of the coalitions in $\Sigma^+ \backslash \{N\}$ is as large as possible. On the other hand, from the point of view of each individual player, the increment of the price of each player in comparison to its reference price is as small as possible. Define

$$\Lambda := \big\{\{i\} \,|\, i \in N\big\} \cup \{N\}$$

and

$$R:\ \Lambda \to \mathbb{R}_{>0},\ R(N) = c(N) \text{ and } R(\{i\}) = r_i,\ \forall i \in N.$$

R is called a *reference price-function* of Γ. We have then a new cost allocation game $\Delta := (N, R, \mathcal{LC}_f(\Gamma), \Lambda)$. The imputation set $\mathcal{X}(\Delta)$ of Δ is $\mathcal{LC}_f(\Gamma)$ since

$$x(N) = c(N) = R(N),\ \forall x \in \mathcal{LC}_f(\Gamma).$$

For each price vector x and player $i \in R$, the R-excess of the coalition $\{i\}$ at x is

$$e_R(\{i\}, x) = \frac{R(\{i\}) - x_i}{R(\{i\})} = 1 - \frac{x_i}{r_i}.$$

Due to Theorem 2.3.30, the R-nucleolus of Δ is well-defined and contains a unique point. It maximizes $\theta_{R,\Delta}(x)$ in $\mathcal{X}(\Delta)$ with respect to the lexicographic ordering, where $\theta_{R,\Delta}(x)$ is the R-excess vector of Δ at x, i.e., the vector in $\mathbb{R}^N$ whose components are the R-excesses $e_R(\{i\}, x)$, $i \in N$, arranged in increasing order. Let us define $\vartheta_{R,\Delta}(x)$ as the vector in $\mathbb{R}^N$ whose components are the ratios $\frac{x_i}{r_i}$, $i \in N$, arranged in decreasing order. Then, equivalently, the R-nucleolus of Δ minimizes $\vartheta_{R,\Delta}(x)$ in $\mathcal{X}(\Delta)$ with respect to the lexicographic ordering. That means, by using the R-nucleolus of Δ as the price, the ratios $\frac{x_i}{r_i}$, $i \in N$, are kept as small as possible.

Definition 2.3.39. *Given a cost allocation game $\Gamma = (N, c, P, \Sigma)$ with a non-empty imputation set $\mathcal{X}(\Gamma)$, a weight function $f : \Sigma^+ \to \mathbb{R}_{>0}$, and a reference price-vector $r \in \mathbb{R}^N_{>0}$ of Γ. Define*

$$\Lambda := \big\{\{i\} \,|\, i \in N\big\} \cup \{N\}$$

and

$$R : \Lambda \to \mathbb{R}_{>0},\ R(N) = c(N) \text{ and } R(\{i\}) = r_i,\ \forall i \in N.$$

The R-nucleolus of $\Delta = (N, R, \mathcal{LC}_f(\Gamma), \Lambda)$ is called the (f, r)-least core *of Γ, denoted by $\mathcal{LC}_{f,r}(\Gamma)$. Since Δ is full dimensional, due to Theorem 2.3.30, the set $\mathcal{LC}_{f,r}(\Gamma)$ contains a unique point.*

Remark 2.3.40. *The (f, r)-least core concept works also for each bounded general cost allocation game $\Gamma = (N, c, P, \Sigma)$ with $c(N) > 0$. If the cost function is positive in Σ^+, then we can use directly the definition of the (f, r)-least core of cost allocation games. However, without the positivity of the cost function, it is not clear whether a positive reference price vector makes sense. Therefore, for the sake of simplicity, we just consider cost allocation games.*

Definition 2.3.41. *The* (f, r)-least core cost allocation method *is the cost allocation method defined for all cost allocation games whose output is the vector in the (f, r)-least core.*

Beside the coalitional and individual fairness, the (f, r)-least core cost allocation method has some other interesting properties, which will be presented in the following. From the definition we have that the (f, r)-least core is a subset of the f-least core. That means the (f, r)-least core cost allocation method has all properties of a f-least core cost allocation method, e.g., validity, efficiency, and core cost allocation.

Fairness

If the reference price-vector belongs to the f-least core, then the (f, r)-least core is the set of the reference price-vector. In that case, both coalitional goal as well as individual target are reached.

Proposition 2.3.42. *Given a cost allocation game $\Gamma = (N, c, P, \Sigma)$ with a non-empty imputation set $\mathcal{X}(\Gamma)$, a weight function $f : \Sigma^+ \to \mathbb{R}_{>0}$, and a reference price-vector $r \in \mathbb{R}^N_{>0}$ of Γ. If r belongs to the f-least core of Γ, then $\{r\}$ is the (f, r)-least core of Γ.*

Proof. Define

$$R : \Lambda \to \mathbb{R}_{>0},\ R(N) = c(N) \text{ and } R(\{i\}) = r_i,\ \forall i \in N.$$

We have to prove that $\{r\}$ is the R-nucleolus of $\Delta = (N, R, \mathcal{LC}_f(\Gamma), \Lambda)$. Due to the assumption we have $r \in \mathcal{LC}_f(\Gamma)$. Let x be a vector in $\mathcal{LC}_f(\Gamma)\backslash\{r\}$, we have

$$x(N) = c(N) = \sum_{i \in N} r_i.$$

Since $x \neq r$, there exists $j \in N$ such that

$$x_j > r_j.$$

Hence, there holds

$$\min_{i \in N} \frac{r_i - x_i}{r_i} \leq \frac{r_j - x_j}{r_j} < 0 = \min_{i \in N} \frac{r_i - r_i}{r_i}.$$

Therefore, we have

$$\theta_{R,\Delta}(r) \succ \theta_{R,\Delta}(x).$$

That means $\{r\}$ is the R-nucleolus of Δ. $\square$

Scalar multiplicativity

Using Propositions 2.3.14, 2.3.15, and 2.3.35, we can prove the scalar multiplicativity of the (f, r)-least core.

Proposition 2.3.43. *For any cost allocation game $\Gamma = (N, c, P, \Sigma)$ with a non-empty imputation set $\mathcal{X}(\Gamma)$, any weight function $f : \Sigma^+ \to \mathbb{R}_{>0}$, any reference price-vector r of Γ, and any positive numbers λ and μ, there holds*

$$\mathcal{LC}_{\mu f, \lambda r}(\lambda\Gamma) = \lambda\mathcal{LC}_{f,r}(\Gamma).$$

Proof. Since r is a reference price-vector of Γ, λr is a reference price-vector of $\lambda\Gamma$. Define

$$\Lambda := \big\{\{i\} \,|\, i \in N\big\} \cup \{N\}$$

and

$$R : \Lambda \to \mathbb{R}_{>0},\ R(N) = c(N) \text{ and } R(\{i\}) = r_i,\ \forall i \in N.$$

Due to Proposition 2.3.15, we have

$$\mathcal{LC}_{\mu f}(\lambda\Gamma) = \lambda\mathcal{LC}_f(\Gamma).$$

From the above result and the definition of (f, r)-least core, we have

$$\mathcal{LC}_{f,r}(\Gamma) = \mathcal{N}_R\big((N, R, \mathcal{LC}_f(\Gamma), \Lambda)\big) \tag{2.51}$$

and

$$\mathcal{LC}_{\mu f,\lambda r}(\lambda\Gamma) = \mathcal{N}_{\lambda R}\big((N, \lambda R, \mathcal{LC}_{\mu f}(\lambda\Gamma), \Lambda)\big) = \mathcal{N}_{\lambda R}\big((N, \lambda R, \lambda\mathcal{LC}_f(\Gamma), \Lambda)\big), \tag{2.52}$$

From (2.51), (2.52), and Proposition 2.3.35 it follows that

$$\mathcal{LC}_{\mu f,\lambda r}(\lambda\Gamma) = \lambda\mathcal{N}_R\big((N, R, \mathcal{LC}_f(\Gamma), \Lambda)\big) = \lambda\mathcal{LC}_{f,r}(\Gamma).$$

□

Symmetry

Proposition 2.3.44. *For any cost allocation game $\Gamma = (N, c, P, \Sigma)$ with a non-empty imputation set $\mathcal{X}(\Gamma)$ and two equivalent players i and j, any weight function $f : \Sigma^+ \to \mathbb{R}_{>0}$ satisfying*

$$f(S \cup \{i\}) = f(S \cup \{j\}),\ \forall S \subseteq N\backslash\{i, j\} : S \cup \{i\} \in \Sigma,$$

and any reference price-vector r of Γ with $r_i = r_j$, let $\{x\}$ be the (f, r)-least core of Γ, there holds

$$x_i = x_j.$$

Proof. Define

$$\Lambda := \big\{\{i\} \,|\, i \in N\big\} \cup \{N\}$$

and

$$R : \Lambda \to \mathbb{R}_{>0},\ R(N) = c(N) \text{ and } R(\{i\}) = r_i,\ \forall i \in N.$$

Then $\{x\}$ is the R-nucleolus of the cost allocation game $\Delta := (N, R, \mathcal{LC}_f(\Gamma), \Lambda)$. Assume that $x_i \neq x_j$. We want to find $y \in \mathcal{X}(\Delta)$ satisfying

$$\theta_{R,\Delta}(y) \succ \theta_{R,\Delta}(x),$$

which contradicts $x \in \mathcal{N}_R(\Delta)$. Define y as follows

$$y_k = \begin{cases} \frac{1}{2}(x_i + x_j) & \text{if } k \in \{i, j\} \\ x_k & \text{otherwise.} \end{cases}$$

Due to Proposition 2.3.17, we have that

$$y \in \mathcal{LC}_f(\Gamma) = \mathcal{X}(\Delta).$$

Without loss of generality, we assume that $x_i > x_j$. There holds

$$x_i > y_i = y_j > x_j.$$

From this and the definition of y it follows that

$$\frac{r_k - x_k}{r_k} = \frac{r_k - y_k}{r_k},\ \forall k \in N \backslash \{i, j\}$$

and

$$\frac{r_i - x_i}{r_i} < \frac{r_i - y_i}{r_i} = \frac{r_j - y_j}{r_j} < \frac{r_j - x_j}{r_j}.$$

These mean

$$\theta_{R,\Delta}(y) \succ \theta_{R,\Delta}(x),$$

which contradicts $x \in \mathcal{N}_R(\Delta)$. Therefore, $x_i = x_j$. □

Dummy property

Since each (f, r)-least core cost allocation method is a f-least core cost allocation method, there hold the results on dummy player presented in section 2.3.1 for f-nucleolus cost allocation methods as well. We have then each (f, r)-least core cost allocation method charges every dummy player of any cost allocation game with a non-empty core exactly its cost. Each (f, r)-least core cost allocation method with $f = 1$ charges every dummy player of any cost allocation game $(N, c, \mathbb{R}^N_+, \Sigma)$ with a non-empty imputation set exactly its cost. Unfortunately, this does not hold in general. But for any (f, r)-least core cost allocation method which does not have the dummy-property, we still can apply the dummy operator $\mathcal{D}$. Clearly, if a cost allocation game $\Gamma = (N, c, P, \Sigma)$ satisfies

$$\{x \in \mathcal{X}(\Gamma) \,|\, x_i = c(\{i\}),\ \forall i \in d(\Gamma)\} \neq \emptyset,$$

then the imputation set $\mathcal{X}(\Gamma_D)$ of its dummy-free subgame Γ_D is non-empty. With other words, for any weight function f and any reference price-vector r, the (f,r)-least core cost allocation method is defined for both Γ and Γ_D in that case. Because of Proposition 2.2.7, since each (f,r)-least core cost allocation method Φ is a valid, efficient, symmetric, and scalar multiplicative core cost allocation method, so is its dummy-friendly version $\mathcal{D} \circ \Phi$, which charges every dummy player exactly its cost.

Summary. The (f,r)-least core cost allocation method is a f-least core cost allocation method and has the following properties:

- Validity
- Efficiency
- Core cost allocation method (Proposition 2.3.12)
- Scalar multiplicativity (Proposition 2.3.43)
- Stability (Proposition 2.3.16)
- Symmetry (Proposition 2.3.44)
- Dummy player property for some cost allocation games (Propositions 2.3.18–2.3.20)
- If the reference price vector r belongs to the f-least core, then $\{r\}$ is the (f,r)-least core (Proposition 2.3.42).

2.3.4 Choosing the Weight Function

The outputs of the game theoretical concepts presented in the previous sections depend on the weight function. Typical choice for it is either the constant 1, the cardinality function, or the cost function. Here, we consider only cost allocation games, since in practice the cost functions are positive and the prices should be non-negative. Let us consider a simple example to demonstrate how different the results with these weight functions are. From this we may have an idea about the most suitable choice. Let $\Gamma =$

$(N, c, \mathbb{R}_+^N, 2^N)$ be a cost allocation game of two players, $N = \{1, 2\}$, and the cost function is defined by

$$c(\{1\}) = 900, \quad c(\{2\}) = 100, \quad \text{and} \quad c(\{1, 2\}) = 900.$$

A cooperation of the two players will give them a profit of 100. The distribution of this profit among the players varies depending on the weight function. One can easily show that the least core and weak least core of Γ contain exactly one point, namely, $(850, 50)$, while the proportional least core of Γ is $\{(810, 90)\}$. Least core cost allocation methods, i.e., f-least core cost allocation methods with $f = 1$, allocate the total cost to players such that the corresponding smallest profit among the coalitions differing from N and $\emptyset$ is kept as high as possible. A least core cost allocation method treats the coalitions equally and independently on their cardinalities and their costs. While weak least core cost allocation methods distribute the common cost to the players in such a way that the smallest average profit per player of the coalitions differing from N and $\emptyset$ is as high as possible. That means, in comparison to a least core cost allocation method, the profit of each coalition is scaled down by its cardinality. For our example, there are only two coalitions differing from N and $\emptyset$, namely, $\{1\}$ and $\{2\}$. Since their cardinality is 1, the outcomes of least core and weak least core cost allocation methods coincide. With the price vector $(850, 50)$, player 1 saves only 5.56% of its cost, while the saving of player 2 is 50%. This will make player 1 dissatisfied. And since his relative saving is although positive but quite small especially in comparison to the one of player 2, he may refuse to join in the cooperation. The same thing will not happen by using the price vector $(810, 90)$, since each player has the same saving of 10%. It is also large enough to prevent the collapse of the grand coalition N. The proportional cost allocation methods do not consider the profit or the average profit of each coalition differing from N and $\emptyset$ but its relative profit, i.e., the ratio between the profit and the cost of each coalition. Thereby the common cost is allocated proportionally. Because of this reason, the cost function may be the most appropriate weight function.

2.3.5 The Shapley Value

The Shapley value of a general cost allocation game $\Gamma = (N, c, \mathbb{R}^N, 2^N)$ is defined as following

$$\phi_i^{Sh}(\Gamma) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(|N \setminus S| - 1)!}{|N|!} \big(c(S \cup \{i\}) - c(S)\big), \quad i \in N.$$

Here for simplicity we assume that $c(\emptyset) = 0$, otherwise we just eliminate every term with $c(\emptyset)$ in the above form. Directly from the definition we have that the Shapley value is symmetric, additive, and scalar multiplicative, and charges zero-player nothing. It is also well-known that the Shapley value is efficient.

Proposition 2.3.45. *For any general cost allocation game* $\Gamma = (N, c, \mathbb{R}^N, 2^N)$ *the Shapley value is efficient, symmetric, additive, and scalar multiplicative and charges dummy players exactly their costs.*

Proof. We only prove that the Shapley value is efficient and charges dummy players exactly their costs, since the other properties are obvious. Let $\Gamma = (N, c, \mathbb{R}^N, 2^N)$ be an arbitrary general cost allocation game. Denote $n := |N|$ and

$$\alpha_k := \frac{k!\,(n-k-1)!}{n!}.$$

From the definition, we have

$$\begin{aligned}
\sum_{i\in N} \phi_i^{Sh}(\Gamma) &= \sum_{i\in N} \sum_{S\subseteq N\setminus\{i\}} \alpha_{|S|} c(S\cup\{i\}) - \sum_{i\in N} \sum_{\emptyset\neq S\subseteq N\setminus\{i\}} \alpha_{|S|} c(S) \\
&= \sum_{\emptyset\neq S\subseteq N} |S|\alpha_{|S|-1} c(S) - \sum_{\emptyset\neq S\subsetneq N} |N\setminus S|\alpha_{|S|} c(S) \\
&= n\alpha_{n-1} c(N) + \sum_{\emptyset\neq S\subsetneq N} \Big(|S|\alpha_{|S|-1} - |N\setminus S|\alpha_{|S|}\Big) c(S).
\end{aligned}$$

On the other hand, we have

$$n\alpha_{n-1} = n\frac{(n-1)!\,0!}{n!} = 1$$

and

$$\begin{aligned}
|S|\alpha_{|S|-1} - |N\setminus S|\alpha_{|S|} &= |S|\frac{(|S|-1)!\,(n-|S|)!}{n!} \\
&\quad - (n-|S|)\frac{|S|!\,(n-|S|-1)!}{n!} \\
&= 0
\end{aligned}$$

From these it follows that

$$\sum_{i\in N} \phi_i^{Sh}(\Gamma) = c(N),$$

i.e., ϕ^{Sh} is efficient.

Now assume that Γ has a dummy player k, i.e.,

$$c(S \cup \{k\}) = c(S) + c(\{k\}),\ \forall S \subseteq N \backslash \{k\}.$$

We split Γ into two games $\Gamma_1 = (N, c_1, \mathbb{R}^N, 2^N)$ and $\Gamma_2 = (N, c_2, \mathbb{R}^N, 2^N)$ with

$$c_1(S) = \begin{cases} c(S\backslash\{k\}) & \text{if } S \ni k \\ c(S) & \text{otherwise} \end{cases}$$

and

$$c_2(S) = \begin{cases} c(\{k\}) & \text{if } S \ni k \\ 0 & \text{otherwise} \end{cases}$$

We have then $c = c_1 + c_2$. Since the Shapley value is additive, there holds

$$\phi_i^{Sh}(\Gamma) = \phi_i^{Sh}(\Gamma_1) + \phi_i^{Sh}(\Gamma_2),\ \forall i \in N. \tag{2.53}$$

Clearly, we have that

$$\phi_k^{Sh}(\Gamma_1) = 0 \tag{2.54}$$

and

$$\phi_i^{Sh}(\Gamma_2) = 0,\ \forall i \in N\backslash\{k\}. \tag{2.55}$$

Combining (2.53) and (2.55) yields

$$\phi_i^{Sh}(\Gamma) = \phi_i^{Sh}(\Gamma_1),\ \forall i \in N\backslash\{k\}. \tag{2.56}$$

From (2.54) and (2.56) and since the Shapley value is efficient it follows that

$$\begin{aligned} \phi_k^{Sh}(\Gamma) &= c(N) - \sum_{i \in N\backslash\{k\}} \phi_i^{Sh}(\Gamma) \\ &= c(N) - \sum_{i \in N\backslash\{k\}} \phi_i^{Sh}(\Gamma_1) \\ &= c(N) - \sum_{i \in N} \phi_i^{Sh}(\Gamma_1) \\ &= c(N) - c_1(N) \\ &= c(\{k\}). \end{aligned}$$

□

Interestingly, Shapley proved in [54] that there exists an unique cost allocation method having all properties listed in Proposition 2.3.45.

Theorem 2.3.46. (Shapley [54])
For each fixed finite set N there exists an unique cost allocation method that is efficient, additive, and symmetric and charges zero players nothing for all general cost allocation games $(N, c, \mathbb{R}^N, 2^N)$, namely the Shapley value.

However, the Shapley value has a weak point, namely, it may lie outside of the core in the case the core is non-empty.

Proposition 2.3.47. *The Shapley cost allocation method does not have uniformly bounded variation and the Shapley value may lie outside of the core (in the case it is non-empty).*

Proof. We again consider the counter example in the proof of Proposition 2.2.5. For arbitrary $k > 0$ and $0 < \alpha \leq \frac{1}{k}$, let $N = \{1, 2, 3\}$ and the cost functions c and $\tilde{c}$ be defined as follows

S	$c(S)$	$\tilde{c}(S)$
$\emptyset$	0	0
$\{1\}$	2	2
$\{2\}$	$3+k$	$3+k$
$\{3\}$	$4+k$	$4+k$
$\{1,2\}$	$3+k$	$3+(1+\alpha)k$
$\{2,3\}$	$5+2k$	$5+2k$
$\{3,1\}$	$4+k$	$4+(1+\alpha)k$
$\{1,2,3\}$	$6+2k$	$6+(2+\alpha)k$

There holds

$$0 \leq \tilde{c}(S) - c(S) \leq \alpha c(S), \ \forall S \subseteq N.$$

Denote $\Gamma_k := (N, c, \mathbb{R}^3_+, 2^N)$ and $\tilde{\Gamma}_k := (N, \tilde{c}, \mathbb{R}^3_+, 2^N)$. We have

$$\phi^{Sh}(\Gamma_k) = (1, 2+k, 3+k),$$

$$\phi^{Sh}(\tilde{\Gamma}_k) = \left(1 + \frac{2}{3}\alpha k, 2 + \left(1 + \frac{\alpha}{6}\right)k, 3 + \left(1 + \frac{\alpha}{6}\right)k\right),$$

and

$$\mathcal{C}(\tilde{\Gamma}_k) = \{(1 + \alpha k, 2 + k, 3 + k)\}.$$

From these it follows that

$$\phi^{Sh}(\tilde{\Gamma}_k) \notin \mathcal{C}(\tilde{\Gamma}_k)$$

and

$$\phi^{Sh}(\tilde{\Gamma}_k)_1 - \phi^{Sh}(\Gamma_k)_1 = \frac{2k}{3}\alpha\phi^{Sh}(\Gamma_k)_1,$$

i.e., ϕ^{Sh} does not have uniformly bounded variation. □

As we have seen, the Shapley value concept has some nice properties but it does not belong to the core in general. Moreover, most of real world applications has some side constraints on the allocated prices, which can be violated by the Shapley value, since this concept does not take any conditions on the output vector into account. The family of possible coalitions in real applications may also differ from the power set. It is not clear how to generalize the Shapley value in order to cover that case. Therefore, the Shapley value may be not suitable for many real world applications, e.g., our ticket pricing problem.

Besides, since the Shapley cost allocation method is efficient, it is neither coalitionally stable, nor user friendly, nor monotonic due to Propositions 2.2.1, 2.2.2, and 2.2.3.

2.3.6 Another Conflict

Theorem 2.3.46 and Proposition 2.3.47 give us the following result.

Proposition 2.3.48. *There exists no core cost allocation method that is efficient, additive, and symmetric and charges zero players nothing.*

As efficiency and symmetry are indispensable, we can only give up one of the other properties. If it is the core cost allocation property, then we have the Shapley value as the only one method which is efficient, additive, and symmetric and charges zero players nothing for all general cost allocation games $(N, c, \mathbb{R}^N, 2^N)$. If additivity is negligible, then we have the dummy-friendly version of each efficient and symmetric core cost allocation method fulfilling all the other properties. Recalling that each f-nucleolus allocation method and each (f, r)-least core allocation method are efficient, symmetric core cost allocation methods.

2.4 Alternative Ansatz

There are several articles on the cost allocation problem where the authors favour the monotonicity over the efficiency. In order to obtain the monotonicity, they have to sacrifice the efficiency, but they want to cover the common cost as much as possible. A cost allocation method is called α-budget balanced for some $\alpha \leq 1$ if the total price is not larger than the total cost and not smaller than α times the total cost. For certain cost allocation games, one can construct cost allocation methods which are monotonic and α-budget balanced for some $\alpha \leq 1$. For example, the Steiner forests game with $\alpha = 0.5$ [38], the metric facility location game with $\alpha = 1/3$ [49], or the single-source rent-or-buy game with $\alpha = 1/15$ [49], etc. There are also randomized cost allocation methods for these game which are α-budget balanced for some $\alpha \leq 1$ with high probality and monotonic [18, 40]. Immorlica et al. [33] showed that every monotonic cost allocation method for the metric facility location game is at most 1/3-budget balanced. That means the factor 1/3 is the best cover-ratio for the metric facility location game. These cost allocation methods are based on the combinatorial structure of the underlying cost function and there are no other requirements on the prices besides their non-negativity. It is not clear how to construct such methods if some real-world conditions on the prices exist. It is also unclear in the case of cost allocation problems whose cost functions do not have a common combinatorial structure. Apart from that, the low cover ratio of the common cost makes these cost allocation methods hardly applicable in practice. In our opinion, the monotonicity is too strict and only provides some interesting properties for the mechanism design [34].

2.5 f-Least Core Radius, Balanced Game, and Non-emptiness of the Core

In this section, we will give explicit formulations of the f-least core radius of the general cost allocation game and the cost allocation game under certain assumptions. Using these formulations one can directly prove the Bondareva-Shapley theorem on the non-emptiness of the core for the so-called balanced game. For a given finite non-empty set N, a collection $\mathcal{B} \subseteq 2^N \backslash \{\emptyset, N\}$ is called *balanced* if

$$\exists (\lambda_S)_{S\in\mathcal{B}} \in \mathbb{R}^{\mathcal{B}}_{>0} : \sum_{S\in\mathcal{B}} \lambda_S \chi_S = \chi_N. \tag{2.57}$$

A pair $(\mathcal{B}, \lambda)$ of a collection $\mathcal{B}$ of coalitions and a vector $(\lambda_S)_{S\in\mathcal{B}} \in \mathbb{R}^{\mathcal{B}}_{>0}$ is called *balanced* if $\mathcal{B}$ is balanced and λ fulfils (2.57). A general cost allocation game (N, c, P, Σ), where Σ is a partitioning family of N, is called *balanced* if for any balanced pair $(\mathcal{B}, \lambda)$ satisfying $\mathcal{B} \subseteq \Sigma^+ \backslash \{N\}$ we have

$$\sum_{S\in\mathcal{B}} \lambda_S c(S) \geq c(N).$$

Bondareva [3] and Shapley [55] proved the equivalence between the balanceness of a general cost allocation game and the non-emptiness of its core in the case $P = \mathbb{R}^N$ and $\Sigma = 2^N$.

Theorem 2.5.1. *(Bondareva-Shapley)*
A general cost allocation game $(N, c, \mathbb{R}^N, 2^N)$ is balanced if and only if its core is non-empty.

The Bondareva-Shapley theorem follows Theorem 2.5.2 below. Moreover, using it, we can prove the same result for general cost allocation games with $P = \mathbb{R}^N$ and Σ must not be 2^N but some partitioning family of N.

Theorem 2.5.2. *Given a general cost allocation game $\Gamma = (N, c, P, \Sigma)$, where Σ is a partitioning family of N, and a weight function $f : \Sigma^+ \to \mathbb{R}_{>0}$. Assume that the imputation set $\mathcal{X}(\Gamma)$ is non-empty. Let ε^* be the f-least core radius of Γ, then there holds*

$$\varepsilon^* \leq \min_{\substack{\mathcal{B}\subseteq\Sigma^+\backslash\{N\} \\ (\mathcal{B},\lambda) \text{ balanced}}} \frac{\sum_{S\in\mathcal{B}} \lambda_S c(S) - c(N)}{\sum_{S\in\mathcal{B}} \lambda_S f(S)}.$$

If $P = \mathbb{R}^N$, then we have

$$\varepsilon^* = \min_{\substack{\mathcal{B}\subseteq\Sigma^+\backslash\{N\} \\ (\mathcal{B},\lambda) \text{ balanced}}} \frac{\sum_{S\in\mathcal{B}} \lambda_S c(S) - c(N)}{\sum_{S\in\mathcal{B}} \lambda_S f(S)}.$$

Proof. Due to Proposition 2.3.8 and Proposition 2.3.9, the f-least core of Γ is well-defined and non-empty. Hence, there exists an optimal solution of the following linear program

$$\begin{aligned} &\max_{(x,\varepsilon)} \ \varepsilon && (2.58)\\ s.t.\ \ &x(S) + \varepsilon f(S) \leq c(S),\ \forall S \in \Sigma^+ \backslash \{N\}\\ &x(N) = c(N)\\ &x \in P. \end{aligned}$$

Let (x^*, ε^*) be an optimal solution of (2.58). For each balanced pair $(\mathcal{B}, \lambda)$ satisfying $\mathcal{B} \subseteq \Sigma^+ \backslash \{N\}$, there holds

$$\begin{aligned} \frac{\sum_{S\in\mathcal{B}} \lambda_S c(S) - c(N)}{\sum_{S\in\mathcal{B}} \lambda_S f(S)} &\geq \frac{\sum_{S\in\mathcal{B}} \lambda_S \big(x^*(S) + \varepsilon^* f(S)\big) - c(N)}{\sum_{S\in\mathcal{B}} \lambda_S f(S)}\\ &= \frac{x^*(N) + \varepsilon^* \sum_{S\in\mathcal{B}} \lambda_S f(S) - c(N)}{\sum_{S\in\mathcal{B}} \lambda_S f(S)}\\ &= \varepsilon^*, \end{aligned}$$

i.e.,

$$\varepsilon^* \leq \min_{\substack{\mathcal{B}\subseteq\Sigma^+\backslash\{N\}\\ (\mathcal{B},\lambda)\text{ balanced}}} \frac{\sum_{S\in\mathcal{B}} \lambda_S c(S) - c(N)}{\sum_{S\in\mathcal{B}} \lambda_S f(S)}. \tag{2.59}$$

Now let $P = \mathbb{R}^N$. The dual problem of (2.58) with $P = \mathbb{R}^N$ is the following linear program

$$\begin{aligned} &\min_{(\nu,\mu)} \sum_{S\in\Sigma^+\backslash\{N\}} c(S)\nu_S + c(N)\mu && (2.60)\\ s.t.\ \ &\sum_{\substack{S\in\Sigma^+\backslash\{N\}\\ S\ni i}} \nu_S + \mu = 0,\ \forall i \in N\\ &\sum_{S\in\Sigma^+\backslash\{N\}} \nu_S f(S) = 1\\ &\nu_S \geq 0,\ \forall S \in \Sigma^+\backslash\{N\}. \end{aligned}$$

Let (ν^*, μ^*) be an optimal solution of (2.60). Since

$$\sum_{S\in\Sigma^+\backslash\{N\}} \nu_S^* f(S) = 1 > 0$$

and $f(S) > 0$ for each $S \in \Sigma^+$, the set of coalitions S whose corresponding values ν_S^* are positive is non-empty

$$\mathcal{B}^* := \{S \in \Sigma^+ \backslash \{N\} \mid \nu_S^* > 0\} \neq \emptyset.$$

Let T be a set in $\mathcal{B}^*$ and j be an element of T. Since $\nu^* \geq 0$, there holds

$$-\mu^* = \sum_{\substack{S \in \Sigma^+ \backslash \{N\} \\ S \ni j}} \nu_S^* \geq \nu_T > 0.$$

Define $(\lambda_S^*)_{S \in \mathcal{B}^*} := -\frac{1}{\mu^*}(\nu_S^*)_{S \in \mathcal{B}^*}$. The pair $(\mathcal{B}^*, \lambda^*)$ is then balanced. Moreover, due to the complementary slackness theorem, we have

$$x^*(S) + \varepsilon^* f(S) = c(S),\ \forall S \in \mathcal{B}^*.$$

Therefore, it holds

$$\begin{aligned} \frac{\sum_{S \in \mathcal{B}^*} \lambda_S^* c(S) - c(N)}{\sum_{S \in \mathcal{B}^*} \lambda_S^* f(S)} &= \frac{\sum_{S \in \mathcal{B}^*} \lambda_S^* \big(x^*(S) + \varepsilon^* f(S)\big) - c(N)}{\sum_{S \in \mathcal{B}^*} \lambda_S^* f(S)} \\ &= \frac{x^*(N) + \varepsilon^* \sum_{S \in \mathcal{B}} \lambda_S^* f(S) - c(N)}{\sum_{S \in \mathcal{B}} \lambda_S^* f(S)} \\ &= \varepsilon^*. \end{aligned} \tag{2.61}$$

Combining (2.59) and (2.61), we have

$$\varepsilon^* = \min_{\substack{\mathcal{B} \subseteq \Sigma^+ \backslash \{N\} \\ (\mathcal{B}, \lambda) \text{ balanced}}} \frac{\sum_{S \in \mathcal{B}} \lambda_S c(S) - c(N)}{\sum_{S \in \mathcal{B}} \lambda_S f(S)}.$$

□

Theorem 2.5.2 provides also a heuristic to calculate a good upper bound of the f-least core radius. For details see Section 4.2.3. It gives us also the Bondareva-Shapley theorem, which is a special case of the following one.

Theorem 2.5.3. *For each general cost allocation game $\Gamma = (N, c, \mathbb{R}^N, \Sigma)$, where Σ is a partitioning family of N, Γ is balanced if and only if its core is non-empty.*

Proof. On one hand, the core of Γ is non-empty if and only if its f-least core radius is non-negative. On the other hand, due to Theorem 2.5.2, this is equivalent to

$$\sum_{S \in \mathcal{B}} \lambda_S c(S) \geq c(N),\ \forall (\mathcal{B}, \lambda) \text{ balanced}, \mathcal{B} \subseteq \Sigma^+ \backslash \{N\},$$

i.e., Γ is balanced. □

Due to Theorem 2.5.2, we have that for each general cost allocation game $\Gamma = (N, c, P, \Sigma)$ the value ε_U defined by

$$\varepsilon_U = \min_{\substack{\mathcal{B} \subseteq \Sigma^+ \setminus \{N\} \\ (\mathcal{B},\lambda) \text{ balanced}}} \frac{\sum_{S \in \mathcal{B}} \lambda_S c(S) - c(N)}{\sum_{S \in \mathcal{B}} \lambda_S f(S)}$$

is an upper bound of the f-least core radius ε^* of Γ. In the case $P = \mathbb{R}^N$, we have then $\varepsilon^* = \varepsilon_U$. It gives us the necessary and sufficient conditions for that the equality holds.

Corollary 2.5.4. *Given a general cost allocation game $\Gamma = (N, c, P, \Sigma)$, where Σ is a partitioning family of N, and a weight function $f : \Sigma^+ \to \mathbb{R}_{>0}$. Assume that the imputation set $\mathcal{X}(\Gamma)$ is non-empty. Let ε^* be the f-least core radius of Γ, then there holds*

$$\varepsilon^* \leq \varepsilon_U$$

and the equality holds if and only if the optimal values of the following two linear programs

$$\begin{aligned} &\max_{(x,\varepsilon)} \ \varepsilon \\ s.t.\ \ & x(S) + \varepsilon f(S) \leq c(S),\ \forall S \in \Sigma^+ \setminus \{N\} \\ & x(N) = c(N) \\ & x \in P \end{aligned} \tag{2.62}$$

and

$$\begin{aligned} &\max_{(x,\varepsilon)} \ \varepsilon \\ s.t.\ \ & x(S) + \varepsilon f(S) \leq c(S),\ \forall S \in \Sigma^+ \setminus \{N\} \\ & x(N) = c(N) \end{aligned} \tag{2.63}$$

coincide.

Proof. Due to Theorem 2.5.2, we have that ε_U is the f-least core radius of the general cost allocation game $(c, N, \mathbb{R}^N, \Sigma)$, which is also the optimal value of the linear program (2.63), and there holds

$$\varepsilon^* \leq \varepsilon_U.$$

On the other hand, ε^* is the optimal value of the linear program (2.62). We have then our corollary. $\square$

In the following, using Corollary 2.5.4, we can derive a form of the f-least core radius of the cost allocation game like the one in Theorem 2.5.2 under certain assumptions. We now consider a cost allocation game $(N, c, \mathbb{R}^N_+, 2^N)$. Since the outcome of our cost allocation game is non-negative, we may need some assumptions on the cost function. We will see that the monotonicity of the cost function is necessary. A function $f : \Sigma \to \mathbb{R}$, $\Sigma \subseteq 2^N$, is called *monotonically increasing* if there holds

$$f(S) \leq f(T),\ \forall S, T \in \Sigma : S \subseteq T.$$

To archive our final theorem we have to prove the following lemma.

Lemma 2.5.5. *Given a finite set N and monotonically increasing functions $c, f : 2^N \backslash \{\emptyset\} \to \mathbb{R}_{>0}$, then for each optimal solution (x, ε) of the linear program*

$$\begin{aligned} &\max_{(x,\varepsilon)} \ \varepsilon \\ s.t.\ &x(S) + \varepsilon f(S) \leq c(S),\ \forall \emptyset \neq S \subsetneq N \\ &x(N) = c(N), \end{aligned} \tag{2.64}$$

x is non-negative.

Proof. Due to Proposition 2.3.9 and since 2^N is a partitioning family of N, the linear program (2.64) has optimal solutions. Let (x, ε) be an optimal solution of (2.64). We now consider two cases.

Case 1 - $\varepsilon \geq 0$: Since $\varepsilon \geq 0$ we have

$$x(N \backslash \{i\}) \leq c(N \backslash \{i\}),\ \forall i \in N.$$

From this and because of the monotonicity of c, it follows that there holds for each $i \in N$

$$\begin{aligned} x_i &= x(N) - x(N \backslash \{i\}) \\ &= c(N) - x(N \backslash \{i\}) \\ &\geq c(N) - c(N \backslash \{i\}) \\ &\geq 0. \end{aligned}$$

Case 2 - $\varepsilon < 0$: Denote

$$\Lambda := \{\emptyset \neq S \subsetneq N \,|\, c(S) - x(S) = \varepsilon f(S)\}.$$

Clearly, since $\varepsilon < 0$, there holds $|N| > 1$. We firstly want to prove that for each $i \in N$ there exists a set $S \in \Lambda$ such that S contains i. Suppose that it were wrong, i.e., there exists $i \in N$ such that i does not belongs to $\cup_{S \in \Lambda} S$. We have then

$$c(S) - x(S) > \varepsilon f(S), \ \forall S \subsetneq N : S \ni i,$$

i.e.,

$$\delta := \min_{i \in S \subsetneq N} \frac{c(S) - x(S)}{f(S)} > \varepsilon.$$

We choose $\nu > 0$ such that

$$\min_{i \in S \subsetneq N} \frac{c(S) - x(S) - \nu}{f(S)} > \varepsilon,$$

and define for each $j \in N$

$$\tilde{x}_j := \begin{cases} x_j + \nu & \text{if } j = i \\ x_j - \frac{\nu}{|N|-1} & \text{otherwise.} \end{cases}$$

Clearly $\tilde{x}(N) = c(N)$ and

$$\tilde{\varepsilon} := \min_{\emptyset \neq S \subsetneq N} \frac{c(S) - \tilde{x}(S)}{f(S)} > \varepsilon,$$

i.e., $(\tilde{x}, \tilde{\varepsilon})$ is a better feasible solution than (x, ε), which contradicts the fact that (x, ε) is an optimal solution of (2.64).

Suppose that there exists $i \in N$ such that $x_i < 0$. From the above result, there exists a set $T \in \Lambda$ that contains i. Since

$$c(\{i\}) - x_i > 0 > \varepsilon f(\{i\}),$$

$\{i\} \notin \Lambda$ and therefore the set $T \backslash \{i\}$ is non-empty. Since $x_i < 0$, $\varepsilon < 0$, and c and f are monotonically increasing, there holds

$$\begin{aligned} c(T\backslash\{i\}) - x(T\backslash\{i\}) - \varepsilon f(T\backslash\{i\}) &\leq c(T) - x(T\backslash\{i\}) - \varepsilon f(T) \\ &< c(T) - x(T) - \varepsilon f(T) \\ &= 0, \end{aligned}$$

i.e., (x, ε) is not feasible for (2.64), which is a contradiction. $\square$

Corollary 2.5.4 and Lemma 2.5.5 give us the following theorem.

Theorem 2.5.6. *Given a cost allocation game* $\Gamma := (N, c, \mathbb{R}^N_+, 2^N)$ *with a monotonically increasing cost function* c *and a monotonically increasing weight function* $f : 2^N \backslash \{\emptyset\} \to \mathbb{R}_{>0}$*. Let* ε^* *be the* f*-least core radius of* Γ*, then there holds*

$$\varepsilon^* = \min_{(\mathcal{B},\lambda)\ balanced} \frac{\sum_{S\in\mathcal{B}} \lambda_S c(S) - c(N)}{\sum_{S\in\mathcal{B}} \lambda_S f(S)}.$$

Remark 2.5.7. *The condition on the monotonicity of* c *in Lemma 2.5.5 and Theorem 2.5.6 is indispensable, since for each cost allocation game* $(N, c, \mathbb{R}^N_+, 2^N)$ *the optimal values of* (2.62) *and* (2.63) *with* $P = \mathbb{R}^N_+$ *and* $\Sigma = 2^N$ *do not coincide in general for cost functions which are not monotonically increasing. Let* $N = \{1, 2\}$ *and* c *be given by*

$$c(\{1\}) = 1, \;\; c(\{2\}) = 5, \;\; c(N) = 2.$$

For $f = 1$ *or* $f = |\cdot|$, (2.62) *has exactly one optimal solution, namely,* $(0, 2)$ *with the optimal value 1, while* (2.63) *has exactly one optimal solution, namely,* $(-1, 3)$ *with the optimal value 2.*

We can also construct a counterexample for the case $f = c$*. We now consider another cost allocation game* $(N, c, \mathbb{R}^N_+, 2^N)$ *with* $N = \{1, 2, 3\}$ *and* c *is defined as follows*

$$c(\{1\}) = 1, \;\; c(\{2\}) = 2, \;\; c(\{3\}) = 2, \;\; c(N) = 1$$
$$c(\{1, 2\}) = 1, \;\; c(\{2, 3\}) = 4, \;\; c(\{3, 1\}) = 1.$$

Let (x^1, ε^1) *be a feasible solution of* (2.62)*, i.e.,*

$$x^1 \in \mathbb{R}^3_+, \; x^1(N) = c(N) = 1, \; \textit{and } x^1(S) + \varepsilon^1 c(S) \leq c(S), \forall \emptyset \neq S \subsetneq N.$$

We have then

$$\begin{aligned} 2\varepsilon^1 &= c(\{1, 2\})\varepsilon^1 + c(\{3, 1\})\varepsilon^1 \\ &\leq c(\{1, 2\}) - x^1(\{1, 2\}) + c(\{3, 1\}) - x^1(\{3, 1\}) \\ &= 2 - x^1(N) - x^1_1 \\ &= 1 - x^1_1 \\ &\leq 1, \end{aligned}$$

i.e., $\varepsilon^1 \leq \frac{1}{2}$. *On the other hand, with* $x^2 := (-\frac{1}{3}, \frac{2}{3}, \frac{2}{3})$, $(x^2, \frac{2}{3})$ *is a feasible solution of* (2.63), *i.e.,*

$$x^2(N) = c(N) = 1 \text{ and } x^2(S) \leq \frac{1}{3}c(S),\ \forall \emptyset \neq S \subsetneq N.$$

Hence, the optimal value of (2.62) *is strictly smaller than the one of* (2.63).

Theorem 2.5.6 gives us the Bondareva-Shapley theorem for cost allocation game.

Theorem 2.5.8. *For a cost allocation game* $\Gamma := (N, c, \mathbb{R}^N_+, 2^N)$ *with a monotonically increasing cost function* c, Γ *is balanced if and only if its core is non-empty.*

Proof. Due to Theorem 2.5.6, the least core radius of Γ is

$$\varepsilon^* = \min_{(\mathcal{B},\lambda) \text{ balanced}} \frac{\sum_{S\in\mathcal{B}} \lambda_S c(S) - c(N)}{\sum_{S\in\mathcal{B}} \lambda_S}.$$

The core of Γ is non-empty if and only if its least core radius ε^* is non-negative, which is equivalent to

$$\sum_{S\in\mathcal{B}} \lambda_S c(S) \geq c(N),\ \forall (\mathcal{B}, \lambda) \text{ balanced},$$

i.e., Γ is balanced. $\square$

Remark 2.5.9. *Again the requirement in Theorem 2.5.8 that* c *is monotonically increasing is inevitably. We consider the cost allocation game* $(N, c, \mathbb{R}^N_+, 2^N)$ *where* $N = \{1, 2, 3\}$ *and* c *is defined by*

$$c(\{1\}) = 2,\ \ c(\{2\}) = 2,\ \ c(\{3\}) = 1,\ \ c(N) = 3$$

$$c(\{1, 2\}) = 4,\ \ c(\{2, 3\}) = 1,\ \ c(\{3, 1\}) = 1.$$

The core of the general cost allocation game $(N, c, \mathbb{R}^N, 2^N)$ *is non-empty, since it contains* $(2, 2, -1)$. *And, therefore, due to the Bondareva-Shapley theorem,* $(N, c, \mathbb{R}^N, 2^N)$ *is balanced. That means* $(N, c, \mathbb{R}^N_+, 2^N)$ *is balanced as well. However, the core of the cost allocation game* $\Gamma := (N, c, \mathbb{R}^N_+, 2^N)$ *is empty. Suppose that it were non-empty, i.e.,*

$$\mathcal{C}(\Gamma) = \{x \in \mathbb{R}^3_+ \mid x(N) = c(N) \text{ and } x(S) \leq c(S),\ \forall \emptyset \neq S \subsetneq N\} \neq \emptyset.$$

Let x be a vector in $\mathcal{C}(\Gamma)$. We have

$$\begin{aligned} x_3 &= x(\{2,3\}) + x(\{3,1\}) - x(N) \\ &= x(\{2,3\}) + x(\{3,1\}) - c(N) \\ &\leq c(\{2,3\}) + c(\{3,1\}) - c(N) \\ &= -1 \\ &< 0, \end{aligned}$$

contradiction.

2.6 Conclusions

In this section, we presented the mathematical definition of cost allocation problem as a cost allocation game. Several cost allocation methods based on game theoretical concepts, namely, the core, the f-least core, the f-nucleolus, the (f, r)-least core, and the Shapley value, were presented. Each of them has some relevant properties for the cost allocation problem. However, there exists no cost allocation method that satisfies all the desired properties. We proved that several properties cannot hold simultaneously, e.g.,

- efficiency and coalitional stability,
- efficiency and user friendly,
- equality and core cost allocation,
- core cost allocation and bounded variation,
- efficiency, symmetry, core cost allocation, additivity, and charging zero-player nothing.

Table 2.4 indicates which properties fail/hold for which cost allocation methods. In our opinion, validity, efficiency, and symmetry are indispensable. The Shapley value may violate the validity property in general, while each f-nucleolus and (f, r)-least core cost allocation method and their dummy-friendly versions have all these properties. The Shapley value is additive, but not the other above-mentioned methods. On the other hand, the Shapley value does not belong to the core in general.

Core cost allocation property is immense important for the stability of the cooperation. It is hardly to convince somebody that a price which does not belong to the core in the case it is non-empty is fair. While keeping validity, efficiency, symmetry, core cost allocation, and the dummy-player property (charging dummy players exactly their costs), we have to sacrifice the remaining properties in general. The dummy-friendly versions of each f-nucleolus and each (f, r)-least core cost allocation method fulfil all these properties. These two concepts differ from each other in one point, namely, the objective of the f-nucleolus is the coalitional fairness, where the output reflects the strategic position of every coalitions, while the (f, r)-least core is a compromise solution taking into account both coalitional and individual fairness. The (f, r)-least core firstly keeps the weighted profit of each coalition as high as possible and secondly approximates the equality as well as possible. Which cost allocation method should be used depends on the goal of each cost allocation problem.

	Non-coop	Prop	MinSub	f-nucleolus	(f,r)-least core	Shapley
Validity	No*	No*	Yes	Yes	Yes	No*
Efficiency	No	Yes	No	Yes	Yes	Yes
Coalitional stability	Yes	No	No*	No	No	No
User friendliness	No	No	Yes	No	No	No
Core cost allocation	No	No	Yes	Yes	Yes	No
Monotonicity	Yes	No	No	No	No	No
Bounded variation	Yes	Yes*	No	No	No	No
Symmetry	Yes**	Yes**	-	Yes	Yes	Yes
Additiviy	Yes	No	No	No	No	Yes
Scalar multiplicativity	Yes	Yes	-	-	Yes	Yes
Dummy player	Yes	No	No	-	-	Yes

Yes* means that the property holds for cost allocation games; Yes** means that the property holds for general cost allocation games whose sets of possible coalitions contain (the coalition of) each single player; while No* means that the property fails in general but holds for some large classes of general cost allocation games that appear in practice (see Subsection 2.2.1).

Table 2.4: Properties of cost allocation methods

Chapter 3

Complexity

This chapter is devoted to the computational complexity of the considered game theoretical concepts. We recall a result in [17] to show that calculating a point in the f-least core is NP-hard even for simple games like the minimum cost spanning tree game, whose cost function is in polynomial time evaluable. Consequently, it is also NP-hard to compute the f-nucleolus and the (f, r)-least core in general. However, if the cost function is submodular, then they can be computed in oracle-polynomial time using the ellipsoid method. This result is based on the well-known theorems of Grötschel, Lovász, and Schrijver on the equivalence between separation and optimization and the oracle-polynomial solvability of submodular function minimization on certain sets [26, 27].

3.1 NP-Hardness of Cost Allocation Game

Computing a vector in the f-least core of a given strongly bounded general cost allocation game is NP-hard in general. Clearly, it is NP-hard if the underlying cost function is given by a NP-hard optimization problem. However, even if the cost function is evaluable in polynomial time, the problem may be still NP-hard. An example is the so called *minimum cost spanning tree game*. A minimum cost spanning tree game is defined by a set N of players, a supply node $s \notin N$, a complete graph with vertex set $V = N \cup \{s\}$, and a non-negative distance function l defined on the edge set of the graph. The set of possible coalitions is 2^N while the set of valid prices is $\mathbb{R}^N$. The cost $c(S)$ of a coalition S in 2^N is the length of a minimum spanning tree in the subgraph induced by $S \cup \{s\}$. Faigle et al. [17] proved that the problem of computing a vector in the f-least core of the minimum cost spanning tree game is NP-hard for a certain class of weight functions including $f = 1$, $f = |\cdot|$, and $f = c$. In that article, the authors reduce the minimum cover problem to the problem of finding a vector in the f-least core of a minimum cost spanning tree game. The minimum cover problem is NP-hard and defined as follows. Let $q \in \mathbb{N}$ and let U be a set of $k \geq q$ elements and W be a set of $3q$ elements. Consider a bipartite graph with the node set $U \cup W$ (partitioned into U and W) such that each node $u \in U$ is adjacent to exactly three nodes in W. We say that the node $u \in U$ *covers* its three neighbors in W. A set $D \subseteq U$ is called a *cover* if each $w \in W$ is incident with some $u \in D$. A *minimum cover* is a cover that minimizes $|D|$. For further details see [17].

3.2 Ellipsoid Method and Submodular Function Minimization

In 1979, Khachiyan indicated how the ellipsoid method can be used in order to check the feasibility of a system of linear inequalities in polynomial time [36], which implies the polynomial time solvability of linear programming problems. This result was sharpened by Grötschel, Lovász, and Schrijver in [26, 27], where they showed the equivalence between separation and optimization for any so called well-described polyhedron. Consequently, the submodular function minimization problem on certain sets can be solved in oracle-polynomial time. In the following, we recall these results since they will be needed in the next section. But let us start with some definitions from [27].

Informally, we imagine an *oracle* as a device that solves certain problems for us. We make no assumptions on how a solution is found. An *oracle algorithm* is an algorithm which can "ask questions" from an oracle, and can use the answers supplied. An oracle algorithm is called *oracle-polynomial* if the number of computational steps, counting each call to the oracle as one step, is polynomial.

Let $P \subseteq \mathbb{R}^n$ be a polyhedron and let φ be a positive integer. We say that P has *facet-complexity at most φ* if there exists a system of inequalities with rational coefficients that has solution set P and such that the encoding length of each inequalities of the system is at most φ. In case $P = \mathbb{R}^n$ we require $\varphi \leq n + 1$. A *well-described polyhedron* is a triple $(P; n, \varphi)$

where $P \subseteq \mathbb{R}^n$ is a polyhedron with facet-complexity at most φ. The *encoding length* $\langle P \rangle$ of a well-described polyhedron $(P; n, \varphi)$ is $\varphi + n$.

For well-described polyhedra, Grötschel, Lovász, and Schrijver showed the following result.

Theorem 3.2.1. ([27], Theorem 6.6.5) *There exist algorithms that, for any well-described polyhedron $(P; n, \varphi)$ specified by a strong separation oracle, and for any given vector $c \in \mathbb{Q}^n$,*
(a) *solve the strong optimization problem* $\max\{c^T x \mid x \in P\}$*, and*
(b) *find an optimum vertex solution of* $\max\{c^T x \mid x \in P\}$ *if one exists.*
The number of calls on separation oracle, and the number of elementary arithmetic operations executed by the algorithms are bounded by a polynomial in φ. All arithmetic operations are performed in numbers whose encoding length is bounded by a polynomial in $\varphi + \langle c \rangle$, where $\langle c \rangle$ denotes the encoding length of c.

Based on the above and several other results, Grötschel, Lovász, and Schrijver showed that the submodular function minimization problem on certain subcollections of a lattice family can be solved in oracle-polynomial time [26, 27]. Let N be a finite set and $\mathcal{C}$ be a *lattice family* on N, i.e., a non-empty collection of subset of N satisfying

$$S, T \in \mathcal{C} \;\Rightarrow\; S \cap T \in \mathcal{C} \text{ and } S \cup T \in \mathcal{C}.$$

Let $\mathcal{S}$ be a subcolletion of $\mathcal{C}$ satisfying the following condition for all $S, T \subseteq N$:

$$\begin{array}{l}\text{If three of the sets } S, T, S \cap T, S \cup T \text{ belong to } \mathcal{C} \backslash \mathcal{S}, \\ \text{then also the fourth belongs to } \mathcal{C} \backslash \mathcal{S}.\end{array} \tag{3.1}$$

Let $F : \mathcal{C} \to \mathbb{Q}$ be a *submodular* function on $\mathcal{C}$, i.e.,

$$F(S \cup T) + F(S \cap T) \leq F(S) + F(T),\ \forall S, T \in \mathcal{C}.$$

We assume that F is given by an oracle and that we know an upper bound ϱ on the encoding lengths of the outputs of this oracle. Moreover, $\mathcal{S}$ is also given by a membership oracle: we can give any set S to the oracle, and it answers whether S belongs to $\mathcal{S}$ or not. Then the following theorem holds.

Theorem 3.2.2. ([27], Theorem 10.4.6) *For any $\mathcal{C}, \mathcal{S}, F, \varrho$ (given as described above), where $\mathcal{S}$ satisfies* (3.1)*, one can find a set $T \in \mathcal{S}$ with*

$$F(T) = \min\{F(S) \mid S \in \mathcal{S}\}$$

in oracle-polynomial time.

Remark 3.2.3. *Let N be a finite set and $\mathcal{C}$ be a lattice family on N satisfying the following condition for all $S, T \subseteq N$:*

$$\text{If three of the sets } S, T, S \cap T, S \cup T \text{ belong to } \mathcal{C}, \text{ then also the fourth belongs to } \mathcal{C}. \tag{3.2}$$

Let $\mathcal{A}$ be a subset of $\chi_{\mathcal{C}}$,

$$\chi_{\mathcal{C}} = \{\chi_S \mid S \in \mathcal{C}\}.$$

Define

$$\mathcal{S} := \mathcal{C} \backslash \{S \in \mathcal{C} \mid \chi_S \in \mathrm{span}(\mathcal{A})\}.$$

For arbitrary sets $S, T \subseteq N$, because of (3.2) *and since*

$$\chi_S + \chi_T = \chi_{S \cap T} + \chi_{S \cup T}$$

and

$$\mathcal{C} \backslash \mathcal{S} = \{S \in \mathcal{C} \mid \chi_S \in \mathrm{span}(\mathcal{A})\},$$

if three of the sets $S, T, S \cap T, S \cup T$ belong to $\mathcal{C} \backslash \mathcal{S}$, then also the fourth belongs to $\mathcal{C} \backslash \mathcal{S}$. Therefore, the subcollection $\mathcal{S}$ satisfies (3.1)*.*

If $\mathcal{C}$ is given by a membership oracle and $|\mathcal{A}| \leq |N|$, then for each set S we can answer whether S belongs to $\mathcal{S}$ or not in oracle-polynomial time, since only polynomial time is needed to check whether χ_S belongs to span($\mathcal{A}$) *or not.*

3.3 Polynomial Time Algorithms for Submodular Games

The problem of calculating a vector in the f-least core, f-nucleolus, or (f, r)-least core is NP-hard in general. However, there exist oracle-polynomial time algorithms for special general cost allocation games. In [16], the authors proved that there exists an oracle-polynomial time algorithm calculating the nucleolus, i.e., f-nucleolus with $f = 1$, of any general cost allocation game of type $(N, c, \mathbb{R}^N, 2^N)$ whose cost function is submodular. It is, however, unclear how to generalize this result for any submodular general cost allocation game using their proof technique. In the following, we present another proof for the general case based on Theorem 3.2.1 and Theorem 3.2.2 where the weight function satisfies a certain property.

3.3.1 Algorithm for the f-Nucleolus

In this section, we investigate the complexity of the f-nucleolus of a strongly bounded general cost allocation game $\Gamma = (N, c, P, \Sigma)$ which has a non-empty imputation set $\mathcal{X}(\Gamma)$ and satisfies the following requirements. We require that the family Σ of possible coalitions is a latice family on N which is given by a membership oracle and satisfies the following condition for all $S, T \subseteq N$:

$$\text{If three of the sets } S, T, S \cap T, S \cup T \text{ belong to } \Sigma, \text{ then also the fourth belongs to } \Sigma. \tag{3.3}$$

The cost function is assumed to be rational-valued and submodular. We consider a special class of the weight function f, namely, $f = \alpha g + \beta c$ with $\alpha, \beta \in \mathbb{Q}_+$, $\alpha + \beta > 0$, and some modular function $g : \Sigma \to \mathbb{Q}$, i.e.,

$$g(S \cup T) + g(S \cap T) = g(S) + g(T),\ \forall S, T \in \Sigma,$$

which satisfies

$$g(S) > 0,\ \forall S \in \Sigma^+.$$

β is equal to 0 if Γ is not a cost allocation game, i.e., if either c is not positive or P does not belong to $\mathbb{R}^N_+$. We can choose, for example, $g = \alpha_1 + \alpha_2 |\cdot|$ with some numbers $\alpha_1, \alpha_2 \in \mathbb{Q}_+$, $\alpha_1 + \alpha_2 > 0$. We also require that the functions c and g are given by two oracles whose outputs' encoding lengths are bounded above by $\langle c \rangle$ and $\langle g \rangle$, respectively. Moreover, we denote $|N|$ by n and assume that $(P; n, \varphi)$ is a well-described polyhedron which is specified by a strong separation oracle. Let $Ax \leq b$ be a system of inequalities with rational coeficients that has solution set P and such that the encoding length of each inequality of the system is at most φ. We will show that there exists an algorithm calculating the f-nucleolus of Γ in oracle-polynomial time. The number of calls on the oracles and the number of elementary arithmetic operations executed by the algorithms are bounded by a polynomial in $\langle N, P, c, f\rangle := n + \varphi + \langle c\rangle + \langle g\rangle + \langle \alpha\rangle + \langle \beta\rangle$, where $\langle\cdot\rangle$ is the function evaluating the encoding length. All arithmetic operations are performed in numbers whose encoding length is bounded by a polynomial in $\langle N, P, c, f\rangle$.

Firstly, we consider the following linear program

$$\begin{aligned} &\max_{(x,\varepsilon)} \ \varepsilon \\ s.t.\ \ & x(S) + \varepsilon f(S) \leq c(S),\ \forall S \in \mathcal{S} \\ & x \in Q, \end{aligned} \tag{3.4}$$

with some non-empty well-described polyhedron $(Q; n, \zeta)$ which is specified by a strong separation oracle and belongs to the imputation set $\mathcal{X}(\Gamma)$, and some collection $\emptyset \neq \mathcal{S} \subseteq \Sigma\backslash\{\emptyset, N\}$ satisfying

Assumption 1.1: For every $S, T \subseteq N$, if three of the sets $S, T, S \cap T, S \cup T$ belong to $\Sigma\backslash\mathcal{S}$, then also the fourth belongs to $\Sigma\backslash\mathcal{S}$

and

Assumption 1.2: There exists a membership oracle to answer whether a set S belongs to $\mathcal{S}$ or not for each set $S \subseteq N$.

The chosen weight function $f : \Sigma \to \mathbb{Q}$ is such that f is positive in Σ^+ and at least one of the following properties holds

Assumption 2.1: The function $c - \varepsilon f$ is submodular for all $\varepsilon \in \mathbb{R}$;

Assumption 2.2: There exists an upper bound $\varepsilon_0 \in \mathbb{Q}$ of (3.4) which can be determined in oracle-polynomial time and has an encoding length bounded by a polynomial in $n + \langle c\rangle + \langle f\rangle$ such that the function $c - \varepsilon f$ is submodular for every $\varepsilon \leq \varepsilon_0$.

The weight function f is given by an oracle and $\langle f\rangle$ is an upper bound on the encoding lengths of the outputs of the f-oracle.

Proposition 3.3.1. *Given $\alpha, \beta \in \mathbb{Q}_+$, $\alpha + \beta > 0$, such that $\beta = 0$ if Γ is not a cost allocation game, and a modular function $g : \Sigma \to \mathbb{Q}$ satisfying*

$$g(S) > 0,\ \forall S \in \Sigma^+,$$

then the function $f = \alpha g + \beta c$ fulfils either Assumption 2.1 or Assumption 2.2.

Proof. If $\beta = 0$, then since

$$\alpha g(S) + \alpha g(T) = \alpha g(S \cap T) + \alpha g(S \cup T),\ \forall S, T \in \Sigma \tag{3.5}$$

and c is submodular, Assumption 2.1 holds.

For $\beta > 0$, we have that Γ is a cost allocation game, i.e., c is positive in Σ^+ and P belongs to $\mathbb{R}^N_+$. The optimal value of (3.4) is the f-least core radius of the cost allocation game $(N, c, \mathcal{S} \cup \{N\}, Q)$ whose imputation set coincides with the set Q and therefore is non-empty. Due to Proposition 2.3.3 and Proposition 2.3.4, the linear program (3.4) has an optimal solution and $\varepsilon_0 := \beta^{-1}$ is an upper bound of its optimal value. Trivially, for all $\varepsilon \leq \varepsilon_0$, since c is submodular, $\varepsilon\beta \leq \varepsilon_0\beta = 1$, and because of (3.5), the function

$$c - \varepsilon f = (1 - \varepsilon\beta)c - \varepsilon\alpha g$$

is submodular as well. This means that Assumption 2.2 holds. $\square$

Using Theorem 3.2.1 and Theorem 3.2.2 we have the following result.

Lemma 3.3.2. *Given c, f, Σ, and $\mathcal{S}$ that fulfil the above assumptions, the separation problem of* (3.4) *can be solved in oracle-polynomial time and one can find an optimal solution of* (3.4) *or assert that the problem is infeasible or unbounded in oracle-polynomial time. The number of calls on oracles and the number of elementary arithmetic operations executed are bounded by a polynomial in $\omega := n + \zeta + \langle c\rangle + \langle f\rangle$. All arithmetic operations are performed in numbers whose encoding length is bounded by a polynomial in ω.*

Proof. We denote

$$Q_{\mathcal{S}} = \{(x, \varepsilon) \,|\, x(S) + \varepsilon f(S) \leq c(S),\ \forall S \in \mathcal{S}\}.$$

Clearly, the polyhedron $Q_{\mathcal{S}}$ has facet complexity at most $n + \langle c\rangle + \langle f\rangle$. Therefore, the polyhedron $\{(x, \varepsilon) \in Q_{\mathcal{S}} \,|\, x \in Q\}$ has facet complexity at most ω. We want to prove that the separation problem of (3.4) can be solved in oracle-polynomial time. Let $(\bar{x}, \bar{\varepsilon}) \in \mathbb{Q}^{n+1}$ be an arbitrary vector. Firstly, we use the separation oracle of Q to check whether x belongs to Q or not. If $x \notin Q$, then we have a valid cut that cuts off the point $(\bar{x}, \bar{\varepsilon})$.

Otherwise, we should check whether this point fulfils the remaining constraints. From the assumptions, either we have a submodular function $F = c - \bar{\varepsilon} f$ or there holds $\bar{\varepsilon} > \varepsilon_0$ for an upper bound ε_0 of (3.4), which can be calculated in oracle-polynomial time. So we only have to consider the first case. Since F is submodular, $F - \bar{x}$ is also submodular. From Theorem 3.2.2 it follows that one can find an optimal solution T of the following optimization problem

$$\min_{S \in \mathcal{S}} c(S) - \bar{\varepsilon} f(S) - \bar{x}(S)$$

in oracle-polynomial time. If the optimal value is non-negative, then $(\bar{x}, \bar{\varepsilon})$ is feasible. Otherwise, $x(T) + \varepsilon f(T) \leq c(T)$ is a valid cut that cuts off the point $(\bar{x}, \bar{\varepsilon})$. Therefore, the separation problem of (3.4) can be solved in oracle-polynomial time. It follows from Theorem 3.2.1 that one can find an optimal solution of (3.4) in oracle-polynomial time or assert that the problem is infeasible or unbounded. Note that each input vector $(\bar{x}, \bar{\varepsilon})$ of the separation oracle in the algorithm is given by solving the non-emptiness problem of a polyhedron in $\mathbb{R}^n$ whose facet complexity is bounded by some polynomial in ω. Hence, the encoding length of $(\bar{x}, \bar{\varepsilon})$ is bounded by a polynomial in ω (see Theorem 6.4.1, Theorem 6.4.9, and Theorem 6.6.5 in [27] for details). Therefore, the number of calls on the oracles and the number of elementary arithmetic operations executed are bounded by a polynomial in ω and all arithmetic operations are performed in numbers whose encoding length is bounded by a polynomial in ω. □

Lemma 3.3.3. *Let $\Gamma = (N, c, P, \Sigma)$ be a strongly bounded general cost allocation game, where c and Σ satisfy the above assumptions and $(P; n, \varphi)$ is a well-defined polyhedron specified by a strong separation oracle. Choose the weight function $f = \alpha g + \beta c$, where $\alpha, \beta \in \mathbb{Q}_+$ satisfy $\alpha + \beta > 0$ and $g : \Sigma \to \mathbb{Q}$ is a modular function whose value is positive in Σ^+. Besides, if Γ is not a cost allocation game, then $\beta = 0$. Given two oracles calculating the values $c(S)$ and $g(S)$ for $S \in \Sigma$, if the imputation set $\mathcal{X}(\Gamma)$ is non-empty, then for each k-th. step of Algorithm 2.3.29 with $\mathcal{S}_k \neq \emptyset$ the separation problem of (2.28_k) can be solved in oracle-polynomial time and one can find an optimal solution of (2.28_k) in oracle-polynomial time* *based on the oracles of c and g, the membership oracle of Σ, and the separation oracle of P. The number of calls on oracles and the number of elementary arithmetic operations executed are bounded by a polynomial in $\nu := n+\varphi+\langle c\rangle+\langle g\rangle+\langle\alpha\rangle+\langle\beta\rangle$. All arithmetic operations are performed in numbers whose encoding lengths are bounded by a polynomial in ν.*

Proof. With $\mathcal{X}(\Gamma) \neq \emptyset$, due to Theorem 2.3.30, the f-nucleolus of Γ is nonempty and the linear program (2.28_k) is neither infeasible nor unbounded. With $\mathcal{S}_k \neq \emptyset$, we have $k < \dim(\text{span}\chi_\Sigma) \leq n$. This lemma is then a direct conclusion of Lemma 3.3.2. We only have to show that the encoding length of the optimal value ε^i of (2.28_i) is bounded by a polynomial in ν for every $i \in [1, k)$ and all the assumptions of Lemma 3.3.2 are fulfilled. Let k^* be the largest number such that $\mathcal{S}_{k^*} \neq \emptyset$, we have $k^* < \dim(\text{span}\chi_\Sigma) \leq n$. We now prove that the encoding length of the optimal value ε^i of (2.28_i) is bounded by a polynomial in ν for every $i \in [1, k^*]$. For each $i \in [1, k^*]$, we can rewrite (2.28_i) as follows

$$
\begin{aligned}
& \max_{(x,\varepsilon)} \ \varepsilon & (3.6)\\
s.t.\ \ & x(S) + \varepsilon f(S) \leq c(S),\ \forall S \in \mathcal{S}_i \\
& Ax \leq b \\
& x(N) \leq c(N) \\
& -x(N) \leq -c(N) \\
& x(S) \leq c(S) - \varepsilon^j f(S),\ \forall S \in \mathcal{B}_{j+1},\ \forall 1 \leq j \leq i-1 \\
& -x(S) \leq \varepsilon^j f(S) - c(S),\ \forall S \in \mathcal{B}_{j+1},\ \forall 1 \leq j \leq i-1.
\end{aligned}
$$

The dual problem of (3.6) is the following linear program

$$
\begin{aligned}
\min \sum_{S \in \mathcal{S}_i} c(S)\lambda_S + b^T\mu + c(N)(\xi^+ - \xi^-) \\
+ \sum_{j=1}^{i-1} \sum_{S \in \mathcal{B}_{j+1}} \big(c(S) - \varepsilon^j f(S)\big)(\rho_S^+ - \rho_S^-) \qquad (3.7)
\end{aligned}
$$

$$
s.t.\ \sum_{\substack{S \in S_i \\ S \ni l}} \lambda_S + \big(A_{\cdot l}\big)^T \mu + \xi^+ - \xi^- + \sum_{j=1}^{i-1} \sum_{\substack{S \in \mathcal{B}_{j+1} \\ S \ni l}} (\rho_S^+ - \rho_S^-) = 0,\ \forall l \in N \quad (3.8)
$$

$$\sum_{S\in S_i} f(S)\lambda_S = 1 \tag{3.9}$$

$$\lambda \geq 0,\ \mu \geq 0,\ \xi^+ \geq 0,\ \xi^- \geq 0,\ \rho^+ \geq 0,\ \rho^- \geq 0. \tag{3.10}$$

The polyhedron Q defined by (3.8)–(3.10) has got vertices. Therefore, there exists an optimal solution $(\tilde{\lambda}, \tilde{\mu}, \tilde{\xi}^+, \tilde{\xi}^-, \tilde{\rho}^+, \tilde{\rho}^-)$ of (3.7), which is a vertex of Q. Since $(\tilde{\lambda}, \tilde{\mu}, \tilde{\xi}^+, \tilde{\xi}^-, \tilde{\rho}^+, \tilde{\rho}^-)$ is a vertex of Q, the number of its non-zero coordinates is not greater than $n+1$ and its encoding length is bounded by a polynomial in ν. Moreover, due to the duality theorem, since ε^i is the optimal value of (3.6) we have

$$\varepsilon^i = \sum_{S\in\mathcal{S}_i} c(S)\tilde{\lambda}_S + b^T\tilde{\mu} + c(N)(\tilde{\xi}^+ - \tilde{\xi}^-) + \sum_{j=1}^{i-1}\sum_{S\in\mathcal{B}_{j+1}} \big(c(S) - \varepsilon^j f(S)\big)(\tilde{\rho}^+_S - \tilde{\rho}^-_S),$$

i.e.,

$$\varepsilon^i + \sum_{j=1}^{i-1}\sum_{S\in\mathcal{B}_{j+1}} f(S)(\tilde{\rho}^+_S - \tilde{\rho}^-_S)\varepsilon^j$$
$$= \sum_{S\in\mathcal{S}_i} c(S)\tilde{\lambda}_S + b^T\tilde{\mu} + c(N)(\tilde{\xi}^+ - \tilde{\xi}^-) + \sum_{j=1}^{i-1}\sum_{S\in\mathcal{B}_{j+1}} c(S)(\tilde{\rho}^+_S - \tilde{\rho}^-_S).$$

It means that $(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_{k^*})$ is the solution of a system of linear equations with k^* variables and k^* equations. The coefficients of this system are rational numbers, whose encoding lengths are bounded by a polynomial in ν. Hence, the encoding length of ε^i is bounded by a polynomial in ν for every $i \in [1, k^*]$.

It remains to show that all the assumptions of Lemma 3.3.2 are fulfilled. Since

$$P_1 = \{x \in P \mid x(N) = c(N)\}$$

and

$$P_k = \{x \in P_1 \mid x(S) = c(S) - \varepsilon^i f(S),\ \forall S \in \mathcal{B}_{i+1}, \forall 1 \leq i < k\},\ \forall k \geq 2,$$

P_k is in oracle-polynomial time separable based on the separation oracle of P, the oracle calculating g, and the cost-function oracle. P_k has facet complexity at most polynomial in ν. f fulfils either Assumption 2.1 or

Assumption 2.2 due to Proposition 3.3.1. Finally, due to the definition of $\mathcal{S}_k$, we have

$$\begin{aligned}\mathcal{S}_k &= \Sigma^+ \backslash \{S \in \Sigma^+ \,|\, \chi_S \in \operatorname{span}\mathcal{A}_k\} \\ &= \Sigma \backslash \{S \in \Sigma \,|\, \chi_S \in \operatorname{span}\mathcal{A}_k\}.\end{aligned}$$

Due to Remark 3.2.3, $\mathcal{S}_k$ fulfils Assumption 1.1 and for each set $S \subseteq N$ we can answer whether S belongs to $\mathcal{S}_k$ or not in oracle-polynomial time based on the membership oracle of Σ. □

For the upcoming results, we strengthen Assumption 2.2 as follows:

Assumption 2.3: There exists a *strict* upper bound $\varepsilon_0 \in \mathbb{Q}$ of (3.4) which can be determined in oracle-polynomial time and has an encoding length bounded by a polynomial in $n + \langle c \rangle + \langle f \rangle$ such that the function $c - \varepsilon f$ is submodular for every $\varepsilon \leq \varepsilon_0$.

Lemma 3.3.4. *For each c and Σ that fulfil the assumptions presented in the beginning of this section, each rational-valued weight function f which fulfils either Assumption 2.1 or Assumption 2.3, and each non-empty collection $\mathcal{S} \subseteq \Sigma \backslash \{\emptyset, N\}$ satisfying Assumption 1.1 and Assumption 1.2, given two oracles calculating the values $c(S)$ and $f(S)$ for $S \in \mathcal{S}$, one can find a basic dual optimal solution of* (3.4) *or assert that the dual problem is infeasible or unbounded in oracle-polynomial time. The number of calls on oracles and the number of elementary arithmetic operations executed are bounded by a polynomial in $\omega := n + \zeta + \langle c \rangle + \langle f \rangle$. All arithmetic operations are performed in numbers whose encoding lengths are bounded by a polynomial in ω.*

Proof. Assume that f satisfies Assumption 2.3. The other case, namely, when f satisfies Assumption 2.1, can be treated similarly. We firstly check whether the primal problem is infeasible or unbounded with the algorithm in Lemma 3.3.2. If one of these cases occurs, because of the duality theorem, we assert that the dual problem is infeasible or unbounded. Otherwise, we solve (3.4) using the ellipsoid method (presented in Theorem 6.4.9 in [27]) with the separation oracle described in the proof of Lemma 3.3.2. There are three types of the output of the separation oracle:

- $c_j x \leq d_j$ (given by the separation oracle of Q).

- $x(S) + \varepsilon f(S) \leq c(S)$ for some $S \in \mathcal{S}$.
- $\varepsilon \leq \varepsilon_0$.

We consider a new linear program which has the same variables and objective function as (3.4) and the constraints that appear in the output of the separation oracle. The problem is as follows:

$$\max_{(x,\varepsilon)} \ \varepsilon \tag{3.11}$$

$$s.t. \ \ x(S) + \varepsilon f(S) \leq c(S), \ \forall S \in \bar{\mathcal{S}} \tag{3.12}$$

$$Cx \leq d \tag{3.13}$$

$$\varepsilon \leq \varepsilon_0, \tag{3.14}$$

with some set $\bar{\mathcal{S}} \subseteq \mathcal{S}$, matrix C, and vector d such that (3.12)–(3.13) represent the constraints of the first two types described above. Since ε_0 is an upper bound of (3.4) and each feasible solution of (3.4) satisfies the constraints (3.12)–(3.13), the optimal value of (3.11) is an upper bound for (3.4). On the other hand, if the ellipsoid method concludes that ε^* is the optimal value of (3.4) using only inequalities (3.12)–(3.14), then it necessarily has to conclude that ε^* is the optimal value of (3.11) as well. Moreover, because of Assumption 2.3, we have that ε_0 is a strict upper bound of (3.4), and therefore of (3.11). Hence, we can remove the inequality (3.14) from the linear program (3.11) without changing the optimal value. So we can rewrite (3.11) as follows

$$\begin{aligned} &\max_{(x,\varepsilon)} \ \varepsilon && (3.15)\\ &s.t. \ \ x(S) + \varepsilon f(S) \leq c(S), \ \forall S \in \bar{\mathcal{S}} &&\\ &Cx \leq d. && \end{aligned}$$

Clearly, each basic optimum solution of the dual problem of (3.15) is a basic optimum solution of the dual problem of (3.4). Using the ellipsoid method, we can solve the dual problem of (3.15) in oracle-polynomial time. One can easily show that the number of calls on oracles and the number of elementary arithmetic operations executed are bounded by a polynomial in ω, and all arithmetic operations are performed in numbers whose encoding length is bounded by a polynomial in ω. □

Using the lemma above, we have the following results.

Lemma 3.3.5. *Let $\Gamma = (N, c, P, \Sigma)$ be a strongly bounded general cost allocation game where c and Σ fulfil the assumptions presented in the beginning of this section and $(P; n, \varphi)$ is a well-defined polyhedron specified by a strong separation oracle. Let f be a modular, rational-valued weight function. Given two oracles calculating the values $c(S)$ and $f(S)$ for $S \in \Sigma$, if the imputation set $\mathcal{X}(\Gamma)$ is non-empty, then for each k-th. step of Algorithm 2.3.29 with $\mathcal{S}_k \neq \emptyset$ one can find a basic dual optimal solution of (2.28_k) in oracle-polynomial time based on the oracles of c and f, the membership oracle of Σ, and the separation oracle of P. The number of calls on oracles and the number of elementary arithmetic operations executed are bounded by a polynomial in $\vartheta := n + \varphi + \langle c \rangle + \langle f \rangle$. All arithmetic operations are performed in numbers whose encoding length is bounded by a polynomial in ϑ.*

Proof. Because $\mathcal{X}(\Gamma) \neq \emptyset$, due to Theorem 2.3.30, the f-nucleolus of Γ is non-empty and the linear program (2.28_k) is neither infeasible nor unbounded. Therefore, its dual problem has an optimal solution. This lemma follows directly from Lemma 3.3.4, since all its assumptions are fulfilled due to the proof of Lemma 3.3.3. □

Lemma 3.3.6. *Let $\Gamma = (N, c, P, \Sigma)$ be a cost allocation game where c and Σ fulfil the assumptions presented in the beginning of this section and $(P; n, \varphi)$ is a well-defined polyhedron specified by a strong separation oracle. Choose the weight function $f = \alpha g + \beta c$ for some numbers $\alpha \in \mathbb{Q}_+$, $\beta \in \mathbb{Q}_{>0}$, and modular function $g : \Sigma \to \mathbb{Q}$ whose value is positive in Σ^+. Two oracles calculating the values $c(S)$ and $g(S)$ for $S \in \Sigma$ are given. For each k-th. step of Algorithm 2.3.29 with $\mathcal{S}_k \neq \emptyset$, if the imputation set $\mathcal{X}(\Gamma)$ is non-empty and the optimal value of (2.28_k) is smaller than β^{-1}, then one can find a basic dual optimal solution of (2.28_k) in oracle-polynomial time based on the oracles of c and g, the membership oracle of Σ, and the separation oracle of P. The number of calls on oracles and the number of elementary arithmetic operations executed are bounded by a polynomial in $\nu := n + \varphi + \langle c \rangle + \langle g \rangle + \langle \alpha \rangle + \langle \beta \rangle$. All arithmetic operations are performed in numbers whose encoding length is bounded by a polynomial in ν.*

Proof. Because $\mathcal{X}(\Gamma) \neq \emptyset$, due to Theorem 2.3.30, the f-nucleolus is non-empty and the linear program (2.28_k) is neither infeasible nor unbounded. Therefore, its dual problem has an optimal solution. This lemma follows directly from Lemma 3.3.4, since all assumptions of Lemma 3.3.4 are fulfilled due to the proof of Lemma 3.3.3. Here, f satisfies Assumption 2.3 with $\varepsilon_0 = \beta^{-1}$. □

Finally, using the above results, we have the following theorem.

Theorem 3.3.7. *Let $\Gamma = (N, c, P, \Sigma)$ be a strongly bounded general cost allocation game where c and Σ fulfil the assumptions presented in the beginning of this section and $(P; n, \varphi)$ is a well-defined polyhedron specified by a strong separation oracle. Choose the weight function $f = \alpha g + \beta c$, where $\alpha, \beta \in \mathbb{Q}_+$ satisfy $\alpha + \beta > 0$ and $g : \Sigma \to \mathbb{Q}$ is a modular function whose value is positive in Σ^+. Besides, if Γ is not a cost allocation game, then $\beta = 0$. Given two oracles calculating the values $c(S)$ and $g(S)$ for $S \in \Sigma$, one can find the f-nucleolus $\mathcal{N}_f(\Gamma)$ of Γ in oracle-polynomial time based on the oracles of c and g, the membership oracle of Σ, and the separation oracle of P. The number of calls on oracles and the number of elementary arithmetic operations executed are bounded by a polynomial in $\nu := n + \varphi + \langle c \rangle + \langle g \rangle + \langle \alpha \rangle + \langle \beta \rangle$. All arithmetic operations are performed in numbers whose encoding length is bounded by a polynomial in ν.*

Proof. We first check whether

$$\mathcal{X}(\Gamma) := \{x \in P \mid x(N) = c(N)\}$$

is empty. This can be done in oracle-polynomial time using the ellipsoid method. If it is the case, then the f-nucleolus $\mathcal{N}_f(\Gamma)$ is the empty set. Otherwise, due to Theorem 2.3.30, the f-nucleolus of Γ is non-empty and can be calculated by Algorithm 2.3.29. It terminates after k^* steps and we obtain $\mathcal{N}_f(\Gamma) = P_{k^*+1}$. We prove that the running time of this algorithm is oracle-polynomial. Since Algorithm 2.3.29 needs at most $n-1$ steps, we only have to show that each step takes oracle-polynomial time.

We firstly consider the case $\beta > 0$. In this case, Γ is a cost allocation game, i.e., c is positive in Σ^+ and P belongs to $\mathbb{R}^N_+$. For each $k \leq$

k^*, we have $|\mathcal{A}_k| < \dim(\mathrm{span}\chi_\Sigma)$ and the set $\mathcal{S}_k$ is non-empty. Due to Lemma 3.3.3, one can find an optimal solution (x^k, ε^k) of (2.28_k) in oracle-polynomial time. Recalling that $\varepsilon^k \leq \beta^{-1}$. We consider two cases: $\varepsilon^k < \beta^{-1}$ and $\varepsilon^k = \beta^{-1}$.

(i) **Case 1.1 -** $\varepsilon^k < \beta^{-1}$: One can find a basic dual optimal solution (λ^k, μ^k) of (2.28_k) in oracle-polynomial time due to Lemma 3.3.6. The number of positive coordinates of λ^k is then polynomial in ν. Therefore, we can calculate Π_{k+1}, $\mathcal{B}_{k+1}$, and $\mathcal{A}_{k+1}$ in polynomial time. The polyhedron P_{k+1} is well-defined and its facet complexity is bounded by a polynomial in ν.

(ii) **Case 1.2 -** $\varepsilon^k = \beta^{-1}$: We have

$$x^k(S) + \beta^{-1}\alpha g(S) + c(S) \leq c(S),\ \forall S \in \mathcal{S}_k.$$

Since $x^k \in P \subseteq \mathbb{R}^N_+$, $g|_{\mathcal{S}_k} > 0$, $\alpha \geq 0$, and $\beta > 0$, it follows that

$$x^k(S) = 0,\ \forall S \in \mathcal{S}_k,$$

i.e.,

$$e_f(S, x^k) = \frac{c(S)}{f(S)},\ \forall S \in \mathcal{S}_k. \tag{3.16}$$

Let x^* be an arbitrary vector in the f-nucleolus of Γ. Then we have

$$x^* \in \mathcal{N}_f(\Gamma) = P_{k^*+1} \subseteq P_k.$$

Since $x^* \in P \subseteq \mathbb{R}^N_+$ and $f|_{\mathcal{S}_k} > 0$, there holds

$$e_f(S, x^*) = \frac{c(S) - x^*(S)}{f(S)} \leq \frac{c(S)}{f(S)},\ \forall S \in \mathcal{S}_k. \tag{3.17}$$

On the other hand, since $x^k, x^* \in P_k$, due to (2.31), we have

$$e_f(S, x^k) = e_f(S, x^*),\ \forall S \in \Sigma^+ \backslash \mathcal{S}_k. \tag{3.18}$$

Since x^* belongs to the f-nucleolus of Γ, from (3.16)–(3.18) it follows that

$$x^*(S) = 0,\ \forall S \in \mathcal{S}_k.$$

Therefore, we have

$$\begin{aligned}
\mathcal{N}_f(\Gamma) &= P_{k^*+1} \\
&= \{x \in P_k \,|\, x(S) = 0,\ \forall S \in \mathcal{S}_k\} \\
&= \{x \in P_k \,|\, x_i = 0,\ \forall i \in \cup_{S \in \mathcal{S}_k} S\}.
\end{aligned}$$

So, we can stop Algorithm 2.3.29 and do not need to calculate a basic dual optimal solution of (2.28_k).

The case $\beta = 0$ can be treated similarly using Lemma 3.3.3 and Lemma 3.3.5. □

3.3.2 Algorithms for the f-Least Core and the (f, r)-Least Core

We consider any bounded general cost allocation game $\Gamma = (N, c, P, \Sigma)$ with a non-empty imputation set $\mathcal{X}(\Gamma)$, where c is rational-valued and submodular, $(P, |N|, \varphi)$ is a well-described polyhedron given by a strong separation oracle, and Σ is a latice family on N given by a membership oracle, which satisfies (3.3). Let f be a rational-valued weight function satisfying either Assumption 2.1 or Assumption 2.2 in the previous subsection with $\mathcal{S} = \Sigma \backslash \{\emptyset, N\}$ and $Q = \mathcal{X}(\Gamma)$. Function f can be chosen as in Proposition 3.3.1. Functions c and f are also given by two oracles. Due to Lemma 3.3.2 with $\mathcal{S} = \Sigma \backslash \{\emptyset, N\}$ and $Q = \mathcal{X}(\Gamma)$, there exists an algorithm calculating the f-least core radius $\varepsilon_f(\Gamma)$ and, therefore, the f-least core $\mathcal{LC}_f(\Gamma)$ of Γ,

$$\mathcal{LC}_f(\Gamma) = \big\{x \in \mathcal{X}(\Gamma) \,|\, x(S) + \varepsilon_f(\Gamma) f(S) \leq c(S), \ \forall S \in \Sigma \backslash \{\emptyset, N\}\big\},$$

in oracle-polynomial time. From this lemma, it also follows that the separation problem of the f-least core can be solved in oracle-polynomial time. Let $r \in \mathbb{Q}^N_{>0}$ be a reference price-vector of Γ. Define

$$\Lambda := \big\{\{i\} \,|\, i \in N\big\} \cup \{N\}$$

and

$$R : \Lambda \to \mathbb{R}_{>0}, \ R(N) = c(N) \text{ and } R(\{i\}) = r_i, \ \forall i \in N.$$

The (f, r)-least core of Γ is the R-nucleolus of $\Delta := (N, R, \mathcal{LC}_f(\Gamma), \Lambda)$. We use Algorithm 2.3.29 to calculate the R-nucleolus of Δ. With a similar proof as the one of Lemma 3.3.3, we can prove that the encoding length of the optimal value of the linear program in each step of Algorithm 2.3.29 for Δ is bounded by a polynomial in the encoding length of the input data. Since $|\Lambda| = |N| + 1$ and the separation problem of the f-least core can be solved in oracle-polynomial time, one can prove that each step of Algorithm 2.3.29 for Δ can be solved in oracle-polynomial time. On the other hand, since Algorithm 2.3.29 needs at most $|N| - 1$ steps, from this it follows that one can calculate the (f, r)-least core of Γ in oracle-polynomial time.

Chapter 4

Computational Aspects

The goal of this chapter is to investigate computational methods for the game theoretical solutions of the cost allocation problem. Finding a point in the f-least core, the f-nucleolus, or the (f, r)-least core is NP-hard in general. The biggest challenge is that there is exponential number of possible coalitions which must be taken into account. A medium-sized cost allocation game with 30 players has $2^{30} - 1$ possible coalitions, i.e., more than one billion. A cost allocation problem of this size is not unusual in practice and we want to solve even larger problems. For example, the number of players in our ticket pricing game is almost three times as large. The difficulty is not only to solve a huge linear program itself but also to calculate its coefficients. In order to write it completely, we have to know the cost of every coalition, which is practically impossible for games having more than 30 players. For many real world applications, the cost function is given by a NP-hard optimization problem, which is not easy to compute for every (large) coalition. Knowing explicitly the linear program will takes us centuries in those cases. Apart from that, such a huge linear program with several billions constraints is unlikely solvable. However, in practice, we may not need to know the complete information. Just a few constraints influence the optimal solutions of the linear program, while the remainders are just redundant. An usual idea to overcome the difficulty is using a constraint generation approach [30, 15, 11]. We start with a simplified version of the original linear program formed by a few of its constraints. The remaining constraints are gradually added to the simplified linear program until its optimal solution is also feasible and therefore optimal for the original problem. With this technique we are able to solve large real world applications.

4.1 Combinatorial Game

In order to apply a constraint generation approach, one requirement must be made, namely, the cost function is given by a minimization problem. This is reasonable since people usually want to pay the minimal cost. A general cost allocation game $\Gamma = (N, c, P, \Sigma)$ whose cost function c is given by a optimization problem of the following form

$$\begin{aligned} \forall S \in \Sigma^+ : \quad c(S) := \min_{\xi} \ & c\xi \\ s.t. \ & B\xi \geq C\chi_S \\ & D\xi \geq d \\ & \xi_j \in Z_j, \ j = 1, 2, \ldots, k, \end{aligned} \tag{4.1}$$

where χ_S is the incidence vector of S and Z_j is the set of either real, or integer, or binary numbers, is called a *combinatorial cost allocation game*. We will see in Section 4.2.4 that for a combinatorial cost allocation game one can formulate the separation problem of the constraint generation approach as a maximization problem, which can be solved without evaluating the cost of every possible coalition.

4.2 Constraints Generation Approaches

In this section, we present constraint generation approaches for calculating the f-least core, the f-nucleolus, and the (f, r)-least core. Two main questions of such constraint generation approaches, namely, how to find a good starting set of constraints and how to solve its separation problem, will be answered. For the sake of simplicity, we only consider cost allocation games. This is reasonable since all applications that we know are cost allocation games. Many, but not all, of the following results still hold for general cost allocation games.

4.2.1 The f-Least Core and the f-Nucleolus

Let $\Gamma = (N, c, P, \Sigma)$ be a cost allocation game whose imputation set $\mathcal{X}(\Gamma)$ is non-empty and $f : \Sigma^+ \to \mathbb{R}_{>0}$ be a weight function. One can calculate the f-nucleolus of Γ using either Algorithm 2.3.29 or Algorithm 2.3.31. Since the first loop of these algorithms give us the f-least core of Γ, we concentrate on the f-nucleolus. We consider only Algorithm 2.3.31 because

Algorithm 2.3.29 can be treated identically. The biggest challenge thereby is the step 2 in every loop of the algorithm, where we have to solve a linear program with exponential many constraints. The other steps are trivial.

Let us consider an arbitrary k-th loop. We have to solve the following linear program

$$\max_{(x,\varepsilon)} \ \varepsilon \tag{4.2}$$
$$s.t. \ \ x(S) + \varepsilon f(S) \leq c(S), \ \forall S \in \mathcal{S}_k \tag{4.3}$$
$$A^k x \leq b^k \tag{4.4}$$

and its dual program. For the sake of simplicity, assume that the polyhedron P is given by $Ax \leq b$, whose number of constraints is polynomial in the number of players, i.e., the exponential number of constraints of (4.2) comes only from (4.3). This assumption holds for every application that we have found in the literature and practice. If it is not the case, then the constraint (4.4) cannot be treated explicitly, but only via constraint generation.

The following algorithm calculates primal and dual optimal solutions of (4.2).

Algorithm 4.2.1. *Finding primal and dual optimal solutions of* (4.2).
Given a non-empty (small) subset Ω *of* $\mathcal{S}_k$.

1. *Solve the following linear program*

$$\max_{(x,\varepsilon)} \ \varepsilon \tag{4.5}$$
$$s.t. \ \ x(S) + \varepsilon f(S) \leq c(S), \ \forall S \in \Omega$$
$$A^k x \leq b^k.$$

Let (x^k, ε^k) *and* (ν^k, μ^k) *be primal and dual optimal solutions of* (4.5).

2. *Consider the separation problem*

$$\max_{S \in \mathcal{S}_k} \big(x^k(S) + \varepsilon^k f(S) - c(S)\big).$$

If the optimal value is positive, then find a set T *in* $\mathcal{S}_k$ *that satisfies*

$$x^k(T) + \varepsilon^k f(T) - c(T) > 0, \tag{4.6}$$

set $\Omega := \Omega \cup \{T\}$*, and go to 1.*

3. Define $\lambda^k \in \mathbb{R}^{\mathcal{S}_k}$ *as*

$$\lambda_S^k = \begin{cases} \nu_S^k & \text{if } S \in \Omega \\ 0 & \text{otherwise.} \end{cases}$$

(x^k, ε^k) *and* (λ^k, μ^k) *are primal and dual optimal solutions of* (4.2).

Proposition 4.2.2. *Algorithm 4.2.1 works correctly and terminates after a finite number of steps.*

Proof. Let us consider an arbitrary step of the algorithm, where there exists a set T satisfying (4.6). Clearly T does not belong to Ω. On the other hand, the algorithm stops at the latest when $\Omega = \mathcal{S}_k$. Therefore, it terminates after a finite number of steps. So we only have to prove that if

$$\max_{S \in \mathcal{S}_k} \big(x^k(S) + \varepsilon^k f(S) - c(S) \big) \leq 0,$$

then (x^k, ε^k) and (λ^k, μ^k) are primal and dual optimal solutions of (4.2). Trivially, we have that (x^k, ε^k) is a feasible solution of (4.2). On the other hand, the optimal value of (4.5) is an upper bound of (4.2). Hence, (x^k, ε^k) is an optimal solution of (4.2). From this it follows that the dual optimal values of (4.2) and (4.5) coincide. And therefore, since (λ^k, μ^k) is a feasible solution of the dual problem of (4.2), it is also an optimal solution. $\square$

4.2.2 The (f, r)-Least Core

Let $\Gamma = (N, c, P, \Sigma)$ be a cost allocation game whose imputation set $\mathcal{X}(\Gamma)$ is non-empty, $f : \Sigma^+ \to \mathbb{R}_{>0}$ be a weight function, and $r \in \mathbb{R}^N_{>0}$ be a reference price-vector of Γ. Denote

$$\Lambda := \big\{ \{i\} \,|\, i \in N \big\} \cup \{N\}$$

and define

$$R : \Lambda \to \mathbb{R}_{>0}, \; R(N) = c(N) \text{ and } R(\{i\}) = r_i, \; \forall i \in N.$$

The following algorithm gives us the (f, r)-least core of Γ.

Algorithm 4.2.3. *Computing the* (f, r)*-least core of* $\Gamma = (N, c, P, \Sigma)$.

Given a subset Ω *of* Σ *satisfying* $\Omega \ni N$ *and* $\Omega \backslash \{\emptyset, N\} \neq \emptyset$.

1. *Compute the f-least core radius ε_Ω of $\Gamma_\Omega := (N, c|_\Omega, P, \Omega)$. The f-least core of Γ_Ω is the following set*

$$\mathcal{LC}_f(\Gamma_\Omega) = \big\{x \in \mathcal{X}(\Gamma_\Omega)\,|\, x(S) \leq c(S) - \varepsilon_\Omega f(S),\ \forall S \in \Omega^+\backslash\{N\}\big\},$$

   where $\Omega^+ := \Omega\backslash\{\emptyset\}$.

2. *Compute the (f,r)-least core of Γ_Ω, i.e., the R-nucleolus of the cost allocation game $(N, R, \mathcal{LC}_f(\Gamma_\Omega), \Lambda)$, using Algorithm 2.3.29 or Algorithm 2.3.31 and obtain a vector x^*.*

3. *Consider the separation problem*

$$\max_{S\in\Sigma^+\backslash\{N\}} \big(x^*(S) + \varepsilon_\Omega f(S) - c(S)\big).$$

   If the optimal value is positive, then find a set T in $\Sigma^+\backslash\{N\}$ that satisfies

$$x^*(T) + \varepsilon_\Omega f(T) - c(T) > 0, \tag{4.7}$$

   set $\Omega := \Omega \cup \{T\}$, and go to 1.

4. *$\{x^*\}$ is the (f,r)-least core of Γ.*

Proposition 4.2.4. *Algorithm 4.2.3 works correctly and terminates after a finite number of steps.*

Proof. Let us consider an arbitrary step of the algorithm, where there exists a set T satisfying (4.7). Since x^* belongs to $\mathcal{LC}_f(\Gamma_\Omega)$, we have

$$\max_{S\in\Omega^+\backslash\{N\}} \big(x^*(S) + \varepsilon_\Omega f(S) - c(S)\big) = 0.$$

It follows that T does not belong to Ω. On the other hand, the algorithm stops at the latest when $\Omega = \Sigma$. Therefore, it terminates after a finite number of steps. So we only have to prove that if

$$\max_{S\in\Sigma^+\backslash\{N\}} \big(x^*(S) + \varepsilon_\Omega f(S) - c(S)\big) \leq 0, \tag{4.8}$$

then $\{x^*\}$ is the (f,r)-least core of Γ. For each $x \in \mathcal{X}(\Gamma_\Omega) = \mathcal{X}(\Gamma)$, since $\Omega \subseteq \Sigma$ and $\Omega^+\backslash\{N\} \neq \emptyset$, we have

$$\min_{S\in\Omega^+\backslash\{N\}} \frac{c(S) - x(S)}{f(S)} \geq \min_{S\in\Sigma^+\backslash\{N\}} \frac{c(S) - x(S)}{f(S)}.$$

Hence, there holds

$$\varepsilon_\Omega = \max_{x \in \mathcal{X}(\Gamma_\Omega)} \min_{S \in \Omega^+ \setminus \{N\}} \frac{c(S) - x(S)}{f(S)} \geq \max_{x \in \mathcal{X}(\Gamma)} \min_{S \in \Sigma^+ \setminus \{N\}} \frac{c(S) - x(S)}{f(S)}. \quad (4.9)$$

From (4.8) it follows that

$$\frac{c(S) - x^*(S)}{f(S)} \geq \varepsilon_\Omega, \ \forall S \in \Sigma^+ \setminus \{N\}. \quad (4.10)$$

Since $x^* \in \mathcal{LC}_f(\Gamma_\Omega) \subseteq \mathcal{X}(\Gamma)$, from (4.9) and (4.10) it follows that

$$\varepsilon_\Omega = \min_{S \in \Sigma^+ \setminus \{N\}} \frac{c(S) - x^*(S)}{f(S)} = \max_{x \in \mathcal{X}(\Gamma)} \min_{S \in \Sigma^+ \setminus \{N\}} \frac{c(S) - x(S)}{f(S)},$$

i.e., ε_Ω is the f-least core radius of Γ and x^* belongs to $\mathcal{LC}_f(\Gamma)$. We have that $\{x^*\}$ is the R-nucleolus of $(N, R, \mathcal{LC}_f(\Gamma_\Omega), \Lambda)$. On the other hand, $\mathcal{LC}_f(\Gamma)$ is a subset of $\mathcal{LC}_f(\Gamma_\Omega)$ since

$$\begin{aligned}
\mathcal{LC}_f(\Gamma) &= \mathcal{C}_{\varepsilon_\Omega, f}(\Gamma) \\
&= \left\{ x \in \mathcal{X}(\Gamma) \;\middle|\; \frac{c(S) - x(S)}{f(S)} \geq \varepsilon_\Omega, \ \forall S \in \Sigma^+ \setminus \{N\} \right\} \\
&\subseteq \left\{ x \in \mathcal{X}(\Gamma) \;\middle|\; \frac{c(S) - x(S)}{f(S)} \geq \varepsilon_\Omega, \ \forall S \in \Omega^+ \setminus \{N\} \right\} \\
&= \mathcal{C}_{\varepsilon_\Omega, f}(\Gamma_\Omega) \\
&= \mathcal{LC}_f(\Gamma_\Omega).
\end{aligned}$$

Therefore, $\{x^*\}$ is the R-nucleolus of $(N, R, \mathcal{LC}_f(\Gamma), \Lambda)$ as well, i.e., it is the (f, r)-least core of Γ. $\square$

Remark 4.2.5. *We can use Algorithm 4.2.3 to calculate the (f, r)-least core of a bounded general cost allocation game $\Gamma = (N, c, P, \Sigma)$ with $c(N) > 0$. But we have to choose Ω such that the general cost allocation game Γ_Ω is bounded in order to ensure that the (f, r)-least core of Γ_Ω is well-defined. Everything else works similarly.*

4.2.3 Choosing A Good Starting Set

As we have seen in the previous sections, the constraint generation approach can be used to handle the exponential number of constraints. A good starting set of constraints is very important for this method as it may improve the solving time drastically. The goal of this section is to show how to construct such a good starting set.

Let $\Gamma = (N, c, P, \Sigma)$ be a cost allocation game, r be a reference price vector, and $f : \Sigma^+ \to \mathbb{R}_{>0}$ be a weight function. In order to determine either the f-least core, the f-nucleolus, or the (f,r)-least core of Γ, we have to calculate its f-least core or the f-least cores of some modified cost allocation games. Therefore, we can concentrate on the f-least core of Γ.

Let ε^* be the f-least core radius of Γ. Due to Theorem 2.5.2 and Theorem 2.5.6 we have that

$$\varepsilon^* \leq \min_{\substack{\mathcal{B} \subseteq \Sigma^+ \setminus \{N\} \\ (\mathcal{B},\lambda) \text{ balanced}}} \frac{\sum_{S \in \mathcal{B}} \lambda_S c(S) - c(N)}{\sum_{S \in \mathcal{B}} \lambda_S f(S)} =: \varepsilon_U. \tag{4.11}$$

Equality holds for certain types of cost allocation game, e.g., if $\Sigma = 2^N$, $P = \mathbb{R}^N_+$, and c is monotonically increasing. The value ε_U is an upper bound of the f-least core radius, but it is not easy to compute. Hence, instead of considering all balanced collections, we only look at balanced collections of cardinality 2 or 3. Denote

$$\varepsilon_k := \min_{\substack{\mathcal{B} \subseteq \Sigma^+ \setminus \{N\},\, |\mathcal{B}|=k \\ (\mathcal{B},\lambda) \text{ balanced}}} \frac{\sum_{S \in \mathcal{B}} \lambda_S c(S) - c(N)}{\sum_{S \in \mathcal{B}} \lambda_S f(S)}. \tag{4.12}$$

Then ε_2 and ε_3 are upper bounds of ε^*. One can easily prove the following results

- For $\mathcal{B} = \{S_1, S_2\}$ and $\lambda = (\lambda_1, \lambda_2)$, the pair $(\mathcal{B}, \lambda)$ is balanced iff $\chi_{S_1} + \chi_{S_2} = \chi_N$ and $\lambda_1 = \lambda_2 = 1$.
- For $\mathcal{B} = \{S_1, S_2, S_3\}$ and $\lambda = (\lambda_1, \lambda_2, \lambda_3)$, the pair $(\mathcal{B}, \lambda)$ is balanced iff either

$$\chi_{S_1} + \chi_{S_2} + \chi_{S_3} = \chi_N \quad \text{and} \quad \lambda_1 = \lambda_2 = \lambda_3 = 1$$

or

$$\chi_{S_1} + \chi_{S_2} + \chi_{S_3} = 2\chi_N \quad \text{and} \quad \lambda_1 = \lambda_2 = \lambda_3 = 0.5.$$

Using these results, we can solve the problem (4.12) for $k = 2$ and $k = 3$. In the case of balanced collections of cardinality 2, we have

$$\varepsilon_2 = \min_{\substack{S_1, S_2 \in \Sigma^+ \setminus \{N\} \\ \chi_{S_1} + \chi_{S_2} = \chi_N}} \frac{c(S_1) + c(S_2) - c(N)}{f(S_1) + f(S_2)}. \tag{4.13}$$

For $f = 1$, $f = |\cdot|$, or $f = c$, one can easily prove that a pair (S_1, S_2) is an optimal solution of (4.13) if and only if it is an optimal solution of the following problem

$$\min_{\substack{S_1,S_2\in\Sigma^+\setminus\{N\}\\ \chi_{S_1}+\chi_{S_2}=\chi_N}} c(S_1)+c(S_2). \tag{4.14}$$

At this point, the reader may ask how can one solve (4.14), as there are exponentially many coalitions in Σ. Obviously, we do not want to evaluate the cost of every coalition. We consider the case that the cost allocation game Γ is a combinatorial cost game, i.e., the cost function is given by an optimization problem for every coalition $S \in \Sigma^+$ as follows

$$\begin{aligned} c(S) := \min_{\xi}\ & c\xi && (4.15)\\ s.t.\ \ & B\xi \geq C\chi_S \\ & D\xi \geq d \\ & \xi_j \in Z_j,\ j = 1,2,\ldots,k, \end{aligned}$$

where χ_S is the incidence vector of S and Z_j is a set of either real, integer, or binary numbers. Then we can rewrite (4.14) as

$$\begin{aligned} \min_{z^1,z^2,\xi^1,\xi^2}\ & c\xi^1 + c\xi^2 && && (4.16)\\ s.t.\ \ & B\xi^l \geq Cz^l, && l = 1,2 \\ & D\xi^l \geq d, && l = 1,2 \\ & z_i^1 + z_i^2 = 1, && \forall i \in N \\ & 1 \leq z^1(N) \leq z^2(N) \\ & \xi_j^l \in Z_j, && l = 1,2,\ j = 1,2,\ldots,k, \\ & z^1, z^2 \in \chi_\Sigma, \end{aligned}$$

which can be solved even if we do not know the cost of every coalition explicitly. The variables z^1 and z^2 in (4.16) are the incidence vectors of the set S_1 and S_2 in (4.14). It remains to answer the question of how can one handle the last constraint, i.e., $z^1, z^2 \in \chi_\Sigma$. In the case $\Sigma = 2^N$, this constraint is nothing else than $z^1, z^2 \in \{0,1\}^N$. In the general case, with $\Sigma \subsetneq 2^N$, we solve (4.16) using a constraint generation approach. Firstly, we solve a relaxed problem of (4.16), where the requirement $z^1, z^2 \in \chi_\Sigma$ is replaced by $z^1, z^2 \in \{0,1\}^N$. If every obtained optimal solution $(\bar{z}^1, \bar{z}^2, \bar{\xi}^1, \bar{\xi}^2)$ of the relaxed problem is infeasible for (4.16), i.e., if $\bar{z}^1, \bar{z}^2 \in \chi_\Sigma$ is violated, say $\bar{z}^1 \notin \chi_\Sigma$, then we add the constraint

$$\sum_{i\in N:\bar{z}_i^1=1} (1 - z_i^1) + \sum_{i\in N:\bar{z}_i^1=0} z_i^1 \geq 1$$

to the relaxed problem and solve it again. The above constraint is equivalent to $z^1 \neq \bar{z}^1$. This process is repeated until we have an optimal solution of the relaxed problem, which is also a feasible and hence optimal solution of (4.16). The case of balanced collections of cardinality 3 can be treated similarly. Once we have balanced pairs $(\mathcal{B}, \lambda)$, which are optimal or suboptimal solutions of (4.12) for $k = 2$ or $k = 3$, we can add every set in $\mathcal{B}$ to the starting set of our constraint generation approach that finds the f-least core.

For the proportional least core or the weak least core, i.e., $f = c$ or $f = |\cdot|$, we can do even more. If $f = c$, then we have

$$\varepsilon_U = \min_{\substack{\mathcal{B}\subseteq\Sigma^+\setminus\{N\} \\ (\mathcal{B},\lambda) \text{ balanced}}} \left(1 - \frac{c(N)}{\sum_{S\in\mathcal{B}} \lambda_S c(S)}\right). \tag{4.17}$$

If $f = |\cdot|$, then we have

$$\varepsilon_U = \min_{\substack{\mathcal{B}\subseteq\Sigma^+\setminus\{N\} \\ (\mathcal{B},\lambda) \text{ balanced}}} \frac{\sum_{S\in\mathcal{B}} \lambda_S c(S) - c(N)}{|N|}. \tag{4.18}$$

The sets of optimal solutions of (4.17) and (4.18) coincide. A pair $(\mathcal{B}, \lambda)$ is an optimal solution of these problems if and only if it is an optimal solution of the following problem

$$\min_{\substack{\mathcal{B}\subseteq\Sigma^+\setminus\{N\} \\ (\mathcal{B},\lambda) \text{ balanced}}} \sum_{S\in\mathcal{B}} \lambda_S c(S).$$

We can use this fact to improve a given starting set Ω, which differs from the set $\Sigma^+\setminus\{N\}$. Given two numbers $\delta_1, \delta_2 \in [0, 1]$, $\delta_1 + \delta_2 > 0$, we find two sets $T_1, T_2 \in \Sigma\setminus\{N\}$ and a vector $\lambda \in \mathbb{R}_+^\Omega$ such that at least one of the two sets T_1 and T_2 is non-empty and $(\chi_{T_1}, \chi_{T_2}, \lambda)$ is an optimal solution of the following problem

$$\begin{aligned}
&\min_{z,\lambda} \delta_1 c[z^1] + \delta_2 c[z^2] + \sum_{S\in\Omega} \lambda_S c(S) &\quad (4.19)\\
s.t.\ \ & \delta_1 z_i^1 + \delta_2 z_i^2 + \sum_{S\in\Omega: S\ni i} \lambda_S = 1,\ \forall i \in N\\
& z^1(N) \le z^2(N) \le |N| - 1\\
& z^2(N) \ge 1\\
& z^1, z^2 \in \chi_\Sigma\\
& \lambda_S \in [0,1],\ \forall S \in \Omega,
\end{aligned}$$

where $c[z]$ is the cost of the coalition whose incidence vector is z. This problem is feasible if, for example, Ω contains every set of each single player. In order to solve (4.19), we also use the definition of the cost function in (4.15) and rewrite it as a minimization problem similarly to what we have done with (4.16). We then add the non-empty sets T_1 and T_2 to Ω.

The above tricks provide a good starting set in the case that (4.11) holds equality, i.e., $\varepsilon^* = \varepsilon_U$, or $\varepsilon_U - \varepsilon^*$ is small. In general, the quality of the starting set obtained in that way may be unsatisfactory, since the gap between ε^* and ε_U can be large. It happens because the requirement that the price must belong to the polyhedron P,

$$P = \{x \in \mathbb{R}^N \mid Ax \le b\},$$

is ignored in the considered heuristics. Therefore, in that case, we have to consider this requirement as well. In the following, we present a heuristic to improve a given starting set of constraints in the case $b = 0$ *and* $f = c$. The problem of finding the proportional least core of Γ is the following

$$\begin{aligned}
&\max_{(x,\varepsilon)} \ \varepsilon &\quad (4.20)\\
s.t.\ \ & x(S) + \varepsilon c(S) \le c(S),\ \forall S \in \Sigma^+ \backslash \{N\}\\
& x(N) = c(N)\\
& Ax \le 0.
\end{aligned}$$

Its dual problem is

$$\min_{(\lambda,\mu,\nu)} \sum_{S\in\Sigma^+\setminus\{N\}} \lambda_S c(S) + \mu c(N) \tag{4.21}$$

$$\begin{aligned} s.t. \quad & \sum_{\substack{S\in\Sigma^+\setminus\{N\} \\ S\ni i}} \lambda_S + \mu + (A_{\cdot i})^T \nu = 0,\ \forall i \in N \\ & \sum_{S\in\Sigma^+\setminus\{N\}} \lambda_S c(S) = 1 \\ & \lambda \geq 0,\ \nu \geq 0, \end{aligned}$$

where $A_{\cdot i}$ denotes the i-th. column of the matrix A. If $P = \mathbb{R}^N$, we just ignore the matrix A in the above linear program. The dual problem (4.21) can be rewritten as

$$\min_{(\lambda,\mu,\nu)} 1 + \mu c(N) \tag{4.22}$$

$$\begin{aligned} s.t. \quad & \sum_{\substack{S\in\Sigma^+\setminus\{N\} \\ S\ni i}} \lambda_S + \mu + (A_{\cdot i})^T \nu = 0,\ \forall i \in N \\ & \sum_{S\in\Sigma^+\setminus\{N\}} \lambda_S c(S) = 1 \\ & \lambda \geq 0,\ \nu \geq 0. \end{aligned}$$

The dual problem in this form is unlikely solvable. On one hand, it is impossible to evaluate the cost function explicitly for every coalition with games having many players. On the other hand, if we want to use the definition of the cost function directly as we did above, then it is unknown how to formulate the second constraint of the dual problem. Therefore, we have to transform it into some equivalent problem, which can be handled more easily. If (4.22) has an optimal solution $(\lambda^*, \mu^*, \nu^*)$ with $\mu^* < 0$, then $(\lambda^*, \mu^*, \nu^*)$ is an optimal solution of the following optimization problem

$$\min_{(\lambda,\mu,\nu)} 1 + \mu c(N) \tag{4.23}$$

$$\begin{aligned} s.t. \quad & \sum_{\substack{S\in\Sigma^+\setminus\{N\} \\ S\ni i}} \lambda_S + \mu + (A_{\cdot i})^T \nu = 0,\ \forall i \in N \\ & \sum_{S\in\Sigma^+\setminus\{N\}} \lambda_S c(S) = 1 \\ & \lambda \geq 0,\ \nu \geq 0,\ \mu < 0, \end{aligned}$$

which can be reformulated by using the transformation

$$\mu := -\frac{1}{\sum_{S\in\Sigma^+\setminus\{N\}} \theta_S c(S)},$$

$$\lambda := -\mu\theta, \text{ and } \nu := -\mu\tau$$

as follows:

$$\begin{aligned} &\min_{(\theta,\tau)} \; 1 - \frac{c(N)}{\sum_{S\in\Sigma^+\setminus\{N\}} \theta_S c(S)} \\ s.t. \; &\sum_{\substack{S\in\Sigma^+\setminus\{N\} \\ S\ni i}} \theta_S + (A_{\cdot i})^T\tau = 1, \; \forall i \in N \\ &\theta \geq 0, \; \theta \neq 0, \; \tau \geq 0. \end{aligned} \tag{4.24}$$

On the other hand, the optimization problem (4.24) is equivalent to the following one

$$\begin{aligned} &\min_{(\theta,\tau)} \sum_{S\in\Sigma^+\setminus\{N\}} \theta_S c(S) \\ s.t. \; &\sum_{\substack{S\in\Sigma^+\setminus\{N\} \\ S\ni i}} \theta_S + (A_{\cdot i})^T\tau = 1, \; \forall i \in N \\ &\theta \geq 0, \; \theta \neq 0, \; \tau \geq 0. \end{aligned} \tag{4.25}$$

Therefore, if (4.22) has an optimal solution $(\lambda^*, \mu^*, \nu^*)$ with $\mu^* < 0$, then, instead of solving (4.22), we can find an optimal solution (θ^*, τ^*) with $\theta^* \neq 0$ of the following linear program

$$\begin{aligned} &\min_{(\theta,\tau)} \sum_{S\in\Sigma^+\setminus\{N\}} \theta_S c(S) \\ s.t. \; &\sum_{\substack{S\in\Sigma^+\setminus\{N\} \\ S\ni i}} \theta_S + (A_{\cdot i})^T\tau = 1, \; \forall i \in N \\ &\theta \geq 0, \; \tau \geq 0. \end{aligned} \tag{4.26}$$

Proposition 4.2.6. *If Σ is a partitioning family, then $\mu^* < 0$ in every optimal solution $(\lambda^*, \mu^*, \nu^*)$ of (4.22).*

Proof. If we can prove that there exists a feasible solution (λ, μ, ν) of (4.22) with $\mu < 0$, then the proposition is true. Since Σ is a partitioning family, there exists $\delta \in \mathbb{R}_+^{\Sigma^+ \setminus \{N\}}$ such that

$$\sum_{S \in \Sigma^+ \setminus \{N\}} \delta_S \chi_S = \chi_N.$$

Clearly, $\delta \neq 0$. Since the cost function is positive, we have

$$\sum_{S \in \Sigma^+ \setminus \{N\}} \delta_S c(S) > 0.$$

Define

$$\mu := -\frac{1}{\sum_{S \in \Sigma^+ \setminus \{N\}} \delta_S c(S)} \quad \text{and} \quad \lambda := -\mu\delta.$$

Then there hold

$$\mu < 0, \; \lambda \geq 0,$$
$$\sum_{S \in \Sigma^+ \setminus \{N\}} \lambda_S \chi_S + \mu \chi_N = 0,$$

and

$$\sum_{S \in \Sigma^+ \setminus \{N\}} \lambda_S c(S) = 1,$$

i.e., $(\lambda, \mu, 0)$ is a feasible solution of (4.22) with $\mu < 0$. □

From the above results we have the following proposition.

Proposition 4.2.7. *If Σ is a partitioning family, then there exists an optimal solution (θ^*, τ^*) of* (4.26) *which satisfies $\theta^* \neq 0$. Moreover, for any optimal solution (θ^*, τ^*) with this property, the triple $(\bar{\lambda}, \bar{\mu}, \bar{\nu})$ defined by*

$$\bar{\mu} := -\frac{1}{\sum_{S \in \Sigma^+ \setminus \{N\}} \theta_S^* c(S)},$$

$$\bar{\lambda} := -\bar{\mu}\theta^* \;\; \textit{and} \;\; \bar{\nu} := -\bar{\mu}\tau^*$$

is an optimal solution of (4.21).

Therefore, under the assumption that Σ is a partitioning family, instead of solving the dual problem (4.21) we can find an optimal solution (θ^*, τ^*) of (4.26) satisfying $\theta^* \neq 0$. Apparently, (4.26) can only be solved via a column generation approach. This method is as time-consuming as solving the original problem (4.20) itself, which is undesirable. However, we do not want to solve the linear program (4.26). But based on it we may improve some given non-empty starting set $\Omega \subsetneq \Sigma^+ \backslash \{N\}$ of constraints as follows. Let α be a given positive number in $(0,1]$. We calculate the cost of each coalition S in Ω. Consider the following optimization problem

$$\begin{aligned} \min_{(z,\theta,\tau)} \ & \alpha c[z] + \sum_{S \in \Omega^+ \backslash \{N\}} c(S)\theta_S \qquad (4.27) \\ s.t. \ & \alpha z_i + \sum_{\substack{S \in \Omega^+ \backslash \{N\} \\ S \ni i}} \theta_S + (A_{\cdot i})^T \tau = 1, \ \forall i \in N \\ & \theta \geq 0, \ \tau \geq 0 \\ & z \in \chi_{\Sigma^+} \backslash \chi_{\Omega \cup \{N\}}, \end{aligned}$$

where $c[z]$ denotes the cost of the coalition having the incidence vector z. The optimization problem (4.27) is a relaxation of (4.26) and is feasible if, for example, Ω contains every set of each single player. Using the definition of the cost function in (4.15), we can rewrite (4.27) as

$$\begin{aligned} \min_{(z,\xi,\theta,\tau)} \ & \alpha c\xi + \sum_{S \in \Omega^+ \backslash \{N\}} c(S)\theta_S \qquad (4.28) \\ s.t. \ & \alpha z_i + \sum_{\substack{S \in \Omega^+ \backslash \{N\} \\ S \ni i}} \theta_S + (A_{\cdot i})^T \tau = 1, \ \forall i \in N \\ & B\xi \geq Cz \\ & D\xi \geq d \\ & \theta \geq 0, \ \tau \geq 0 \\ & \xi_j \in Z_j, \ j = 1, 2, \ldots, k, \\ & z \in \chi_{\Sigma^+} \backslash \chi_{\Omega \cup \{N\}}. \end{aligned}$$

It remains to answer the question of how can one handle the last constraint on z, i.e., $z \in \chi_{\Sigma^+} \backslash \chi_{\Omega \cup \{N\}}$. If $\Sigma = 2^N$, then this constraint is nothing else than

$$z \in \{0,1\}^N, \ \ 1 \leq z(N) \leq |N| - 1, \ \ \text{and} \ \ z \neq \chi_S, \ \forall S \in \Omega.$$

The condition $z \neq \chi_S$ for any given coalition S can be written as

$$\sum_{i \in N : \chi_S^i = 1} (1 - z_i) + \sum_{i \in N : \chi_S^i = 0} z_i \geq 1.$$

If $\Sigma \subsetneq 2^N$, then we solve (4.28) using a constraint generation approach. Firstly, we solve a relaxed problem of (4.28), where the requirement $z \in$

$\chi_{\Sigma^+} \backslash \chi_{\Omega \cup \{N\}}$ is replaced by $z \in \chi_{2^N \backslash \{\emptyset\}} \backslash \chi_{\Omega \cup \{N\}}$. If every obtained optimal solution $(z^*, \xi^*, \theta^*, \tau^*)$ of the relaxed problem violates the constraint that z^* belongs to $\chi_{\Sigma^+} \backslash \chi_{\Omega \cup \{N\}}$, then we add the constraint $z \neq z^*$, i.e.,

$$\sum_{i \in N: z_i^* = 1} (1 - z_i) + \sum_{i \in N: z_i^* = 0} z_i \geq 1,$$

to the relaxed problem and solve it again. This process is repeated until we have an optimal solution of the relaxed problem which is feasible and hence optimal for (4.28). Once we have an optimal (or sub-optimal) solution $(z^*, \xi^*, \theta^*, \tau^*)$ of (4.28), we add the set S^* whose incidence vector is z^* to Ω. This procedure can be repeated several times in order to get a good starting set.

In problem (4.28), α is a given parameter. We can also consider it as a non-negative real variable, but then the problem becomes non-linear and can only be easily solved for "simple" cost functions c.

4.2.4 The Separation Problem

An important part in the constraint generation approaches for the f-least core, the f-nucleolus, and the (f, r)-least core consists of solving the separation problem to verify whether a given pair $(\bar{x}, \bar{\varepsilon})$ is feasible, and if not, finding a cut that separates $(\bar{x}, \bar{\varepsilon})$ from the set of feasible solutions. The separation problem in general has the form

$$\max_{S \in \Lambda} \big(\bar{x}(S) + \bar{\varepsilon} f(S) - c(S) \big) \tag{4.29}$$

with some non-empty family $\Lambda \subseteq \Sigma^+ \backslash \{N\}$. In the case of the f-least core or the (f, r)-least core, we have $\Lambda = \Sigma^+ \backslash \{N\}$. For the f-nucleolus, the family Λ is equal to $\Sigma^+ \backslash \{N\}$ in the first step and shrinks during the calculation step by step. Again, we consider the case that the cost allocation game Γ is a combinatorial cost allocation game, where the cost function is given by the optimization problem (4.15). For $f = \alpha + \beta |\cdot| + \gamma c$, with $\alpha, \beta, \gamma \geq 0$ and $\alpha + \beta + \gamma > 0$, if $\gamma \bar{\varepsilon} \leq 1$, then we can rewrite (4.29) as follows

$$\begin{aligned} \max_{(z,\xi)} \ & \alpha \bar{\varepsilon} + \sum_{i \in N} (\bar{x}_i + \beta \bar{\varepsilon}) z_i + (\gamma \bar{\varepsilon} - 1) c \xi \qquad (4.30) \\ s.t. \ & B\xi \geq Cz \\ & D\xi \geq d \\ & \xi_j \in Z_j, \ j = 1, 2, \ldots, k, \\ & z \in \chi_\Lambda. \end{aligned}$$

This trick can be used for calculating the f-least core, the f-nucleolus, and the (f, r)-least core because of Proposition 2.3.4 and because each input $(\bar{x}, \bar{\varepsilon})$ of the separation problem in our algorithms is a pair composed of a vector in the f-least core of some cost allocation game and the f-least core radius of that game. If the cardinality of the set $2^N \backslash \Lambda$ is small, then we formulate the constraint $z \in \chi_\Lambda$ as

$$z \in \{0,1\}^N \quad \text{and} \quad z \neq \chi_S, \ \forall S \in 2^N \backslash \Lambda,$$

where the condition $z \neq \chi_S$ is equivalent to

$$\sum_{i \in N : \chi_S^i = 1} (1 - z_i) + \sum_{i \in N : \chi_S^i = 0} z_i \geq 1.$$

If the cardinality of Λ is small, then we can introduce some binary variables w_S, $S \in \Lambda$, satisfying

$$\sum_{S \in \Lambda} w_S = 1$$

and represent the constraint $z \in \chi_\Lambda$ as

$$z = \sum_{S \in \Lambda} w_S \chi_S.$$

In general, it may happen that the sets Λ and $2^N \backslash \Lambda$ have exponential many elements. In this case, to solve (4.30), we should not formulate the constraint $z \in \chi_\Lambda$ explicitly, but use a constraint generation approach as we did in Subsection 4.2.3.

The separation problem is NP-hard in general and it may be very time-consuming to find an optimal solution of (4.30). However, in order to find a violated constraint, it is not necessary to solve (4.30) to optimality. On the other hand, adding constraints with small violations leads to only slight changes of the solution. Consequently, it converges slowly and solving each step at the end is expensive because of the large number of added constraints. Therefore, good feasible solutions of (4.30) with positive objective value are needed. We also should add not only one but several violated constraints in each step. However, their number should be kept small in order to prevent a fast grow in the size of the constraints set. For this, we need a criterion assessing the obtained violated constraints. Given two violated coalitions S_1 and S_2, i.e., their corresponding constraints are violated, we have

$$\bar{x}(S_i) + \bar{\varepsilon} f(S_i) - c(S_i) > 0, \ i = 1, 2.$$

We say that S_1 is more violated than S_2 if

$$\frac{\bar{x}(S_1) - c(S_1)}{f(S_2)} > \frac{\bar{x}(S_2) - c(S_2)}{f(S_2)}. \tag{4.31}$$

In our code, it is allowed to add up to 10 violated constraints per step in the first 50 steps and up to 5 violated constraints per step later on.

4.2.5 Heuristics for the Separation Problem

In this subsection, we consider several heuristics for the separation problem. Heuristics do not solve the original problem in general, but they may speed up the solving process immensely. We are interested in primal and dual heuristics. The primal heuristic provides good starting solutions for the mixed integer program (4.30), while the dual heuristic give us some measure to answer the question of whether the current best solution is good enough.

Primal Heuristics

During the constraint generation process, we add several constraints to the starting set. A natural idea is to create a heuristical method where some z-variables in (4.30) are fixed to 1 in a reliable way. For this, a function evaluating the history of the added constraints is required. Let h be a function defined for every finite sequence of binary numbers. Typically, we can choose h equal to the sum of the values of some function $e^{at}x$ with $a > 0$ applied to the elements of the input sequence, where t and x correspond to their indexes and their values, respectively. This choice is reasonable, since the recent added constraints are more likely to provide reliable information for fixing variables than the older ones, while the ones that were added long before may have no connection anymore with the current separation problem and should be ignored. As a result we can define h for any given sequence of binary numbers $\{\omega_1, \omega_2, \ldots, \omega_m\}$ as

$$h(\omega_1, \omega_2, \ldots, \omega_m) = \begin{cases} \sum_{j=1}^{m} e^{aj}\omega_j & \text{if } m \leq K \\ \sum_{j=n-K+1}^{m} e^{aj}\omega_j & \text{otherwise,} \end{cases}$$

for some given number $K \in \mathbb{N}$. Let $\mathbf{1_m}$ denote the sequence of m numbers 1. For our calculations, we choose $a = 0.1$ and $K = 30$. Define

$$h_m := h(\mathbf{1_m})$$

and

$$H(\omega_1, \omega_2, \ldots, \omega_m) := \frac{h(\omega_1, \omega_2, \ldots, \omega_m)}{h_m}.$$

The function H can be used to find heuristically the players which are likely to belong to a violated coalition of the current separation problem as follows. We consider now the $m+1$-th separation problem. Since it is allowed to add more than one violated coalition at each separation step, we only consider the most violated one according to (4.31). Denote this set of coalitions as $(S_1, S_2, \ldots, S_m)$. For each player $i \in N$, we have a sequence $\{\chi^i_{S_1}, \chi^i_{S_2}, \ldots, \chi^i_{S_m}\}$ which tells us whether the player i belongs to the most violated coalition found in each past separation step. For each player i, denote $H_m(i) := H(\chi^i_{S_1}, \chi^i_{S_2}, \ldots, \chi^i_{S_m})$. $H_m(i) = 1$ means that i belongs to every coalition in $\{S_1, S_2, \ldots, S_m\}$. If $H_m(i)$ is almost 1, then i may belong to a violated coalition in the present step. Let ν_1 be a positive number which is smaller but close to 1, e.g., $\nu_1 = 0.97$. For every player i satisfying $H_m(i) \geq \nu_1$, we fix $z_i = 1$ in (4.30) and solve (4.30).

Another fixing method is based on the idea of Relaxation Induced Neighborhood Search (RINS) [9]. Let ν_2 and ν_3 be positive numbers which are smaller but close to 1 and $\nu_2 < \nu_1$ (e.g., $\nu_2 = 0.94$ and $\nu_3 = 0.91$). Let z^R be an optimal LP-relaxation solution of (4.30). We then fix $z_i = 1$ for every player i satisfying $H_m(i) \geq \nu_2$ and $z^R_i \geq \nu_3$ and solve (4.30).

Computational results show that our heuristical fixing methods are very effective. They can find violated coalitions in almost every separation steps. It is not clear whether we just interrupted the heuristics when they fail too early or there does not exist any violated coalition satisfying the fixation. By using the found violated coalitions as starting solutions for the original separation problem (4.30), we can identify violated coalitions faster or find worse coalitions in the same given time limit. The solving process also seems to be more stable in the sense that it needs less separation steps and has better running time. Table 4.1 presents the running times (in seconds) and the separation steps numbers for our ticket pricing application in Chapter 8 with and without the primal heuristics. These numbers are the averages of three runs. The computations were done on a PC with an Intel Core2 Quad 2.83GHz processor and 16GB RAM. CPLEX 11.2 was used as linear and integer program solver. The use of the primal heuristics yields a speedup of factor 1.45.

We also use the solution polishing heuristic of CPLEX. It often finds a much better solution of the separation problem from an initial one after just a few seconds.

	With primal heuristics	Without primal heuristics
Running time (s)	41048	59375
Separation steps	75.67	91.33

Table 4.1: With primal heuristics vs. without primal heuristics

Stopping Criterion and Dual Heuristic

Since we only want to find a good solution of (4.30) with a positive objective value, we need a stopping criterion. We stop the solver whenever a time limit or a gap limit is exceeded. The gap limit is given by a function depending on the objective value of the current best found solution of (4.30). Let p_1 and p_2 be two price vectors such that the optimal value of the separation problem (4.30) is large when $\bar{x} = p_1$ and small when $\bar{x} = p_2$. It is much easier to solve (4.30) with $\bar{x} = p_1$ to a given gap than with $\bar{x} = p_2$ to the same gap. Therefore, the gap limit function should be decreasing on the objective value of the current best solution. For example, we can choose it equal to av^b with some numbers $a > 0$ and $b < 0$ for $v \geq 0$, where v is the objective value of the current best solution.

It is hard to improve not only the primal bound of (4.30) but also its dual bound. However, the optimal values of successive separation problems are only slightly different. Therefore, since we do not solve the separation problem to optimality, an exact dual bound is not required and we can use the best dual bound of the separation problems in the past to evaluate the obtained solutions in the current separation step. The dual heuristic works as follows. Define a number called heuristical dual bound and set it to infinite at the beginning of the constraint generation process. Whenever it is larger than the dual bound of the current separation problem or smaller than the current best found feasible solution, we set it to the current dual bound. The later can happen, but in pratice it only occurs a few (say, less than five) times in the first steps of the constraint generation process. The heuristical gap is defined as

$$\frac{|\text{heuristical dual bound} - \text{best objective value}|}{10^{-10} + |\text{best objective value}|}.$$

The solver is stopped whenever either the time limit is exceeded or the minimum of the gap and the heuristical gap is smaller than the gap limit. But if the heuristical dual bound and the dual bound of the current separation problem are close to each other, then we keep the solver running for a short time in order to guarantee that if the heuristical dual bound is updated then the improvement is not too small. Numerical computations for the ticket pricing problem in Chapter 8 show that the number of separation steps where the heuristical dual bound needs to be updated is only 21.33, while the total number of steps is 75.67. These numbers are the averages of three runs. Moreover, most of the updates happen at the beginning of the constraint generation process, while later on, when the separation problem is very expensive, only a few updates are required. That means we can save a lot of time spent on improving the dual bound. Using dual heuristic, we can reduce the average computational time of three runs for the ticket pricing problem by a factor of 1.34.

Chapter 5

The Fairness Distribution Diagram

This chapter deals with the question of how one can evaluate the fairness of a given price vector and/or compare two different price vectors visually. For games with many players, it is impossible to calculate the cost and profit of every coalition with a given price vector. Therefore, we should only consider some *essential coalitions* and plot some graphs representing their profits. These graphs are called *fairness distribution diagram*. We are going to describe its construction in the follwing.

Given are a combinatorial cost allocation game $\Gamma = (N, c, P, \Sigma)$, i.e., the cost function c can be represented by (4.1), a weight function $f = \alpha + \beta|\cdot| + \gamma c$ with $\alpha, \beta, \gamma \geq 0$ and $\alpha + \beta + \gamma > 0$, and a price vector p. The first task is to create a pool of essential coalitions. This pool should contain coalitions S which have the smallest f-profit $\frac{c(S)-p(S)}{f(S)}$. To do so, we consider the following optimization problem

$$\begin{aligned} &\max_{\varepsilon} \ \varepsilon && (5.1)\\ &s.t. \ \ \varepsilon f(S) \leq c(S) - p(S), \ \forall S \in \Sigma \backslash \{\emptyset, N\}. \end{aligned}$$

The optimal value of (5.1) is the smallest f-profit of all coalitions in $\Sigma\backslash\{\emptyset, N\}$ with the price vector p. This optimazation problem can be solved using constraint generation approach as we did by calculating the f-least core radius of Γ in Chapter 4. The separation problem remains the same. The constraint generation approach collects on its course many coalitions and also coalitions with the smallest f-profit at the end. We add all these coalitions to our pool in non-decreasing order regarding their f-profits. The curve representing the f-profits of the coalitions in the pool may be not smooth. In other words, the number of samplers (coalitions) is not sufficiently large for representing the f-profits of all coalitions in $\Sigma\backslash\{\emptyset, N\}$. Therefore, we need to enlarge our pool. We make a copy of the pool and denote it as Π. Let m be a given parameter satisfying $0 < m < |N|$. Typically we choose $m = 0, 1, 2$, or 3. For each coalition S in Π, we denote its f-profit by ε_S and consider the following optimization problem for each player $i \in S$

$$\begin{aligned} \min_{T \in \Sigma\backslash(\{\emptyset,N\}\cup\Pi)} \quad & c(T) - p(T) - \varepsilon_S f(T) \\ s.t. \quad & S\backslash\{i\} \subseteq T \\ & |T| \leq |S| + m. \end{aligned} \tag{5.2}$$

The constraints of (5.2) means that coalition T is only slightly different from S. One can rewritte (5.2) into a (mixed) integer program as we did in Chapter 4. Using a MIP solver like CPLEX one can solve this problem and obtain a set of feasible solutions. By applying this process for every player i in S we may obtain new coalitions whose f-profits are close to ε_S. We insert these coalitions into the pool such that coalitions in the pool are still sorted in non-decreasing order regarding their f-profits. After considering all coalitions in Π we may repeat this whole process several times in order to have a sufficiently large number of samplers of f-profits. We stop whenever the f-profit curve of coalitions in the pool is smooth or the pool contains already all coalitions in $\Sigma\backslash\{\emptyset, N\}$. The last task is to eliminate coalitions having similar f-profits. Two coalitions are said to have similar f-profits if their relative difference is small, say less than 0.02%. This number is problem-dependent. For each pair of two coalitions

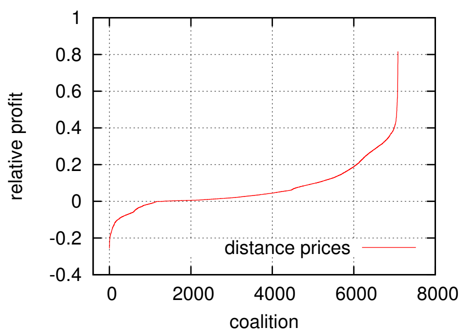

Figure 5.1: c-profits of essential coalitions

having similar f-profits, we just delete one of them from the pool. The reason for this task is that, since it is imposible to calculate the costs and the f-profits of all coalitions in a large cost allocation game, we are only interested in samplers of the possible f-profit values. The f-profit curve of coalitions in the pool at the end looks like Figure 5.1. This figure represents the c-profit curve with the current distance prices in the ticket pricing problem for the Dutch IC network in Chapter 8. The pool in this example contains 7084 essential coalitions. We can see that there exist many coalitions whose c-profits are smaller than -0.1, i.e., they lose more than 10%. This is an indicator for the unfairness of the distance prices.

Given now are two different price vectors. In order to compare them we create for each price vector its pool of essential coalitions as described above. We add these two pools together and select one of the two prices vector as the default price vector. The coalitions in the new pool are sorted in non-decreasing order regarding their f-profits with the default price vector. In Figure 5.2, which compares the distance prices and the (c, r)-least core prices of the ticket pricing problem in Chapter 8, the distance prices was chosen as the default price vector. The picture on the left side plots the c-profits of coalitions in the pool with the two price vectors. The coalitions whose c-profits are smaller than -0.1 with the distance prices have now positive c-profits with the (c, r)-least core prices. That means the (c, r)-least core prices are more favorable for them. The minimal c-profit with the (c, r)-least core prices is considerably increased compared to the smallest c-profit with the distance prices. With other words, the (c, r)-least core prices make the users more satisfied and decrease their incentive to leave the grand coalition. The picture on the right side of Figure 5.2 also plots the c-profits of coalitions in the pool with the two price vectors but the profits are sorted in non-decreasing order for *both* price vectors.

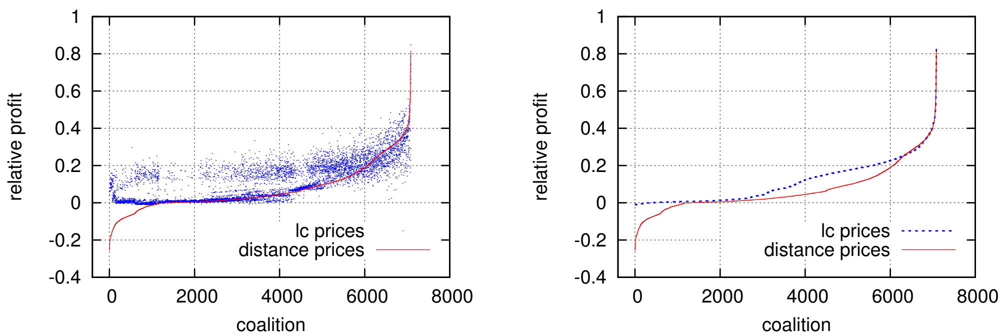

Figure 5.2: Distance vs. (c, r)-least core prices

With the help of the fairness distribution diagram, we have an effective instrument to evaluate the fairness of a given price vector and to compare different price vectors visually.

Chapter 6

A Simple Real Example

In this chapter, we consider a very simple real example with only four players. The goal is to demonstrate how different are the results when we apply different cost allocation methods to the same cost allocation problem. We will also see that the f-nucleolus allocation methods may provide impracticable prices. It is a question of how can one allocate the common cost of a waste water system to the households using it. In Germany, each household must be conneted with the sewage system. They can either do this using the service of the local waterworks company at a high cost or acting themselves. The four considered households represented in Figure 6.1 chose the later. Each household has to be connected to the point S, which denotes a connection point to the sewage system of the city. The cost of each sewage line depends mostly on its length and not on its capacity. The reason is

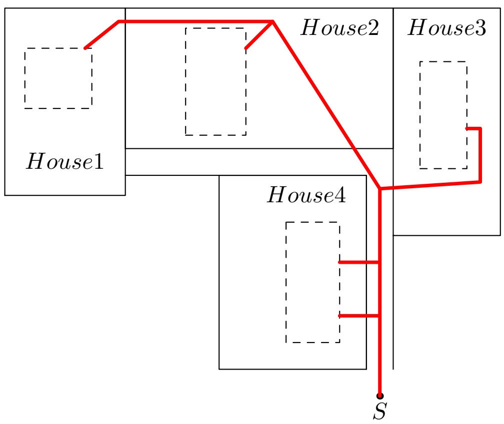

Figure 6.1: A simple real example

that in our example one has to dig the same channel for a pipe independently on its size, while the variation in the pipe's expense is insignificant by choosing pipes with different sizes. The digging cost together with some other costs which are independent on the capacity make up the main part of the total cost. Therefore, a cooperation of the four households saved them a large amount of money. But a cooperation can only be realized if the participants can agree on the price for each of them. In reality, they agreed on a common sewage network represented by the red lines in Figure 6.1 and a simple cost allocation scheme for the common cost. This scheme works as follows. Firstly, the network is decomposed into segments. A segment is just a part of the network such that each household either uses this part completely or does not use it at all in order to connect to S in the built network. For each segment, the price is uniformly distributed among the households who use it. The price for each household is then the sum of the prices of all its segments. The allocated prices was (in German Mark):

$$p_1 = 19159.02,\ p_2 = 13681.18,\ p_3 = 19981.51,\ p_4 = 8754.57.$$

This allocation sounds reasonable and was accepted by all the four households. However, it does not take into account the possibility of contracting out the cooperation and establishing a smaller cooperation of each coalition.

We now construct a cost allocation game for our problem in order to analyse the price p and calculate several prices based on the game theoretical concepts. Consider each household as a player. The player set is $N = \{1, 2, 3, 4\}$, the set of possible coalitions is 2^N, and the set of possible prices is $\mathbb{R}^N_+$. We have a cost allocation game $\Gamma := (N, c, \mathbb{R}^N_+, 2^N)$ where the cost function is given as follows

Coalition	Cost	Coalition	Cost
$\{1\}$	43002.78	$\{2\}$	27223.36
$\{3\}$	30805.28	$\{4\}$	17951.64
$\{1, 2\}$	38770.24	$\{2, 3\}$	42709.61
$\{3, 4\}$	36367.19	$\{1, 3\}$	58489.03
$\{1, 4\}$	48810.10	$\{2, 4\}$	33030.69
$\{1, 2, 3\}$	54299.33	$\{2, 3, 4\}$	49626,30
$\{1, 2, 4\}$	44674.61	$\{1, 3, 4\}$	65405.72
$\{1, 2, 3, 4\}$	61576.32		

Household 1 needs a cooperation with household 2, because without a permission of household 2, household 1 can not use the land of household

2 in order to build his pipe. In that case, he has to build another line, for which an expensive pump system is needed. This is the reason why the cost of household 1 alone is even more than the cost of household 1 and 2 together. And because of the same reason, there hold

$$c(\{1,3\}) > c(\{1,2,3\}),\ c(\{1,4\}) > c(\{1,2,4\}) \text{ and } c(\{1,3,4\}) > c(N).$$

The price p belongs to the core of Γ but it has some drawbacks. The price for coalition $\{1,2,3\}$ with p is then 52821.71 DM, i.e., almost equal to its cost of 54299.33 DM. Coalition $\{1,2,3\}$ can save only 1477.62 DM, while coalition $\{4\}$ can save 9197.07 of 17951.64 DM. Coalition $\{1,2,3\}$ agreed with the price p since the players did not notice their alternative, which can be used as a strong argument in the price negotiation with player 4. The price p favours player 4 and is unfair for coalition $\{1,2,3\}$. If the players 1, 2, and 3 had known it, a cooperation would have been hardly realized with the price p. With other words, player 4 should pay more than he did. The same thing happens for coalition $\{1,2,4\}$ and coalition $\{3\}$. Coalition $\{1,2,4\}$ saves only 3079.84 DM of 44674.61 DM, while coalition $\{3\}$ has a benefit of 10823.77 DM from its individual cost of 30805.28 DM.

Prices obtained by using game theoretical concepts like the f-nucleolus or the (f,r)-least core do not suffer from the above unfairness. Denote l_i as the length of the pipe connecting the household i and S in the network represented in Figure 6.1:

$$l_1 = 113.1,\ l_2 = 87,\ l_3 = 76.4,\ l_4 = 39.38 \text{ (meters)}.$$

Choose the reference price vectors r^1 and r^2 as follows

$$r_i^1 = c(N)\frac{l_i}{\sum_{j=1}^4 l_j},\ i = 1,2,3,4,$$

$$r_i^2 = c(N)\frac{c(\{i\})}{\sum_{j=1}^4 c(\{j\})},\ i = 1,2,3,4.$$

Table 6 presents several prices obtained by using the game theoretical concepts. With our game theoretical allocations, player 3 and player 4 have to pay more while player 2 has to pay less in comparison to p, which eliminates the drawbacks of the price p mentioned above. By choosing c as the weight function f, every vector in the c-least core of Γ has the same third and fourth coordinates. Therefore, the prices for player 3 and player

	Player 1	Player 2	Player 3	Player 4
p	19159.02	13681.18	19981.51	8754.57
1-nucleolus	22525.69	6746.27	20964.54	11339.82
$\lvert\cdot\rvert$-nucleolus	22017.84	6238.42	21472.39	11847.67
c-nucleolus	20735.16	7749.11	20925.00	12167.05
$(1, r^1)$-least core	16545.02	12726.94	20964.54	11339.82
$(\lvert\cdot\rvert, r^1)$-least core	16520.70	11735.55	21472.39	11847.67
(c, r^1)-least core	16419.24	12065.03	20925.00	12167.05
$(1, r^2)$-least core	17924.60	11347.36	20964.54	11339.82
$(\lvert\cdot\rvert, r^2)$-least core	17302.64	10953.62	21472.39	11847.67
(c, r^2)-least core	17442.26	11042.01	20925.00	12167.05

Table 6.1: Several allocations

4 do not change by using c-nucleolus, (c, r^1)-least core, and (c, r^2)-least core cost allocation methods. The same thing happens when we choose the weight function f equal to 1 or the cardinality function $|\cdot|$. Therefore, for each $f \in \{1, |\cdot|, c\}$, the three price vectors f-nucleolus, (f, r^1)-least core, and (f, r^2)-least core differ from each other just in the prices for player 1 and player 2.

The f-nucleolus prices, $f(S) = 1$, $|S|$, $c(S)$, for player 2 are very low, which reflects his strong position against player 1. However, player 1 may think that player 2 is favored with a saving of almost 75 percents in comparison to the individual cost. In contrast, the (f, r^1)-least core price vectors charge player 2 much higher, maybe too high. The reason is that the (f, r^1)-least core prices for player 1 and player 2 are almost proportional to their distances to the connection point S. As we discussed before, without a cooperation with player 2, player 1 has to build an expensive pipe system. That means the price for player 2 must be sufficiently attractive in order to increase his incentive to participate in a common solution for the four households. Choosing r^2, which is proportional to the individual costs, as the reference price vector seems to be more suitable for our cost allocation problem. The (f, r^2)-least core prices for player 1 and player 2 are almost proportional to their individual costs. In our opinion, the (f, r^2)-least core price or the average of the f-nucleolus and the (f, r^2)-least core prices are better than the f-nucleolus and the (f, r^1)-least core prices.

As our earlier observation, one should choose the cost function c as the weight function f, although the difference of the outputs by using the three different weight functions is insignificant in this simple example.

Chapter 7

Allocating Production and Transmission Costs

In this chapter, we want to consider the problem in that a good is produced in some places and then transported to customers via a network. For example the problem of cost distribution for water supply systems, irrigation systems, or gas transportation. Our task is to allocate the common cost fairly to the customers. The cost can be modeled as a non-linear multi-commodity flow problem. However, in order to solve large cost allocation problems, they are linearized using piecewise linear function as in [10]. Based on this model, we solve a cost allocation problem in water resources development in Sweden.

7.1 Production and Transmission Costs

The underlying network is given by a directed graph, where the nodes are production places, customer positions, and intermediate stations. Each arc in the graph represents the direct connection between two nodes. Each customer has a certain demand. The value of the cost function of each group of customers is then the minimal cost of solutions that satisfy the demand of every customer in this group. The cost of each solution is the sum of the production cost and the transmission cost. The production cost is the sum of the production cost of each producer. The production cost of each producer p, denoted by $C_p(q)$, depends on the quantity q of its produced good and some specific parameters. $C_p(q)$ is monotonically increasing, positive for each $q > 0$, and equal to 0 for $q = 0$. For example, C_p

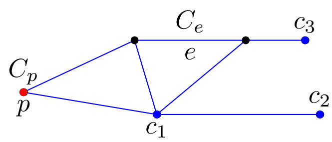

Figure 7.1: Underlying graph

has the formula $\alpha_p q^{\beta_p}$, where α_p is positive and β_p belongs to the interval (0,1] because of the monotonicity of the cost function and the economy of scale. The transmission cost is the sum of the transmission cost on all arcs. The transmission cost on an arc a, denoted by $C_a(q)$, depends on the quantity of goods which are transported through this arc and some specific parameters. $C_a(q)$ is increasing, positive for each $q > 0$, and equal to 0 for $q = 0$. Typically it can be formulated as $\alpha_a q^{\beta_a}$ with some specific parameters α_a and β_a, $\alpha_a > 0$, $\beta_a \in (0, 1]$. The cost function can be modeled as the optimal value of a multi-commodity flow problem with a non-linear objective function as we will see in the next section.

7.2 Nonlinear Multi-commodity Flow Model

To model the cost function as a multi-commodity flow problem, we need to modify the graph slightly. Without loss of generality, we assume that no production place is the tail of an arc in the original graph. If it is not the case, then the following construction must be slightly modified. For each arc a in the original graph whose head does not represent a producer, we define the weight $w_a(q)$ of this arc as the transmission cost $C_a(q)$. For each producer p, we create an artificial node p' and add it to the graph. Then we add the arc pp' to the graph and define its weight $w_{pp'}(q)$ as the production cost $C_p(q)$ of p. For each node i such that pi is an arc in the original graph, we add the arc $p'i$ to the graph, define the weight $w_{p'i}(q)$ of this arc as the transmission cost $C_{pi}(q)$ from p to i, and eliminate the edge pi from the graph. We now have a new weighted directed graph $G = (V, A, w)$.

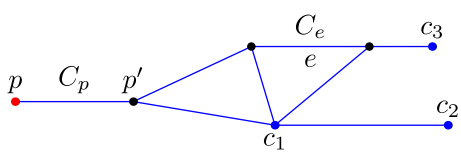

Figure 7.2: Modified graph

For each arc a in A, the weight $w_a(q)$ is monotonically increasing, positive for $q > 0$, and equal to 0 for $q = 0$. We denote the set of customers by C and the set of producers by P. Each customer c in C can be served by some given producers $P_c \subseteq P$. The set P_c is not empty and may differ from P. We can consider the goods that the customer c obtains from the producers P_c as flows from these nodes to c. For each customer $c \in C$, each producer $p \in P_c$, and each arc $a \in A$, the non-negative variable x_a^{pc} denotes the quantity of goods transported from producer p to customer c through the arc a. For some applications the variable x should be integral. We represent this requirement by the constraint $x \in Q$ with some appropriate set Q, which is either a set of integral vectors or a set of real vectors. The quantity q_a of goods on each arc a is then equal to $\sum_{c\in C}\sum_{p\in P_c} x_a^{pc}$. If the head of a is a producer p, then a is the arc between p and its artificial node p' and q_a is the quantity of goods produced by p. If the head of a is not a producer, then q_a represents the quantity of goods transported through the arc a.

Let d_c be the demand of the customer c, d_c is positive and finite. Each group of customers S in C is represented by its incidence vector $z_S \in \{0,1\}^C$

$$\forall k \in C: \; z_S^k = \begin{cases} 1 & \text{if } k \in S \\ 0 & \text{otherwise.} \end{cases}$$

Denote $X_a := \sum_{c\in C}\sum_{p\in P_c} x_a^{pc}$, the minimal cost of each non-empty group S is determined as follows

$$\begin{aligned}
c(S) = \min_x \sum_{a\in A} & w_a(X_a) && (7.1)\\
\sum_{p\in P_c}\sum_{(p,k)\in A} & x_{pk}^{pc} = d_c z_S^c, && \forall c \in C\\
\sum_{p\in P_c}\sum_{(i,p)\in A} & x_{ip}^{pc} = 0, && \forall c \in C\\
\sum_{p\in P_c}\sum_{(c,k)\in A} & x_{ck}^{pc} = 0, && \forall c \in C\\
\sum_{p\in P_c}\sum_{(i,c)\in A} & x_{ic}^{pc} = d_c z_S^c, && \forall c \in C\\
\sum_{(i,j)\in A} x_{ij}^{pc} - \sum_{(j,k)\in A} & x_{jk}^{pc} = 0, && \forall c \in C,\ \forall p \in P_c,\ \forall j \in V\backslash\{p,c\}\\
& x_a^{pc} \geq 0, && \forall a \in A,\ \forall c \in C,\ \forall p \in P_c\\
& x \in Q.
\end{aligned}$$

We require that the functions w_a, $a \in A$, are continuous and the optimization problem (7.1) is feasible for each non-empty subset S of C. Clearly, the set of all feasible solutions is closed and bounded, i.e., compact. On the other hand, the objective function is bounded from below. Therefore, (7.1) has optimal solutions for each non-empty subset S of C and hence the cost function c is well-defined.

7.3 Mixed Integer Model

We now want to approximate the objective function of (7.1) by some piecewise linear function to get an approximated model to (7.1). Assume that there exist positive numbers l_a and non-negative finite numbers u_a for $a \in A$, such that for every non-empty set $S \subseteq C$ there exists an optimal solution x^* of (7.1) that satisfies

$$\left(\sum_{c\in C}\sum_{p\in P_c} x^{*pc}_{\ a} = 0 \vee \sum_{c\in C}\sum_{p\in P_c} x^{*pc}_{\ a} \geq l_a\right) \wedge \sum_{c\in C}\sum_{p\in P_c} x^{*pc}_{\ a} \leq u_a,\ \forall a \in A.$$

We will discuss in the end of this section how to find such numbers. The reason why we need those numbers is that on one hand some types of weight functions that we are interested in cannot be approximated with an arbitrary small given relative error in the neighbourhood of 0 and on the other hand the piecewise linear approximation on a finite grid only works for finite intervals in general. Assume that $u_a \geq l_a$, since if $u_a < l_a$ then we have

$$\sum_{c\in C}\sum_{p\in P_c} x^{*pc}_{\ a} = 0,\ \text{i.e.,}$$

$$x^{*pc}_{\ a} = 0,\ \forall c \in C,\ \forall p \in P_c,$$

and can therefore eliminate these variables without changing the optimal value. We can add to (7.1) the following constraints

$$\sum_{c\in C}\sum_{p\in P_c} x^{pc}_a \leq u_a, \qquad \forall a \in A \tag{7.2}$$

$$\sum_{c\in C}\sum_{p\in P_c} x^{pc}_a = 0 \vee \sum_{c\in C}\sum_{p\in P_c} x^{pc}_a \geq l_a, \qquad \forall a \in A. \tag{7.3}$$

For each arc a, let $\{t^a_1, t^a_2, \ldots, t^a_{k_a}\}$ be a discretization grid of the finite interval $[l_a, u_a]$. We approximate the weight $w_a(q)$ by a function $w^l_a(q)$, which is non-negative and linear in every interval $[t^a_j, t^a_{j+1}]$ for $j = 1, 2, \ldots, k_a - 1$. We set $w^l_a(0) = w_a(0)$. The cost function c is approximated by the function c_l which is defined as follows

$$c_l(S) := \min_x \sum_{a \in A} w^l_a(X_a) \tag{7.4}$$
$$X_a := \sum_{c \in C} \sum_{p \in P_c} x^{pc}_a$$
$$x \text{ satisfies the constraints of (7.1) and (7.2)–(7.3).}$$

Because of the requirements on the original problem (7.1), the approximated problem (7.4) has optimal solutions for each non-empty subset S of C and therefore the function c_l is well-defined.

A natural question arises then, namely how large is the error caused by the linearization. Given numbers $\alpha_2 \geq \alpha_1 > -1$. Assume that for every $a \in A$ the function w^l_a satisfies

$$(1 + \alpha_1) w_a(q) \leq w^l_a(q) \leq (1 + \alpha_2) w_a(q), \ \forall a \in A, \ \forall q \in [l_a, u_a].$$

We will describe in Section 7.4 how one can construct such functions for arbitrary given numbers $\alpha_2 \geq \alpha_1 > -1$. We set $w^l_a(0) = w_a(0) = 0$. Obviously, the following result holds:

Proposition 7.3.1. *If there exist numbers $\alpha_2 \geq \alpha_1 > -1$ such that*

$$(1 + \alpha_1) w_a(q) \leq w^l_a(q) \leq (1 + \alpha_2) w_a(q), \ \forall a \in A, \ \forall q \in \{0\} \cup [l_a, u_a],$$

then there holds

$$(1 + \alpha_1) c(S) \leq c_l(S) \leq (1 + \alpha_2) c(S), \ \forall \emptyset \neq S \subseteq C.$$

The following proposition says that if w^l_a is a good approximation of w_a for each arc a, then the c-least core of Γ is well approximated by the c_l-least core of Γ_l.

Proposition 7.3.2. *Given cost allocation games $\Gamma = (N, c, P, \Sigma)$ and $\Gamma_l = (N, c_l, P, \Sigma)$ with cost functions c and c_l described as above and P satisfies*

$$x \in P \Rightarrow \lambda x \in P, \ \forall \lambda > 0.$$

Let ε be the c-least core radius of Γ and ε_l be the c_l-least core radius of Γ_l. Denote $\beta := \frac{c_l(N)}{c(N)} - 1$ and define

$$\delta(x_l) := \min_{S \in \Sigma^+ \setminus \{N\}} \frac{c(S) - \frac{1}{1+\beta} x_l(S)}{c(S)}.$$

If there exist parameters $\alpha_2 \geq \alpha_1 > -1$ such that

$$(1+\alpha_1) w_a(q) \leq w_a^l(q) \leq (1+\alpha_2) w_a(q),\ \forall a \in A,\ \forall q \in \{0\} \cup [l_a, u_a],$$

then there holds

$$0 \leq \varepsilon - \delta(x_l) \leq (\alpha_2 - \alpha_1) \frac{1-\varepsilon_l}{1+\beta} \leq (\alpha_2 - \alpha_1) \frac{1-\varepsilon}{1+\alpha_1},\ \forall x_l \in \mathcal{LC}_{c_l}(\Gamma_l).$$

Proof. Let x_l be an arbitrary point in the c_l-least core $\mathcal{LC}_c(\Gamma_l)$ of Γ_l. Since there exist parameters $\alpha_2 \geq \alpha_1 > -1$ such that

$$(1+\alpha_1) w_a(q) \leq w_a^l(q) \leq (1+\alpha_2) w_a(q),\ \forall a \in A,\ \forall q \in \{0\} \cup [l_a, u_a],$$

due to Proposition 7.3.1, we have

$$(1+\alpha_1) c(S) \leq c_l(S) \leq (1+\alpha_2) c(S),\ \forall \emptyset \neq S \subseteq C.$$

Hence, due to Proposition 2.3.16, there holds

$$0 \leq \varepsilon - \delta(x_l) \leq (\alpha_2 - \alpha_1) \frac{1-\varepsilon_l}{1+\beta} \leq (\alpha_2 - \alpha_1) \frac{1-\varepsilon}{1+\alpha_1}.$$

□

The optimization problem (7.4) is not in a form which we can handle easily. Therefore, we want to rewrite it as a mixed integer program using the idea of Dantzig [10]. For each arc a, we will represent the function w_a^l in $\{0\} \cup [l_a, u_a]$ by a linear system of equalities and inequalities using the grid $\{t_1^a, t_2^a, \ldots, t_{k_a}^a\}$ and the values y_j^a, $j = 0, 1, \ldots, k_a$, which are defined by

$$y_0^a := w_a^l(0) \quad \text{and} \quad y_j^a := w_a^l(t_j^a),\ \forall j \in \{1, 2, \ldots, k_a\},$$

and some additional variables. The main idea is that each point q_a in the interval $[l_a, u_a]$ lies between two points t_k^a and t_{k+1}^a for some index k, $1 \leq k \leq k_a - 1$. Therefore, q_a can be represented as a convex combination of these two points. The value $w_a^l(q_a)$ is then equal to the corresponding convex combination of y_k^a and y_{k+1}^a. That means only the index k and the coefficients of the convex combination have to be determined. We consider the following problem

$$
\begin{aligned}
& q = \sum_{j=1}^{k_a} \mu_a^j \, t_j^a \\
& \sum_{j=0}^{k_a-1} \lambda_a^j = 1 \\
& \sum_{j=0}^{k_a} \mu_a^j = 1 \\
& \lambda_a^0 \leq \mu_a^0, \\
& \lambda_a^j \leq \mu_a^j + \mu_a^{j+1}, \quad \forall j \in \{1, 2, \dots, k_a - 1\} \\
& \lambda_a^j \in \{0, 1\}, \qquad \forall j \in \{0, 1, \dots, k_a - 1\} \\
& \mu_a^j \in [0, 1], \qquad \forall j \in \{0, 1, \dots, k_a\}.
\end{aligned}
\tag{7.5}
$$

The variables λ_a^j for $j \in \{1, 2, \dots, k_a - 1\}$ indicate whether q belongs to the interval $[t_j^a, t_{j+1}^a]$. This is the case if and only if $\lambda_a^j = 1$. One can easily prove that for each given number q in $\{0\} \cup [l_a, u_a]$ there exist λ_a and μ_a such that (q, λ_a, μ_a) fulfils (7.5). Moreover, for each triple (q, λ_a, μ_a) that fulfils (7.5), it holds that

$$q \in \{0\} \cup [l_a, u_a]$$

and

$$w_a^l(q) = \sum_{j=0}^{k_a} \mu_a^j y_j^a.$$

Hence, we can formulate (7.4) as the following mixed integer program

$$c_l(S) = \min \sum_{a \in A} \sum_{j=0}^{k_a} y_j^a \, \mu_a^j \tag{7.6}$$

$$
\begin{aligned}
& \sum_{p \in P_c} \sum_{(p,k) \in A} x_{pk}^{pc} = d_c z_S^c, \qquad \forall c \in C \\
& \sum_{p \in P_c} \sum_{(i,p) \in A} x_{ip}^{pc} = 0, \qquad \forall c \in C \\
& \sum_{p \in P_c} \sum_{(c,k) \in A} x_{ck}^{pc} = 0, \qquad \forall c \in C \\
& \sum_{p \in P_c} \sum_{(i,c) \in A} x_{ic}^{pc} = d_c z_S^c, \qquad \forall c \in C
\end{aligned}
$$

$$\begin{aligned}
\sum_{(i,j)\in A} x_{ij}^{pc} - \sum_{(j,k)\in A} x_{jk}^{pc} &= 0, && \forall j \in V\backslash\{p,c\},\ \forall c \in C,\ \forall p \in P_c \\
x_a^{pc} &\geq 0, && \forall a \in A,\ \forall c \in C,\ \forall p \in P_c \\
x &\in Q \\
\sum_{c\in C}\sum_{p\in P_c} x_a^{pc} &= \sum_{j=1}^{k_a} t_j^a\, \mu_a^j, && \forall a \in A \\
\sum_{j=0}^{k_a-1} \lambda_a^j &= 1, && \forall a \in A \\
\sum_{j=0}^{k_a} \mu_a^j &= 1, && \forall a \in A \\
\lambda_a^0 &\leq \mu_a^0, && \forall a \in A \\
\lambda_a^j &\leq \mu_a^j + \mu_a^{j+1}, && \forall a \in A,\ \forall j \in \{1,2,\dots,k_a-1\} \\
\lambda_a^j &\in \{0,1\}, && \forall a \in A,\ \forall j \in \{0,1,\dots,k_a-1\} \\
\mu_a^j &\in [0,1], && \forall a \in A,\ \forall j \in \{0,1,\dots,k_a\}.
\end{aligned}$$

In the end of this section, we describe how to calculate good bounds l_a and u_a for each arc $a \in A$. Choosing good bounds is important because the number of variables λ_a and μ_a in the MIP (7.6) depends on the number of discretization grid points of the interval $[l_a, u_a]$, i.e., by choosing a small interval we may reduce the number of variables of the mixed integer model. We assume in the following that for every arc $a \in A$ the weight function w_a is concave in $[0, +\infty)$. Every function of the type αx^β for some numbers $\alpha > 0$ and $\beta \in [0,1]$ and the sum of several functions of that type are concave. Under this assumption, we prove the following proposition:

Proposition 7.3.3. *If the weight function w_a is concave for every arc $a \in A$, then for each non-empty set $S \subseteq C$ there exists an optimal solution $\bar{x}$ of* (7.1) *such that the flow of good from every producer p to every customer c is unsplitted and acyclic, i.e., there exists a path $\mathcal{P}_{pc}$ from p to c such that*

$$\bar{x}_a^{pc} = 0,\ \forall a \notin \mathcal{P}_{pc}.$$

Proof. Let x be an arbitrary optimal solution of (7.1). Let c^* be an arbitrary customer and p^* be an arbitrary producer in P_{c^*}. Due to the flow conservation constraints, an arc $a \in A$ satisfies $x_a^{p^*c^*} > 0$ if and only if a

belongs to a walk from p^* to c^*, whose value $x_e^{p^*c^*}$ is positive for every arc e in this walk. If there exists at most one such walk, then there is nothing to do and we consider another pair of customer and producer. Otherwise, there exist two different walks $\mathcal{P}_1$ and $\mathcal{P}_2$ from p^* to c^* such that

$$x_a^{p^*c^*} > 0,\ \forall a \in \mathcal{P}_1 \cup \mathcal{P}_2.$$

Denote

$$\mu_i := \min_{a \in P_i} x_a^{p^*c^*},\ i = 1, 2$$

and

$$\chi_a^{ipc} := \begin{cases} 1 & \text{if } p = p^*, c = c^*, \text{ and } a \in \mathcal{P}_i \\ 0 & \text{otherwise,} \end{cases} \quad \text{for } a \in A, c \in C, p \in P_c, i = 1, 2.$$

We consider ξ, x^1, and x^2 defined by

$$\xi := x - \mu_1 \chi^1 - \mu_2 \chi^2,$$

and

$$x^i := \xi + (\mu_1 + \mu_2)\chi^i,\ i = 1, 2.$$

Clearly, x^1 and x^2 are feasible solutions of (7.1). For $i = 1, 2$, the number of walks from p^* to c^*, on which there exists a non-empty flow from p^* to c^* in the solution x^i, is one less than the number of these walks in the original solution x. We want to prove that the cost of x is not smaller than the minimum of the costs of x^1 and x^2. Denote

$$w(x) := \sum_{a \in A} w_a\Big(\sum_{c \in C} \sum_{p \in P_c} x_a^{pc}\Big),$$

$$y_a := \sum_{c \in C} \sum_{p \in P_c} \xi_a^{pc},\ \forall a \in A,$$

and

$$h(\lambda) := \sum_{a \in \mathcal{P}_1 \setminus \mathcal{P}_2} w_a\big(y_a + (\mu_1 + \mu_2)\lambda\big) + \sum_{a \in \mathcal{P}_2 \setminus \mathcal{P}_1} w_a\big(y_a + (\mu_1 + \mu_2)(1 - \lambda)\big).$$

We have then

$$\begin{aligned} w(x^1) - w(x) = & \sum_{a\in\mathcal{P}_1\setminus\mathcal{P}_2} w_a(y_a+\mu_1+\mu_2) + \sum_{a\in\mathcal{P}_2\setminus\mathcal{P}_1} w_a(y_a) \\ & - \sum_{a\in\mathcal{P}_1\setminus\mathcal{P}_2} w_a(y_a+\mu_1) - \sum_{a\in\mathcal{P}_2\setminus\mathcal{P}_1} w_a(y_a+\mu_2) \\ = & \; h(1) - h\left(\frac{\mu_1}{\mu_1+\mu_2}\right) \end{aligned}$$

and

$$\begin{aligned} w(x^2) - w(x) = & \sum_{a\in\mathcal{P}_1\setminus\mathcal{P}_2} w_a(y_a) + \sum_{a\in\mathcal{P}_2\setminus\mathcal{P}_1} w_a(y_a+\mu_1+\mu_2) \\ & - \sum_{a\in\mathcal{P}_1\setminus\mathcal{P}_2} w_a(y_a+\mu_1) - \sum_{a\in\mathcal{P}_2\setminus\mathcal{P}_1} w_a(y_a+\mu_2) \\ = & \; h(0) - h\left(\frac{\mu_1}{\mu_1+\mu_2}\right). \end{aligned}$$

Since the function w_a is concave in $[0,+\infty)$ for every $a \in A$, the function h is concave in $[0,1]$. Hence

$$h\left(\frac{\mu_1}{\mu_1+\mu_2}\right) \geq \min\{h(0), h(1)\}.$$

Therefore, we have

$$w(x) \geq \min\{w(x_1), w(x_2)\}.$$

That means either x_1 or x_2 is an optimal solution of (7.1). Assuming that x_1 is an optimal solution of (7.1), we repeat the above process for x_1. This will be repeated until we finally obtain a solution $\bar{x}$ which is unsplitted, i.e., for each customer c and each producer $p \in P_c$ there exists a walk $\bar{\mathcal{P}}_{pc}$ such that

$$\bar{x}_a^{pc} = 0, \; \forall a \notin \bar{\mathcal{P}}_{pc}.$$

If the walk $\bar{\mathcal{P}}_{pc}$ contains circles, we remove them one by one and set the value $\bar{x}_a^{pc}$ to 0 for every arc a in the removed circles. During the elimination the solution that we obtain is still feasible and its cost is not increasing since the weight functions are monotonically increasing, i.e., it remains optimal. After some steps we obtain a path $\mathcal{P}_{pc}$ from p to c from the walk $\bar{\mathcal{P}}_{pc}$. $\square$

Proposition 7.3.3 helps to determine a good interval $[l_a, u_a]$ for each arc $a \in A$ and eliminate superfluous variables. For each arc $a \in A$, we denote $\mathcal{C}_a$ as the set of all customers c in C such that there exists a path $\mathcal{P}$ from some producer $p \in P_c$ to c that contains a. For an arbitrary non-emptyset $S \subseteq C$, let $\bar{x}$ be an optimal solution of (7.1) as described in Proposition 7.3.3. We have then

$$\bar{x}_a^{pc} = 0,\ \forall c \in C,\ \forall p \in P_c,\ \forall a \in A : \mathcal{C}_a = \emptyset,$$

$$\sum_{c \in C} \sum_{p \in P_c} \bar{x}_a^{pc} \leq \sum_{c \in \mathcal{C}_a} d_c$$

and

$$\sum_{c \in C} \sum_{p \in P_c} \bar{x}_a^{pc} = 0 \vee \sum_{c \in C} \sum_{p \in P_c} \bar{x}_a^{pc} \geq \min_{c \in \mathcal{C}_a} d_c.$$

Therefore, we can eliminate all arcs $a \in A$ with $\mathcal{C}_a = \emptyset$ and for the remaining edges we can choose

$$l_a = \min_{c \in \mathcal{C}_a} d_c \tag{7.7}$$

and

$$u_a = \sum_{c \in \mathcal{C}_a} d_c. \tag{7.8}$$

Moreover, for every arc $a \in A$, every customer $c \in C$, and every producer $p \in P_c$, if there does not exist a path from p to c that contains a, we can also eliminate the variable x_a^{pc}. The bounds above are independent of the coalition S. Using this we can create the same discretization grid for both calculating the cost of every coalition and solving the separation problem. One can easily show that, if the capacity for the good which is transmitted through an arc a in the optimal solution $\bar{x}$ of (7.1) is non-zero, then the following numbers are its lower and upper bounds

$$l_a(S) = \min_{c \in \mathcal{C}_a \cap S} d_c \tag{7.9}$$

$$u_a(S) = \sum_{c \in \mathcal{C}_a \cap S} d_c. \tag{7.10}$$

Clearly, we have $u_a = u_a(C)$ and $l_a = l_a(C)$.

7.4 Piecewise Linear Approximation

There are many papers where piecewise linear approximation of concave (convex) functions is used. But to the best of our knowledge, none of them answers the question that we are interested in, namely, how one can construct a piecewise linear approximation of positive, concave functions, especially functions of type ax^b with $a > 0$ and $b \in (0, 1)$, with an arbitrary small *relative error* in any given positive interval explicitly. For functions of this type, we present a very simple, efficient algorithm.

Given a positive, concave function w in an interval $[l, +\infty)$, a number $u > l$, and two numbers $\alpha_2 > \alpha_1 > -1$, we want to determine a grid $l = t_1 < t_2 < \cdots < t_k = u$ and non-negative values y^j, $j \in \{1, 2, \ldots, k\}$, such that the piecewise linear function $\bar{w}$ which is linear in each interval $[t_j, t_{j+1}]$ for every $j \in \{1, 2, \ldots, k-1\}$ with

$$\bar{w}(t_j) = y_j, \ \forall j \in \{1, 2, \ldots, k\},$$

satisfies

$$(1 + \alpha_1)w(q) \leq \bar{w}(q) \leq (1 + \alpha_2)w(q), \ \forall q \in [l, u]. \tag{7.11}$$

Proposition 7.4.1. *Let w be a positive, concave function in $[l, +\infty)$ with some finite number l, α_2 be a real number that satisfies $\alpha_2 > -1$, $l = s_1 < s_2 < \cdots < s_k = u$ be an arbitrary finite grid, and $\bar{w}$ be a function that is linear in each interval $[s_j, s_{j+1}]$ for $1 \leq j \leq k - 1$ and satisfies*

$$\bar{w}(s_j) = (1 + \alpha_2)w(s_j), \ \forall j \in \{1, 2, \ldots, k\}. \tag{7.12}$$

Denote

$$\theta(q) := \frac{\bar{w}(q) - w(q)}{w(q)}$$

and

$$\alpha_1 := \min_{j \in \{1,2,\ldots,k-1\}} \min_{p \in [s_j, s_{j+1}]} \theta(p).$$

Then there holds

$$(1 + \alpha_1)w(q) \leq \bar{w}(q) \leq (1 + \alpha_2)w(q), \ \forall q \in [l, u].$$

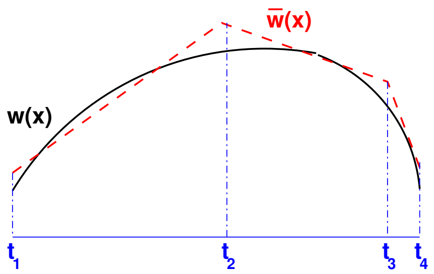

Figure 7.3: Piecewise linear approximation

Proof. Let j be an arbitrary index in $\{1, 2, \ldots, k-1\}$. Since w is positive and concave in $[s_j, s_{j+1}]$, $\bar{w}$ is linear in that interval, and because of (7.12), there holds for any number $\lambda \in [0, 1]$

$$\begin{aligned}\theta\big(\lambda s_j + (1-\lambda)s_{j+1}\big) &= \frac{\bar{w}\big(\lambda s_j + (1-\lambda)s_{j+1}\big)}{w\big(\lambda s_j + (1-\lambda)s_{j+1}\big)} - 1 \\ &\le \frac{\lambda \bar{w}(s_j) + (1-\lambda)\bar{w}(s_{j+1})}{\lambda w(s_j) + (1-\lambda)w(s_{j+1})} - 1 \\ &= \frac{\lambda(1+\alpha_2)w(s_j) + (1-\lambda)(1+\alpha_2)w(s_{j+1})}{\lambda w(s_j) + (1-\lambda)w(s_{j+1})} - 1 \\ &= \alpha_2,\end{aligned}$$

i.e.,

$$\theta(q) \le \alpha_2, \ \forall q \in [s_j, s_{j+1}].$$

Hence, we have

$$\alpha_1 \le \theta(q) \le \alpha_2, \ \forall q \in [s_j, s_{j+1}],$$

i.e.,

$$(1+\alpha_1)w(q) \le \bar{w}(q) \le (1+\alpha_2)w(q), \ \forall q \in [s_j, s_{j+1}]. \qquad \square$$

Based on the above result, we have the following algorithm to determine a function $\bar{w}$ such that (7.11) holds.

Algorithm 7.4.2. *Piecewise linear approximation*

Input: *Two numbers $\alpha_2 > \alpha_1 > -1$, a finite interval $[l, u]$, $l < u$, and a function w that is positive and concave in $[l, +\infty)$.*

Output: *A grid $l = t_1 < t_2 < \cdots < t_k = u$ and a function $\bar{w}$ defined in $[l, u]$ that is positive, linear in each interval $[t_j, t_{j+1}]$, $j = 1, 2, \ldots, k-1$,* *and satisfies*

$$(1+\alpha_1)w(q) \le \bar{w}(q) \le (1+\alpha_2)w(q),\ \forall q \in [l,u].$$

1. *Set* $k = 1$ *and* $s_1 = l$.
2. *Set* $t_k = s_k$ *and* $y_k = (1+\alpha_2)w(t_k)$.
3. *Set* $k := k+1$ *and find a number* $s_k > t_{k-1}$ *such that*
$$\min_{q\in[t_{k-1},s_k]} \frac{f_k(q)}{w(q)} \ge 1+\alpha_1, \tag{7.13}$$
where f_k *is the linear function satisfying* $f_k(t_{k-1}) = y_{k-1}$ *and* $f_k(s_k) = (1+\alpha_2)w(s_k)$.
4. *If* $s_k < u$, *then go to 2.*
5. *Set* $t_k = u$ *and* $y_k = f_k(t_k)$.
6. *Define* $\bar{w}$ *as the function that is linear in each interval* $[t_j, t_{j+1}]$, $j = 1, 2, \ldots, k-1$, *and satisfies*
$$\bar{w}(t_j) = y_j,\ \forall j \in \{1, 2, \ldots, k\}.$$

The main part of Algorithm 7.4.2 is to find s_k such that (7.13) holds. We now consider the case where $w(q) = aq^b$ for some numbers $a > 0$, $b \in (0,1)$, and $l > 0$. The requirement $l > 0$ is needed since one cannot approximate the function $w(q)$ with an arbitrary small given relative error in the neighbourhood of 0 by piecewise linear functions. Since with

$$(1+\alpha_1)w(q) \le \bar{w}(q) \le (1+\alpha_2)w(q),\ \forall q \in [l,u],$$

there also holds

$$(1+\alpha_1)\lambda w(q) \le \lambda\bar{w}(q) \le (1+\alpha_2)\lambda w(q),\ \forall \lambda > 0,\ \forall q \in [l,u],$$

we can assume that $a = 1$ without loss of generality. Given $k > 1$, $t_{k-1} \in [l,u)$, and $y_{k-1} = (1+\alpha_2)t_{k-1}^b$, we want to find a number $s_k > t_{k-1}$ such that

$$\min_{q\in[t_{k-1},s_k]} \frac{f(q)}{q^b} \ge 1+\alpha_1, \tag{7.14}$$

where f is the linear function satisfying

$$f(t_{k-1}) = y_{k-1} \text{ and } f(s_k) = (1+\alpha_2)s_k^b. \tag{7.15}$$

In the following, we show that the largest number s_k which satisfies (7.14) is equal to the product of t_{k-1} and a number which only depends on the exponent b and the parameters α_1 and α_2. From (7.15) it follows that

$$\begin{aligned}
f(q) &= \frac{f(s_k)-f(t_{k-1})}{s_k - t_{k-1}} q + \frac{f(t_{k-1})s_k - f(s_k)t_{k-1}}{s_k - t_{k-1}} \\
&= (1+\alpha_2)\left(\frac{s_k^b - t_{k-1}^b}{s_k - t_{k-1}} q + \frac{t_{k-1}^b s_k - s_k^b t_{k-1}}{s_k - t_{k-1}}\right) \\
&= (1+\alpha_2)\left(t_{k-1}^{b-1}\frac{\left(\frac{s_k}{t_{k-1}}\right)^b - 1}{\frac{s_k}{t_{k-1}} - 1} q + t_{k-1}^b \frac{\frac{s_k}{t_{k-1}} - \left(\frac{s_k}{t_{k-1}}\right)^b}{\frac{s_k}{t_{k-1}} - 1}\right).
\end{aligned}$$

Denote $\gamma := t_{k-1}^{b-1}$ and $z := \dfrac{s_k}{t_{k-1}} > 1$, we have then

$$f(q) = (1+\alpha_2)\gamma\left(\frac{z^b - 1}{z-1} q + t_{k-1}\frac{z - z^b}{z-1}\right).$$

We now consider the function $g(q) := \dfrac{f(q)}{q^b}$ in $[t_{k-1}, s_k]$. We have

$$\begin{aligned}
g'(q) &= \frac{f'(q)q - bf(q)}{q^{b+1}} \\
&= \frac{(1+\alpha_2)\gamma}{q^{b+1}}\left((1-b)\frac{z^b-1}{z-1} q - bt_{k-1}\frac{z - z^b}{z-1}\right).
\end{aligned}$$

Therefore, since $q^* \geq t_{k-1} > 0$, $g'(q^*) = 0$ iff

$$(1-b)\frac{z^b - 1}{z-1} q^* - bt_{k-1}\frac{z - z^b}{z-1} = 0,$$

which is equivalent to

$$q^* = t_{k-1}\frac{b}{1-b}\frac{z - z^b}{z^b - 1}.$$

It holds that $q^* \in (t_{k-1}, s_k)$. Since $\alpha_2 > -1$, $\gamma > 0$, $b \in (0,1)$, $z > 1$, and $t_{k-1} > 0$, there hold

$$g'(q) < 0, \ \forall q \in [t_{k-1}, q^*)$$

and

$$g'(q) > 0, \ \forall q \in (q^*, s_k].$$

This means q^* is the minimizer of g in $[t_{k-1}, s_k]$. We have

$$\begin{aligned}
f(q^*) &= (1+\alpha_2)\gamma\left(\frac{z^b-1}{z-1}\cdot t_{k-1}\frac{b}{1-b}\frac{z-z^b}{z^b-1} + t_{k-1}\frac{z-z^b}{z-1}\right)\\
&= (1+\alpha_2)\gamma\left(t_{k-1}\frac{b}{1-b}\frac{z-z^b}{z-1} + t_{k-1}\frac{z-z^b}{z-1}\right)\\
&= (1+\alpha_2)\gamma t_{k-1}\frac{1}{1-b}\frac{z-z^b}{z-1}\\
&= (1+\alpha_2)t_{k-1}^b\frac{1}{1-b}\frac{z-z^b}{z-1}
\end{aligned}$$

and hence

$$\begin{aligned}
\frac{f(q^*)}{q^{*b}} &= \frac{1+\alpha_2}{b^b(1-b)^{1-b}}\left(\frac{z^b-1}{z-z^b}\right)^b\frac{z-z^b}{z-1}\\
&= \frac{1+\alpha_2}{b^b(1-b)^{1-b}}\left(\frac{z-1}{z-z^b}-1\right)^b\frac{z-z^b}{z-1}.
\end{aligned}$$

Therefore, (7.14) holds iff

$$\frac{1+\alpha_2}{b^b(1-b)^{1-b}}\left(\frac{z-1}{z-z^b}-1\right)^b\frac{z-z^b}{z-1} \geq 1+\alpha_1.$$

Since $0 < b < 1$ and $\alpha_2 > -1$, this is equivalent to

$$v(z) := \left(\frac{z-1}{z-z^b}-1\right)^b\frac{z-z^b}{z-1} \geq b^b(1-b)^{1-b}\frac{1+\alpha_1}{1+\alpha_2}.$$

One can prove that for each number $b \in (0,1)$ the function $\frac{z-1}{z-z^b}$ is strictly decreasing in $(1,+\infty)$ and its value in that interval belongs to $\left(1, \frac{1}{1-b}\right)$, and the function $\frac{(y-1)^b}{y}$ is strictly increasing in $\left(1, \frac{1}{1-b}\right)$. Hence, the function v is strictly decreasing in $(1,+\infty)$. Moreover, using the L'Hopital's rule, we have that

$$\lim_{z\to 1^+}\frac{z-1}{z-z^b} = \frac{1}{1-b} \quad \text{and} \quad \lim_{z\to+\infty}\frac{z-1}{z-z^b} = 1.$$

From these and since

$$\lim_{y\to 1^+}\frac{(y-1)^b}{y} = 0 \quad \text{and} \quad \lim_{y\to \frac{1}{1-b}^-}\frac{(y-1)^b}{y} = b^b(1-b)^{1-b}$$

it follows that

$$\lim_{z\to 1^+} v(z) = b^b(1-b)^{1-b} \quad \text{and} \quad \lim_{z\to +\infty} v(z) = 0.$$

On the other hand, since $\alpha_2 > \alpha_1 > -1$ and $b \in (0,1)$, it holds that

$$b^b(1-b)^{1-b}\frac{1+\alpha_1}{1+\alpha_2} \in \big(0, b^b(1-b)^{1-b}\big).$$

Therefore, there exists exactly one solution $z^* \in (1, +\infty)$ of the following equation

$$v(z) = b^b(1-b)^{1-b}\frac{1+\alpha_1}{1+\alpha_2}. \tag{7.16}$$

We have that $s_k = z^* t_{k-1}$ is the largest number such that (7.14) holds. Moreover, the inequality (7.14) still holds for $s_k = zt_{k-1}$ with every number $z \in (1, z^*]$. It is quite interesting that the optimal factor z^* does not depend on t_{k-1} but only on the exponent b and the parameters α_1 and α_2. We have the following result:

Proposition 7.4.3. *Given $u > l > 0$ and two real numbers $\alpha_2 > \alpha_1 > -1$. For $w(q) = aq^b$ with $a > 0$, $b \in (0,1)$, let z^* be the solution of (7.16) and z be an arbitrary number in $(1, z^*]$. Define the grid $\{t_1, t_2, \ldots, t_k\}$ as follows*

$$t_1 = l, \quad t_k = u, \quad t_k \le zt_{k-1},$$

and

$$t_{i+1} = zt_i, \ \forall i = 1, 2, \ldots, k-2.$$

Let $\bar{w}$ be the piecewise linear function which is linear in each interval $[t_i, t_{i+1}]$ for $i = 1, 2, \ldots, k-1$ and satisfies

$$\bar{w}(t_i) = (1+\alpha_2)w(t_i), \ \forall i = 1, 2, \ldots, k,$$

then there holds

$$(1+\alpha_1)w(q) \le \bar{w}(q) \le (1+\alpha_2)w(q).$$

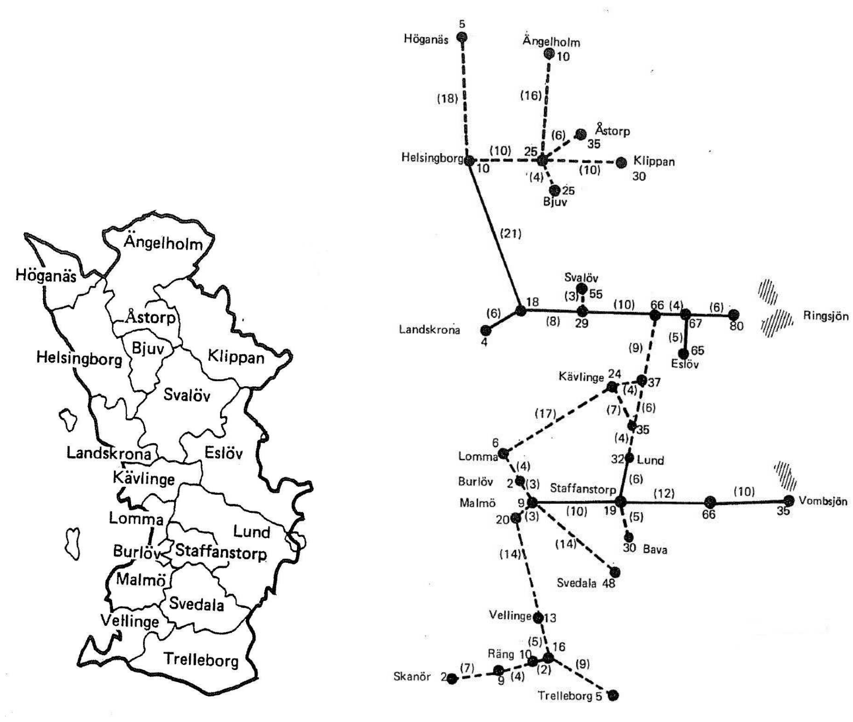

Figure 7.4: The water transmission network

7.5 Cost Allocation in Water Resources Development: A Case Study of Sweden

Water is one of the most important resources. Development and management of water resources are grand issues, which happen all over the world. Such projects are often huge in size and involve many participants. In this thesis, we consider a water supply network of Sweden in the eighties [67]. The Skåne region of southern Sweden consists of eighteen municipalities (see Figure 7.4). In the 1940s several of them banded together to form a regional water supply utility known as the Sydvatten Company. As water demands have grown, the company has been under increasing pressure to increase long-term supply and incorporate outlying municipalities into the system. The question is how the cost should be allocated among the various townships. This problem was already considered by Young et al. [67]. However, due to the absence of computational technology at that time, they were only able to consider a relaxed problem, where the 18 municipalities were grouped into six independent units and a cost allocation problem of 6 players arised. The cost allocation game with 6 players can be written explicitly and easily solvable. We want to reconsider the original game with 18 players using our game theoretical concepts and computational algorithms. Unfortunately, a direct comparison between the two approaches is impossible. Some formulas in [67] are incorrect. Besides, there might be some omitted information, as it is not possible to reproduce the costs reported in [67], even with their formulas.

7.5.1 The Water Resources Development Cost Allocation Game

We formulate the cost allocation problem as a cost allocation game. The set of customers C is the set of all municipalities, $C = \{1, 2, \ldots, 18\}$. Each municipality is considered as a player and each grouping of players is a possible coalition. That means, $N = C$ and $\Sigma = 2^N$. We require only that the price for each player is non-negative, i.e., $P = \mathbb{R}^{18}_{+}$. It remains to define the cost function c.

We follow the cost function description presented in [67] with the corrected formulas. Table 7.1 gives the water demand of every municipality. Bara is included in the municipality of Svedala, while Vellinge, Räng, and Skanör constitute a single municipality. The water supply system includes two lakes (Vombsjön and Ringsjön), one major groundwater aquifer (Alnarp), and other minor on-site sources. The possible routes of a water transmission network are shown in Figure 7.4. The distances between points (in parantheses) and their elevations are also shown in that figure. It is assumed that the pressure at each demand point does not depend on the arrangement by which the water is supplied. This allows us to treat each arc of the transmission network independently. The cost analysis of the network is therefore carried out arc by arc.

The cost of water transmission through a pipe a includes the following components:
Cost of pipelines [Skr]

$$C^1_a = c_1 L_a = (\gamma + \alpha D^{\beta}_a) L_a,$$

cost of pumps [Skr]

$$C^2_a = c_2 f P_a,$$

and cost of electricity [Skr/year]

$$C^3_a = c_3 P_a,$$

Municipality	Water demand x $10^6 \mathrm{m}^3$/year	Municipality	Water demand x $10^6 \mathrm{m}^3$/year
Ängelholm	2.65	Malmö	10.66
Åstorp	0.63	Staffanstorp	2.30
Bjuv	1.36	Svalöv	0.50
Burlöv	1.67		
Eslöv	1.32	Svedala	1.10
Helsingborg	4.60	Bara	0.73
Höganäs	1.48		
Kävlinge	2.74	Trelleborg	1.26
Klippan	0.60		
Landskrona	1.80	Vellinge	2.30
Lomma	1.01	Räng	0.57
Lund	3.53	Skanör	0.25

Bara is included in the municipality of Svedala,
while Vellinge, Räng, and Skanör constitute a single municipality.

Table 7.1: Water demand

where
c_1 is the unit cost of piping [Skr/m],
L_a is the length of pipe [m],
c_2 is the unit cost of a pump [Skr/kW],
f is the safety factor,
c_3 is the unit cost of electricity [Skr/kW year],
P_a is the effective capacity of a pump, $P_a = (9.81/E)Q_a H_a$ [kW],
Q_a is the flow of water through a pipe [m^3/s],
H_a is the required pumping head, $H_a = H_a^0 + I_a L_a$ [m],
H_a^0 is the difference in altitude between origin and destination of pipe [m],
I_a is the hydraulic gradient,
E is the pumping efficiency,
D_a is the pipe diameter [m],
α, β, γ are positive coefficients.

The total annual cost of transmission through an arc a is given by

$$w_a = \left(C_a^1 + \max\{0, C_a^2\}\right)\mathrm{CRF} + \max\{0, C_a^3\},$$

where CRF is the capital recovery factor defined by

$$\mathrm{CRF} = \frac{i(1+i)^n}{(1+i)^n - 1}$$

with the interest rate i and the amortization period in years n. In [67], the calculations of C_a^1, C_a^2 and C_a^3 are incorrect and the authors did not notice that C_a^2 and C_a^3 can be negative, since H_a^0 can be negative. Therefore, we have to take the maximum of these functions and zero. The cost w_a is a function of the pipe diameter D_a, the flow Q_a, the pumping head H_a, and the length of pipe L_a. These factors are related by the Hazen-Williams formula

$$H_a = H_a^0 + 10.7(Cw)^{-1.85} D_a^{-4.87} Q_a^{1.85} L_a,$$

where Cw is the Hazen-Williams coefficient. We have

$$w_a = \begin{cases} C_a^1 \cdot \mathrm{CRF} =: w_a^1, & \text{if } H_a \leq 0 \\ (C_a^1 + C_a^2)\mathrm{CRF} + C_a^3 =: w_a^2, & \text{if } H_a \geq 0. \end{cases}$$

With a given water demand Q_a, we want to calculate the economical pipe diameter D_a^*, i.e., the diameter that minimizes w_a. We consider H_a as a function of D_a. Denote

$$D_a^0 := \left(\frac{4.87\, d_1 d_2}{\alpha\beta\, \mathrm{CRF}} \right)^{1/(\beta+4.87)} Q_a^{2.85/(\beta+4.87)}$$

and

$$D_a^1 := \left(\frac{d_2 Q_a^{1.85} L_a}{|H_a^0|} \right)^{1/4.87},$$

where

$$d_1 = (c_2 f \mathrm{CRF} + c_3) 9.81/E \quad \text{and} \quad d_2 = 10.7(Cw)^{-1.85}.$$

We have that $\partial w_a^2 / \partial D_a$ is equal to 0 with $D_a = D_a^0$, negative in $(0, D_a^0)$, and positive in $(D_a^0, +\infty)$. This means that w_a^2 is strictly decreasing in $(0, D_a^0]$ and strictly increasing in $[D_a^0, +\infty)$. If $H_a^0 \geq 0$, then H_a is non-negative for any $D_a > 0$. Therefore, in this case, $w_a = w_a^2$ and the economical pipe diameter D_a^* is equal to D_a^0. If $H_a^0 < 0$, then H_a is equal to 0 with $D_a = D_a^1$, positive in $(0, D_a^1)$, and negative in $(D_a^1, +\infty)$. From this it follows that

$$w_a = \begin{cases} w_a^1, & \text{if } D_a \in [D_a^1, +\infty) \\ w_a^2, & \text{if } D_a \in (0, D_a^1]. \end{cases}$$

On the other hand, we have

$$\operatorname*{argmin}_{D_a \in [D_a^1, +\infty)} \; w_a^1(D_a) = \{D_a^1\}$$

and

$$\operatorname*{argmin}_{D_a \in (0, D_a^1]} w_a^2(D_a) = \big\{ \min\{D_a^0, D_a^1\} \big\}.$$

Therefore, the optimal diameter D_a^* is

$$D_a^* = \begin{cases} D_a^0 & \text{if } H_a^0 \geq 0 \vee (H_a^0 < 0 \wedge D_a^0 \leq D_a^1) \\ D_a^1 & \text{otherwise.} \end{cases}$$

The economical hydraulic gradient I_a^* is obtained as

$$I_a^* = d_2 (D_a^*)^{-4.87} Q_a^{1.85}.$$

We have

$$\begin{aligned} D_a^0 \leq D_a^1 \iff & \left(\frac{4.87 d_1 d_2}{\alpha\beta \mathrm{CRF}} \right)^{1/(\beta+4.87)} Q_a^{2.85/(\beta+4.87)} \leq \left(\frac{d_2 Q_a^{1.85} L_a}{|H_a^0|} \right)^{1/4.87} \\ \iff & Q_a^{4.87-1.85\beta} \leq \left(\frac{\alpha\beta \mathrm{CRF}}{4.87 d_1} \right)^{4.87} d_2^{\beta} \left(\frac{L_a}{|H_a^0|} \right)^{\beta+4.87}. \end{aligned}$$

The parameters are determined from the Swedish data as follows: $\alpha = 477$ Skr, $\beta = 1.60$, $\gamma = 150$ Skr, $E = 0.63$, $Cw = 100$, $f = 1.33$, $\mathrm{CRF} = 0.0872$, $i = 0.06$, $n = 20$ years, $c_2 = 1893$ Skr/kW, $c_3 = 613$ Skr/kW yr. The results are:

$$D_a^0 \leq D_a^1 \iff Q_a \leq 2.16 \left(\frac{L_a}{10^3 |H_a^0|} \right)^{3.39} =: \bar{q}_a,$$

- If $H_a^0 \geq 0 \vee (H_a^0 < 0 \wedge Q_a \leq \bar{q}_a)$:

$$D_a^* = 1.115 Q_a^{0.44}, \; I_a^* = 1.255 Q_a^{-0.295} \times 10^{-3},$$

$$w_a = w_a^2(D_a^*) = 13.08 L_a + 12964 H_a^0 Q_a + 65.78 L_a Q_a^{0.704}.$$

- If $H_a^0 < 0 \wedge Q_a > \bar{q}_a$:

$$D_a^* = 1.115 \bar{Q}_a^{0.06} Q_a^{0.38}$$

$$w_a = w_a^1(D_a^*) = 13.08 L_a + 49.5 \bar{q}_a^{0.096} L_a Q_a^{0.608}.$$

The water demands are given in $10^6\text{m}^3/\text{year}$, therefore we should convert the flow unit in the above formulas from m^3/s to $10^6\text{m}^3/\text{year}$. Let $\tilde{Q}_a$ be the flow of water in millions of cubic meters per year corresponding to Q_a, i.e., $Q_a\,\text{m}^3/\text{s} = \tilde{Q}_a\,10^6\text{m}^3/\text{year}$, we have $Q_a = 0.03171\tilde{Q}_a$. Denote

$$\tilde{q}_a := \frac{\bar{q}_a}{0.03171} = 68.12\left(\frac{L_a}{10^3|H_a^0|}\right)^{3.39},$$

then the optimal cost of pipe a for a positive demand $\tilde{Q}_a$ ($10^6\text{m}^3/\text{year}$) is

$$w_a(\tilde{Q}_a) = \begin{cases} 13.08L_a + 4.36\tilde{q}_a^{0.096}L_a\tilde{Q}_a^{0.608} & \text{if } H_a^0 < 0 \wedge \tilde{Q}_a > \tilde{q}_a \\ 13.08L_a + 411H_a^0\tilde{Q}_a + 5.79L_a\tilde{Q}_a^{0.704} & \text{otherwise.} \end{cases}$$

The cost data of treating water at the two lakes were not avaiable and hence could only be estimated by the authors in [67]. However, as we cannot reproduce the cost reported in this paper, we do not know how realistic the ratio between the transmission cost and the water treatment cost is with our corrected formulas. Therefore, for the computation part we only consider the transmission cost. Moreover, since the authors did not give the positions of the ground water sources, we will ignore them as well. This means the question now is, how to allocate the transmission cost from the two lakes Vombsjön and Ringsjön to the municipalities.

The water transmission network forms a directed graph $G = (V, A)$. The cost w_a on each arc a is given by a function depending on the length of this edge, the difference in altitude between origin and destination of the pipe, and the water transmission capacity $\tilde{Q}_a$ ($10^6\text{m}^3/\text{year}$): $w_a(0) = 0$ and for $\tilde{Q}_a > 0$

$$w_a(\tilde{Q}_a) = \begin{cases} 13.08L_a + 4.36\tilde{q}_a^{0.096}L_a\tilde{Q}_a^{0.608} & \text{if } H_a^0 < 0 \wedge \tilde{Q}_a > \tilde{q}_a \\ 13.08L_a + 411H_a^0\tilde{Q}_a + 5.79L_a\tilde{Q}_a^{0.704} & \text{otherwise.} \end{cases}$$

We approximate the non-linear part of w_a by piecewise linear functions as presented in the previous sections of this chapter. The cost function can be modeled via the multi-comodity flow model (7.1) with a non-linear objective function which can be approximated by the mixed integer program (7.6). Here the discretization error parameters are chosen as $\alpha_1 = -0.001$ and $\alpha_2 = 0.001$. Note that the same discretization grid must be used for both calculating the cost of each coalition and solving the separation problem. The capacity of a good which is transmitted through an edge e can be restricted to the set $\{0\} \cup [l_a, u_a]$ with $0 < l_a \leq u_a$ defined by (7.7) and (7.8). We now construct an appropriate discretization grid for the piecewise linearization. We only consider the case where $H_a^0 < 0$ and $\tilde{q}_a \in (l_a, u_a]$. Other cases can be treated similarly. Due to Proposition 7.4.3, to achieve a piecewise linear approximation with the lower and upper relative errors α_1 and α_2, we can choose for each arc a the discretization grid $\{t_1^a, t_2^a, \ldots, t_{l_a}^a, \ldots, t_{k_a}^a\}$ as follows

$$t_1^a = l_a, \quad t_{l_a}^a = \tilde{q}_a, \quad t_{l_a}^a \leq z_1^* t_{l_a-1}^a,$$

$$t_{i+1}^a = z_1^* t_i^a, \ \forall i = 1, 2, \ldots, l_a - 2,$$

$$t_{k_a}^a = u_a, \quad t_{k_a}^a \leq z_2^* t_{k_a-1}^a,$$

and

$$t_{j+1}^a = z_2^* t_j^a, \ \forall j = l_a, l_a + 1, \ldots, k_a - 2,$$

where $z_1^* = 1.315$ and $z_2^* = 1.295$. The choice of the increments z_1^* and z_2^* for piecewise linearizing the functions $q^{0.704}$ and $q^{0.608}$ is based on Proposition 7.4.3. Let c_l denote the piecewise linear approximation of the original cost function c defined by (7.6), where

$$y_j^a = (1 + \alpha_2) w_a(t_j^a), \ \forall a \in A, \ \forall j \in \{0, 1, \ldots, k_a\}.$$

Due to Proposition 7.3.1 and Proposition 7.4.3, we have that

$$(1 + \alpha_1)c(S) \leq c_l(S) \leq (1 + \alpha_2)c(S), \ \forall \emptyset \neq S \subsetneq N.$$

The cost allocation game $\Gamma_l := (N, c_l, \mathbb{R}_+^N, 2^N)$ is a well approximation of the the water resources development cost allocation game $\Gamma := (N, c, \mathbb{R}_+^N, 2^N)$.

7.5.2 The Separation Problem

We only consider the separation problem for finding the (f, r)-least core of Γ_l. The separation problem for finding the f-nucleolus of Γ_l is similar. In order to calculate the (f, r)-least core of Γ_l, we use Algorithm 4.2.3 and have to solve the separation problem in every loop. The separation problem for (x^*, ε^*) is to find a coalition $T \in 2^N \backslash \{\emptyset, N\}$ such that (x^*, ε^*) violates the constraint

$$x^*(T) + \varepsilon^* f(T) \leq c_l(T). \tag{7.17}$$

To do so, we solve the following optimization problem

$$\max_{\emptyset \neq S \subsetneq N} x^*(S) + \varepsilon^* f(S) - c_l(S). \tag{7.18}$$

If the optimal value is non-positive then the algorithm terminates and we obtain the (f,r)-least core of Γ. Otherwise, there exists a coalition T which violates the constraint (7.17). We choose $f(S) = \alpha + \beta|S| + \gamma c_l(S)$ for some numbers $\alpha, \beta, \gamma \geq 0$ satisfying $\alpha + \beta + \gamma > 0$. Assume that the pair (x^*, ε^*) is not arbitrary but satisfies $\gamma\varepsilon^* \leq 1$. This assumption holds for every separation problem of Algorithm 4.2.3 because of Proposition 2.3.4. Moreover, due to Section 4.2.4, the optimization problem (7.18) can be reformulated as the following integer program

$$
\begin{aligned}
\max_{(\xi,\rho,z)} \ & \alpha\varepsilon^* + \sum_{i \in N}(x_i^* + \beta\varepsilon^*)z_i + (\gamma\varepsilon^* - 1)\min \sum_{a \in A}\sum_{j=0}^{k_a} y_j^a\, \mu_a^j && && (7.19)\\
& \sum_{p \in P_c}\sum_{(p,k)\in A} x_{pk}^{pc} = d_c z_S^c, && \forall c \in C\\
& \sum_{p \in P_c}\sum_{(i,p)\in A} x_{ip}^{pc} = 0, && \forall c \in C\\
& \sum_{p \in P_c}\sum_{(c,k)\in A} x_{ck}^{pc} = 0, && \forall c \in C\\
& \sum_{p \in P_c}\sum_{(i,c)\in A} x_{ic}^{pc} = d_c z_S^c, && \forall c \in C\\
& \sum_{(i,j)\in A} x_{ij}^{pc} - \sum_{(j,k)\in A} x_{jk}^{pc} = 0, && \forall j \in V \backslash \{p,c\},\ \forall c \in C,\ \forall p \in P_c\\
& x_a^{pc} \geq 0, && \forall a \in A,\ \forall c \in C,\ \forall p \in P_c\\
& x \in Q\\
& \sum_{c \in C}\sum_{p \in P_c} x_a^{pc} = \sum_{j=1}^{k_a} t_j^a\, \mu_a^j, && \forall a \in A\\
& \sum_{j=0}^{k_a-1} \lambda_a^j = 1, && \forall a \in A\\
& \sum_{j=0}^{k_a} \mu_a^j = 1, && \forall a \in A\\
& \lambda_a^0 \leq \mu_a^0, && \forall a \in A\\
& \lambda_a^j \leq \mu_a^j + \mu_a^{j+1}, && \forall a \in A,\ \forall j \in \{1,2,\ldots,k_a-1\}\\
& \lambda_a^j \in \{0,1\}, && \forall a \in A,\ \forall j \in \{0,1,\ldots,k_a-1\}\\
& \mu_a^j \in [0,1], && \forall a \in A,\ \forall j \in \{0,1,\ldots,k_a\}\\
& z \in \{0,1\}^{|N|} \backslash \{\mathbf{0},\mathbf{1}\}.
\end{aligned}
$$

The variables z_i, $i \in N$, correspond to a coalition $S \subseteq N$, z_i is equal to 1 if the player i belongs to S and 0 otherwise. Other variables and constraints come from the integer program (7.6), which models the cost function. A violated constraint exists iff the optimal value is larger than 0. In that case, one can find a feasible solution $(\bar{\xi}, \bar{\rho}, \bar{z})$ of (7.19) with a positive objective function value. Define $T := \{i \in N \mid \bar{z}_i = 1\}$, then (x^*, ε^*) violates the constraint (7.17). We can use the primal and dual heuristics presented in Section 4.2.5 in order to find violated coalitions faster and have a good stopping criterion in each separation step.

7.5.3 Computational Results

In this section, we compare several prices for the water resources development cost allocation game. The first one is the proportional price vector p^1 where the total cost is allocated proportionally to the individual cost:

$$p_i^1 = \frac{c(\{i\})}{\sum_{j \in N} c(\{j\})} c(N), \ \forall i \in N.$$

This price reflects the demand and the distance to the sources of each player, however it does not consider the possibility that players can form coalitions. Consequently, some players prefer to form their own coalition than to accept the proportional price. The second price vector that we considered is the c-nucleolus price. We also consider two (f, r)-least core prices with $(f, r) = (c, p^1)$ and $(f, r) = (c, p^2)$, where p^2 denotes the efficient price vector which has a uniform unit price, i.e., each player has to pay the same transmission cost for a cubic meter:

$$p^2(N) = c(N) \ \text{ and } \frac{p_i^2}{d_i} = \frac{p_j^2}{d_j}, \ \forall i, j \in N.$$

Choosing p^2 as the reference price vector makes sense, since several governments want to keep the living condition equal. Though using p^2 as the price vector will cause unrest among the municipalities. The reason is that the price vector p^2 clearly discriminates the municipalities which are close to the two lakes. They have to pay much more than they should in order to subsidize the faraway municipalities. With the price vector p^2, there are at least seven coalitions with a relative loss (c-loss) of more than 40% compared to their costs. The (c, p^2)-least core price is a compromise between having a uniform unit price and keeping the coalitional stability.

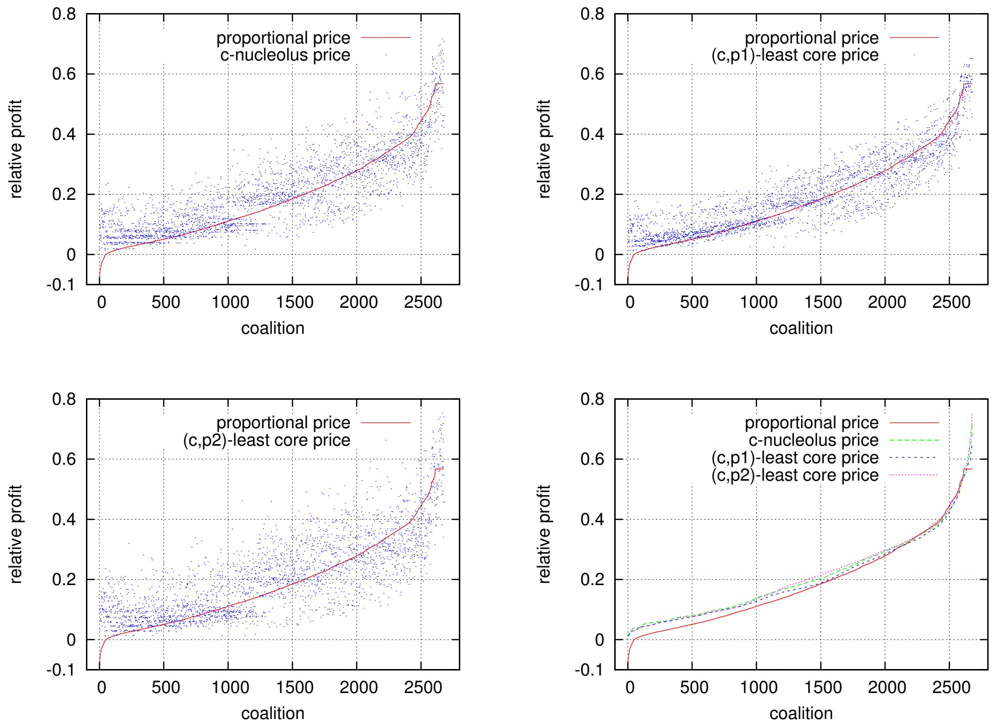

Figure 7.5: Comparison of four price vectors

In order to compare the above four prices vector, we consider their fairness distribution diagram based on a pool of 2678 essential coalitions which contain coalitions having the worst relative profits with these prices (see Chapter 5). The coalitions in the pool are sorted in non-decreasing order regarding their relative profits with the proportional price vector. Computational results show that the c-least core radius of our cost allocation game is 0.013835. This means that the core of this game is non-empty. The c-nucleolus, (c, p^1)-least core, and (c, p^2)-least core prices belong to the c-least core and therefore they are stable, since no coalition has a profit by leaving the grand coalition and acting on its own. Each coalition has a relative profit (c-profit) of at least 1.38% with these prices. On the contrary, the proportional price is unstable, since 8 municipalities in the north, namely, Svalöv, Landskrona, Helsingborg, Höganäs, Bjuv, Klippan, Åstorp, and Ängelholm, have to pay 7.8% more compared to the cost when they form their own coalition and build their pipe system themselves. This means that for this coalition leaving the grand coalition and building its pipe system itself is obviously a better solution than accepting the price vector p^1.

Figure 7.5 compares the four prices. The first three pictures plot the relative profits $\frac{c(S)-x(S)}{c(S)}$ of every coalition S in the pool. The last picture plots also these c-profits but the values are sorted in non-decreasing order, i.e., a point k in the horizontal axis does not represent the k-th. coalition in the pool but the coalitions which have the k-th. smallest c-profits with the four prices.

Table 7.2 presents the unit prices and the ratios to the individual costs of the municipalities with the four considered price vectors. The ratio between the maximal and minimal unit prices of the (c, p^2)-least core price is the smallest. While the (c, p^1)-least core price is the (lexicographically) nearest price vector to the proportional price vector p^1 among the three game theory-based prices.

In our opinion, the three game theory-based prices are significantly better in terms of fairness and coalitional stability than the proportional price vector, which is unfair for the above mentioned coalition consisting of almost half of the players. But the decision which game theory-based price among the three ones should be chosen depends on the additional goals of each specific cost allocation problem beside fairness. If the goal is to have a price vector which is as proportional as possible to the individual costs, then the (c, p^1)-least core price is the most suitable. If the allocated price vector should charge the players as equally as possible for one unit, then the (c, p^2)-least core price is preferred. And if the decision maker is just interested in coalitional fairness, then he should choose the c-nucleolus price.

In [67], the authors assort the 18 municipalities into six groups and solve the problem of six players since in their opinion "to develop the costs for each of the $2^{18} - 1$ possible groupings of the 18 municipalities would be impractical and unrealistic". With the modern techniques, it takes only 1987, 713, and 594 seconds respectively in order to calculate the c-nucleolus, the (c, p^1)-least core, and the (c, p^2)-least core of the original cost allocation game with 18 players respectively. The computations were done on a PC with an Intel Core2 Quad 2.83GHz processor and 16GB RAM. CPLEX 11.2 was used as linear and integer program solver.

	Proportional price		c-nucleolus price		(c, p^1)-least core price		(c, p^2)-least core price	
	Price per 10^6m^3	$\frac{\text{Price}}{\text{Individual cost}}$	Price per 10^6m^3	$\frac{\text{Price}}{\text{Individual cost}}$	Price per 10^6m^3	$\frac{\text{Price}}{\text{Individual cost}}$	Price per 10^6m^3	$\frac{\text{Price}}{\text{Individual cost}}$
Ängelholm	288582	0.432267	297547	0.445696	283767	0.425056	283767	0.425056
Åstorp	761844	0.432267	497757	0.282425	614863	0.348871	435727	0.24723
Bjuv	396286	0.432267	266686	0.2909	319832	0.348871	238146	0.259768
Burlöv	188232	0.432267	145126	0.333275	181915	0.41776	122398	0.281082
Eslöv	69817	0.432267	159280	**0.986163**	159280	**0.986163**	159280	**0.986163**
Helsingborg	126592	0.432267	103547	0.353577	102169	0.348871	150904	0.515285
Höganäs	391998	0.432267	412834	0.455243	387514	0.427322	387514	0.427322
Kävlinge	80443	0.432267	100205	0.538457	85718	0.460609	94178	0.506072
Klippan	**842303**	0.432267	**626013**	0.321268	**679800**	0.348871	**561438**	0.288128
Landskrona	165022	0.432267	177065	0.463812	155370	0.406983	155370	0.406983
Lomma	310545	0.432267	276350	0.384668	300123	0.41776	238959	0.332623
Lund	89180	0.432267	74171	0.359516	86187	0.41776	93729	0.454318
Malmö	61400	0.432267	78745	0.554372	75475	0.531353	93729	**0.659866**
Staffanstorp	94841	0.432267	100891	0.459842	91658	0.41776	93729	0.427198
Svalöv	326321	0.432267	420216	0.556646	340940	0.451632	340940	0.451632
Svedala+Bara	245610	0.432267	328346	**0.577881**	308623	**0.543169**	308623	0.543169
Trelleborg	424808	0.432267	367265	0.373714	451885	0.45982	336560	0.342469
Vellingen+ Räng+Skanör	215400	0.432267	268206	0.538239	229129	0.45982	270188	0.542216

Table 7.2: Comparison of four price vectors

Chapter 8

Ticket Pricing Problem in Public Transport

The subject of this chapter is the ticket pricing problem in public transport, where passengers share a common infrastructure. Public transport ticket prices are well studied in the economic literature on welfare optimization as well as in the mathematical optimization literature on certain network design problems, see, e.g., the literature survey in [7]. To the best of our knowledge, however, the *fairness* of ticket prices has not been investigated yet. The point is that typical pricing schemes are not related to infrastructure operation costs and, in this sense, favor some users, which do not fully pay for the costs they incur. We will show that in the example of the Dutch IC railway network, the current distance tariff results in a situation where the passengers in the central Randstad region of the country pay over 25% more than the costs they incur, and these excess payments subsidize operations elsewhere. One can argue that this is not fair. We therefore ask whether it is possible to construct ticket prices that reflect operation costs better. Our approach calculates fair prices based on the game theoretical concepts in Chapter 2.

8.1 The Ticket Pricing Problem

The problem of designing a system of fares in public transport has been considered in several publications. See, e.g., [7] for an overview. It is an important issue and must be handled in every public transport company. There are several objectives like revenue, social wellfare, or simplicity/clarity, etc. Our interest is however the fairness, which has not been considered.

In order to apply the framework of Chapter 2, we formulate the ticket pricing problem as a cost allocation game $\Gamma = (N, c, P, 2^N)$ as follows. Consider a railway network as a graph $G = (V, E)$, and let $N \subseteq V \times V$ be a set of origin-destination (OD) pairs, between which passengers want to travel, i.e., we consider each (set of passengers of an) OD-pair as a player. We define the cost $c(S)$ of a coalition $S \subseteq N$ as the minimum expense of a network of railway lines in G that services S. Using the classical line planning model in [8], $c(S)$ can be computed by solving the following integer program

$$\begin{aligned} c(S) :=\; & \min_{(\xi,\rho)} \sum_{(r,f)\in\mathcal{R}\times\mathcal{F}} (c^1_{r,f}\xi_{r,f} + c^2_{r,f}\rho_{r,f}) \\ s.t.\; & \sum_{r\in\mathcal{R}, r\ni e} \sum_{f\in\mathcal{F}} c_{cap} f(m\xi_{r,f} + \rho_{r,f}) \geq \sum_{i\in S} P^i_e,\ \forall e \in E \\ & \sum_{r\in\mathcal{R}, r\ni e} \sum_{f\in\mathcal{F}} f\xi_{r,f} \geq F^i_e,\ \forall (i,e) \in S \times E \\ & \rho_{r,f} - (M-m)\xi_{r,f} \leq 0,\ \forall (r,f) \in \mathcal{R}\times\mathcal{F} \\ & \sum_{f\in\mathcal{F}} \xi_{r,f} \leq 1,\ \forall r \in \mathcal{R} \\ & \xi \in \{0,1\}^{|\mathcal{R}\times\mathcal{F}|},\ \rho \in \mathbb{Z}^{|\mathcal{R}\times\mathcal{F}|}_{\geq 0}. \end{aligned} \tag{8.1}$$

The model assumes that the P_i passengers of each OD-pair i travel on a unique shortest path $\mathcal{P}^i$ (with respect to some distance in space or time) through the network, such that demands P^i_e on capacities of edges e arise, and, likewise, demands F^i_e on frequencies of edges. These demands can be covered by a set $\mathcal{R}$ of possible routes (or lines) in G, which can be operated at a (finite) set of possible frequencies $\mathcal{F}$, and with a minimal and maximal number of wagons m and M in each train. c_{cap} is the capacity of a wagon, $c^1_{r,f}$ and $c^2_{r,f}$, $(r, f) \in \mathcal{R}\times\mathcal{F}$, are cost coefficients for the operation of route r at frequency f. The variable $\xi_{r,f}$ is equal to 1 if route r is operated at frequency f, and 0 otherwise, while variable $\rho_{r,f}$ denotes the number of wagons in addition to m on route r with frequency f. The constraints guarantee sufficient capacity and frequency on each edge, link the two types of route variables, and ensure that each route is operated at a single frequency.

Finally, we define the polyhedron P, which gives conditions on the prices x that the players are asked to pay, as follows. Let (u_{j-1}, u_j), $j =$

$1,\ldots,l$, be OD-pairs such that u_j, $j = 0,\ldots,l$, belong to the travel path $\mathcal{P}^{st}$ associated with some OD-pair (s,t), $u_0 = s$, and $u_l = t$, and let (u,v) be an arbitrary OD-pair such that u and v also lie on the travel path $\mathcal{P}^{st}$ from s to t. We then stipulate that the prices x_i/P_i, which individual passengers of OD-pair i have to pay, must satisfy the monotonicity properties

$$0 \leq \frac{x_{uv}}{P_{uv}} \leq \frac{x_{st}}{P_{st}} \leq \sum_{j=1}^{l} \frac{x_{u_{j-1}u_j}}{P_{u_{j-1}u_j}}. \tag{8.2}$$

Moreover, for a given number $K > 1$, we require that the prices should have the following property

$$\max_{st} \frac{x_{st}}{d_{st}P_{st}} \leq K \min_{st} \frac{x_{st}}{d_{st}P_{st}}, \tag{8.3}$$

where d_{st} is the distance of the route (s,t). This inequality guarantees that the price difference per unit of length, say one kilometer, is bounded by a factor of K.

The tuple $\Gamma = (N, c, P, 2^N)$ defines a cost allocation game to determine prices for using the railway network G. Let x^* be an outcome of the game. Then the ticket price of each passenger of the i-th. OD-pair is x_i^* divided by the number of passengers of this OD-pair.

Let $f = \alpha + \beta|\cdot| + \gamma c$ for some non-negative numbers α, β, and γ satisfying $\alpha + \beta + \gamma > 0$ and r be some reference price vector of Γ. We want to use the (f,r)-least core cost allocation method or its dummy friendly version to allocate the common cost to the passengers. These allocation methods are valid, efficient, scalar multiplicative, symmetric core cost allocation methods and the later is dummy friendly due to Section 2.3.3. In practice, the cost allocation game Γ often does not have any dummy player. In that case, the two methods coincide.

Due to Section 2.3.4, the best choice for the weight function f may be the cost function c, i.e., $\alpha = \beta = 0$ and $\gamma = 1$. As the reference price vector r we can choose the distance price. This is reasonable since when two individual passengers compare their prices with each other, they often base on two factors, namely, the traveling distance and the means of transportation. The distance price gives them the feeling that they are treated equally. If they use the same type of train, then their distance prices depend only on the length of their routes. The distance price of a train type is often given by a piecewise linear function, where the average cost for one length unit is decreasing with respect to the traveling distance. With the distance price, the two passengers have to pay almost the same amount of money for a traveling unit by using the same means of transportation. With this choice of f and r, the (f,r)-least core price is firstly fair in the game theoretical meaning and secondly kept as near the distance price as posible. The (f,r)-least core price is more stable than the distance price, since no group of passengers has to pay much more than their own cost as we will see in Section 8.4. In special case, when the reference vector r belongs to the f-least core of Γ, the (f,r)-least core price and r coincide due to Proposition 2.3.42.

One can also use the f-nucleolus cost allocation method. However, on one hand it does not take into account the direct comparison between two individual players, and on the other hand calculating it is very expensive for large applications like our IC ticket pricing problem.

8.2 Calculating the Cost Function

We recall several heuristics presented in [8] for the integer program (8.1). Obviously, its first constraint can be strengthened as follows

$$\sum_{r\in\mathcal{R},r\ni e}\sum_{f\in\mathcal{F}} f(m\xi_{r,f}+\rho_{r,f}) \geq \left\lceil \frac{\sum_{i\in S} P_e^i}{c_{cap}} \right\rceil,\ \forall e\in E.$$

Given a set S of OD-pairs, let $ld_S(e)$ and $F_S(e)$ denote the traffic load and the required frequency on the edge e for serving all passengers in S, i.e.,

$$ld_S(e)=\sum_{i\in S} P_e^i$$

and

$$F_S(e)=\max_{i\in S} F_e^i.$$

A route r operating at frequency f is called a line (r,f). We use the shorthand notation (r,f,γ) for each line (r,f) with γ wagons. The number $\nu_{r,f,\gamma}$ indicates whether it is needed to operate the route r with γ wagons at the frequency f. If $\nu_{r,f,\gamma}$ is set to 0, then the answer is no. At the beginning we set $\nu_{r,f,\gamma}=1$ for every triple (r,f,γ). There are several tricks to eliminate variables, which are listed belows.

If the traffic load for every edge of a route r is satisfied by a single line (r,f^*) with γ^* wagons, i.e.,

$$c_{cap}f^*\gamma^* \geq \max_{e\in r} ld_S(e),$$

then it is not needed to use more wagons, i.e.,

$$\nu_{r,f^*,\gamma} = 0,\ \forall \gamma > \gamma^*. \tag{8.4}$$

Furthermore, if the frequency requirement for every edge of r is also satisfied by the line (r, f^*), i.e.,

$$f^* \geq \max_{e \in r} F_S(e),$$

then no larger frequency is needed, i.e.,

$$\nu_{r,f,\gamma^*} = 0,\ \forall f > f^*. \tag{8.5}$$

If a line (r, f) is selected, then in order to fulfil the frequency requirement at least $k := \max\{0, F_S(e) - f\}$ other trains of other lines must pass e. Since each train consists of at least m wagons, they transport at least kmc_{cap} passengers. Hence, there remains at most $ld_S(e) - kmc_{cap}$ passengers for line (r, f). That means only $\gamma_e^* := \lceil (ld_S(e) - kmc_{cap})/c_{cap} \rceil$ wagons on edge e is needed for line (r, f). Therefore, we can set

$$\nu_{r,f,\gamma} = 0,\ \forall \gamma > \max_{e \in r} \gamma_e^*. \tag{8.6}$$

Let (r^*, f^*) be a line satisfying the frequency requirement of every edge in r^*. Furthermore, assume that the frequency f^* is sufficient to satisfy the demand of wagons of every edge e in r^*, i.e.,

$$\gamma^* := \max \left\{ \left\lceil \frac{1}{f^*} \max_{e \in r^*} \left\lceil \frac{ld_S(e)}{c_{cap}} \right\rceil \right\rceil, m \right\} \leq M.$$

The line (r^*, f^*) with γ^* wagons can fulfil the demands of every edge in r^*. For each line (r, f) with γ wagons, we denote its cost as $c_{r,f,\gamma}$,

$$c_{r,f,\gamma} = c^1_{r,f} + (\gamma - m)c^2_{r,f}.$$

For any frequency $f > f^*$, if the cost of the line (r^*, f) with m wagons, $c_{r^*,f,m}$, is not smaller than c_{r^*,f^*,γ^*}, then we can replace the line (r^*, f) by the line (r^*, f^*) without increasing the objective value. We set

$$\nu_{r^*,f,\gamma} = 0,\ \forall \gamma. \tag{8.7}$$

Even if $c_{r^*,f,m} < c_{r^*,f^*,\gamma^*}$ we can derive a bound on the number of wagons in the line (r^*, f). If the cost of the line (r^*, f) with γ wagons exceeds c_{r^*,f^*,γ^*}, i.e,

$$\gamma \geq \left\lceil \frac{c_{r^*,f^*,\gamma^*} - c^1_{r^*,f}}{c^2_{r^*,f}} \right\rceil + m,$$

then the line (r^*, f) with γ wagons can be replaced by the line (r^*, f^*) with γ^* wagons in any optimal solution. Hence, we can set

$$\nu_{r^*,f,\gamma} = 0, \ \forall \gamma \geq \left\lceil \frac{c_{r^*,f^*,\gamma^*} - c^1_{r^*,f}}{c^2_{r^*,f}} \right\rceil + m. \tag{8.8}$$

The last variable elimination scheme is based on the so called dominance rule. A line (r, f, γ) is dominated by another line (r, f', γ'), $(f, \gamma) \neq (f', \gamma')$, if $c_{r,f,\gamma} \geq c_{r',f',\gamma'}$ and the capacity and frequency of (r, f, γ) are smaller than the ones of (r', f', γ') or the extra amounts are superfluous, i.e.,

$$f\gamma \leq f'\gamma' \quad \text{or} \quad f'\gamma' c_{cap} + \max\{0, F_S(e) - f'\} m c_{cap} \geq ld_S(e), \ \forall e \in r$$

and

$$f \leq f' \quad \text{or} \quad f' + \max\left\{0, \frac{ld_S(e) - \min\{f'\gamma', f\gamma\} c_{cap}}{M c_{cap}}\right\} \geq F_S(e), \ \forall e \in r.$$

The dominated line (r, f, γ) can be replaced by the line (r', f', γ') in every feasible solution without increasing the objective value. Therefore, we can set for all dominated line (r, f, γ)

$$\nu_{r,f,\gamma} = 0. \tag{8.9}$$

We can also use the idea of the dominance rule to define the domination of two lines (r_1, f_1, γ_1) and (r_2, f_2, γ_2) over a line (r, f, γ). Again, a dominated line can be ignored.

Finally, we eliminate the variables $\xi_{r,f}$ and $\rho_{r,f}$ of the integer program (8.1) for every line (r, f) satisfing

$$\nu_{r,f,\gamma} = 0, \ \forall m \leq \gamma \leq M.$$

For a line (r, f), if the above condition does not hold, then we still can derive an upper bound for $\rho_{r,f}$ as follows

$$\rho_{r,f} \leq \max_{\substack{m \leq \gamma \leq M \\ \nu_{r,f,\gamma} \neq 0}} \gamma - m.$$

8.3 The Separation Problem

In order to calculate the (f,r)-least core of Γ, we use Algorithm 4.2.3 and have to solve thereby the separation problem in every loop. The separation problem for (x^*,ε^*) is to find a coalition $T \in 2^N \backslash \{\emptyset, N\}$ such that (x^*,ε^*) violates the constraint

$$x^*(T) + \varepsilon^* f(T) \leq c(T). \tag{8.10}$$

This can be done by solving the optimization problem

$$\max_{\emptyset \neq S \subsetneq N} x^*(S) + \varepsilon^* f(S) - c(S). \tag{8.11}$$

If the optimal value is non-positive, then the algorithm terminates and gives us the (f,r)-least core of Γ. Otherwise, there exists a coalition T, which violates the constraint (8.10). We choose $f = \alpha + \beta|\cdot| + \gamma c$ for some numbers $\alpha, \beta, \gamma \geq 0$ satisfying $\alpha + \beta + \gamma > 0$. Assume that the pair (x^*,ε^*) is not arbitrary but satisfies $\gamma\varepsilon^* \leq 1$. This assumption holds for every separation problem of Algorithm 4.2.3 because of Proposition 2.3.4. Moreover, due to Section 4.2.4, the optimization problem (8.11) can be reformulated as the following integer program

$$\begin{aligned}
\max_{(\xi,\rho,z)} & \alpha\varepsilon^* + \sum_{i \in N}(x_i^* + \beta\varepsilon^*) z_i + (\gamma\varepsilon^* - 1) \sum_{(r,f) \in \mathcal{R} \times \mathcal{F}} (c^1_{r,f}\xi_{r,f} + c^2_{r,f}\rho_{r,f}) \qquad (8.12)\\
s.t. \quad & \sum_{r \in \mathcal{R}, r \ni e} \sum_{f \in \mathcal{F}} c_{cap} f(m\xi_{r,f} + \rho_{r,f}) - \sum_{i \in N} P^i_e z_i \geq 0, \ \forall e \in E\\
& \sum_{r \in \mathcal{R}, r \ni e} \sum_{f \in \mathcal{F}} f\xi_{r,f} - F^i_e z_i \geq 0, \ \forall (i,e) \in N \times E\\
& \rho_{r,f} - (M-m)\xi_{r,f} \leq 0, \ \forall (r,f) \in \mathcal{R} \times \mathcal{F}\\
& \sum_{f \in \mathcal{F}} \xi_{r,f} \leq 1, \ \forall r \in \mathcal{R}\\
& \xi \in \{0,1\}^{|\mathcal{R} \times \mathcal{F}|}, \ \rho \in \mathbb{Z}^{|\mathcal{R} \times \mathcal{F}|}_{\geq 0}, \ z \in \{0,1\}^{|N|} \backslash \{\mathbf{0}, \mathbf{1}\}.
\end{aligned}$$

The variables z_i, $i \in N$, correspond to a coalition $S \subseteq N$, z_i is equal to 1 if the player i belongs to S and 0 otherwise. Other variables and constraints come from the integer program (8.1), which models the cost function. A violated constraint exists iff the optimal value is larger than 0. If it is positive, then one can find a feasible solution $(\bar{\xi}, \bar{\rho}, \bar{z})$ of (8.12) with a positive objective function value. Define $T := \{i \in N \mid \bar{z}_i = 1\}$, then (x^*,ε^*) violates the constraint (8.10).

For solving the separation problem, we can use the primal and dual heuristics presented in Section 4.2.5 in order to find violated coalitions faster and have a good stopping criterion in each separation step. Moreover, one can exploit the special structure of the problem to eliminate some variables of the integer program (8.12). The idea is to modify the heuristics in Section 8.2. Let $ld(e)$ and $F(e)$ denote the traffic load and the required frequency on the edge e for serving all passengers in N, i.e.,

$$ld(e) = \sum_{i \in N} P_e^i$$

and

$$F(e) = \max_{i \in N} F_e^i.$$

Clearly, the traffic load and the required frequency on each edge e for serving any coalition S are bounded by $ld(e)$ and $F(e)$, respectively. Again, at the beginning, we set $\nu_{r,f,\gamma} = 1$ for each line (r, f, γ). The heuristics (8.4), (8.5), (8.7), and (8.8) with $S = N$ are still true for the integer program (8.12), while the heuristics using the dominance rule must be modified. We apply the heuristics (8.4), (8.5), (8.7), and (8.8) with $S = N$ and can set thereby $\nu_{r,f,\gamma} = 0$ for several lines (r, f, γ). The heuristics using the dominance rule can be modified as follows. For each line (r, f, γ), if there exists another line (r, f', γ') satisfying $c_{r,f,\gamma} \geq c_{r',f',\gamma'}$,

$$f\gamma \leq f'\gamma' \quad \text{or} \quad f'\gamma' c_{cap} \geq ld(e), \ \forall e \in r$$

and

$$f \leq f' \quad \text{or} \quad f' \geq F(e), \ \forall e \in r,$$

then we can replace the line (r, f, γ) by (r, f', γ') in any feasible solution of (8.12) without decreasing the objective value. And, hence, in that case we can set $\nu_{r,f,\gamma} = 0$. Finally, we eliminate the variables $\xi_{r,f}$ and $\rho_{r,f}$ of the integer program (8.12) for every line (r, f) satisfing

$$\nu_{r,f,\gamma} = 0, \ \forall m \leq \gamma \leq M.$$

For a line (r, f), if the above condition does not hold, then we still can derive an upper bound for $\rho_{r,f}$ as follows

$$\rho_{r,f} \leq \max_{\substack{m \leq \gamma \leq M \\ \nu_{r,f,\gamma} \neq 0}} \gamma - m.$$

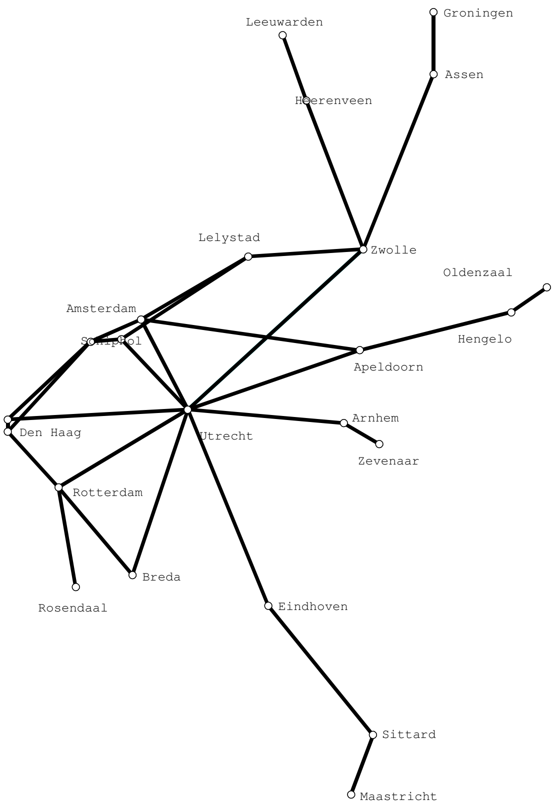

Figure 8.1: The intercity network of the Netherlands

8.4 Ticket Prices for the Dutch IC Network

We now use our ansatz to compute ticket prices for the intercity network of the Netherlands, which is shown in Figure 8.1. Our data is a simplified version of that published in [8], namely, we consider all 23 cities, but reduce the number of OD-pairs to 85 by removing pairs with small demand. However, with $2^{85} - 1$ possible coalitions, the problem is still very large. Since there is only one train type, the distance price depends only on the traveling distance. As reported in [6], the distance price, which has been used by the railway operator NS Reizigers for this network, is piecewise linear depending on the traveling distance, where the average price for one kilometer decreases. However, since the data of this academic example and the real data of NS Reizigers are different, we do not know the coefficients of the distance price function for our application. Hence, instead of using a piecewise linear function, we choose the linear distance price function for pricing. That means each passenger has to pay the same amount of money for one traveling distance, which is called the base price. The distance price of each passenger is then the product of the base price and the traveling distance. The base price is so chosen that the total distance prices cover exactly the common cost, i.e.,

$$\text{base price} = \frac{\text{common cost}}{\text{total traveling kilometers of all passengers}}.$$

The distance price $\bar{x}_i$ of an OD-pair i is the product of the distance price for one passenger in this OD-pair and the number of its passengers. For our (f, r)-least core price, we choose $f = c$ and $r = \bar{x}$. We start with a "pure fairness scenario" where the prices are only required to have the monotonicity property (8.2), i.e., we ignore property (8.3) for the moment. By using Algorithm 4.2.3, we determine the $(c, \bar{x})$-least core, which contains a unique point x^*, and define the $(c, \bar{x})$-*least core ticket price* (lc-price) for each passenger in an OD-pair i as $p_i^* := x_i^*/P_i$.

In order to compare the distance and $(c, \bar{x})$-least core prices, we consider their fairness distribution diagram based on a pool of 7084 essential coalitions (see Chapter 5). The pool contains coalitions which have the worst relative profits with these prices. The coalitions in the pool are sorted in non-decreasing order regarding their relative profits with the distance prices. Figure 8.2 compares the lc-price vector with the distance price vector. The picture on the left side plots the relative profits $\frac{c(S)-x(S)}{c(S)}$ of every coalition S in the pool with $x = x^*$ and $x = \overline{x}$, while the picture on the right side considers only the 100 first coalitions. The picture on the left side of Figure 8.3 plots also these c-profits but the values are sorted in non-decreasing order for both prices x^* and $\overline{x}$, i.e., a point k in the horizontal axis does not represent the k-th. coalition in the pool but the two coalitions which have the k-th. smallest c-profits with the two prices x^* and $\overline{x}$. Note that the core of this particular game is empty and therefore with any price vector there exist coalitions which have to pay more than their costs. The maximum c-loss of any coalition with the lc-prices is a mere 1.1%. This hardly noticeable unfairness is in contrast with the 25.67% maximum c-loss of the distance prices. In fact, there are 10 other coalitions in our pool with losses of more than 20%. Even worse, the coalition with the maximum loss is a large coalition of passengers traveling in the center of the country. It is the coalition of the following 8 OD-pairs: Amsterdam CS – Den Haag HS, Rotterdam CS – Schiphol, Amsterdam CS – Rotterdam CS, Den Haag HS – Rotterdam CS, Roosendaal Grens – Schiphol, Amsterdam CS – Roosendaal Grens, Den Haag HS – Roosendaal Grens,

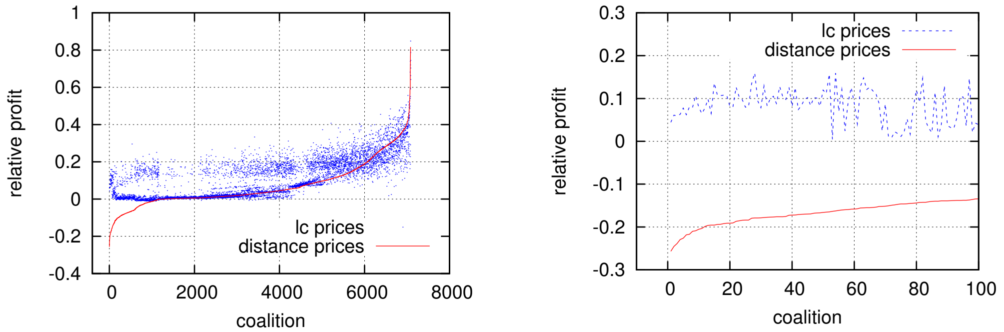

Figure 8.2: Distance vs. unbounded $(c, \bar{x})$-least core prices (1)

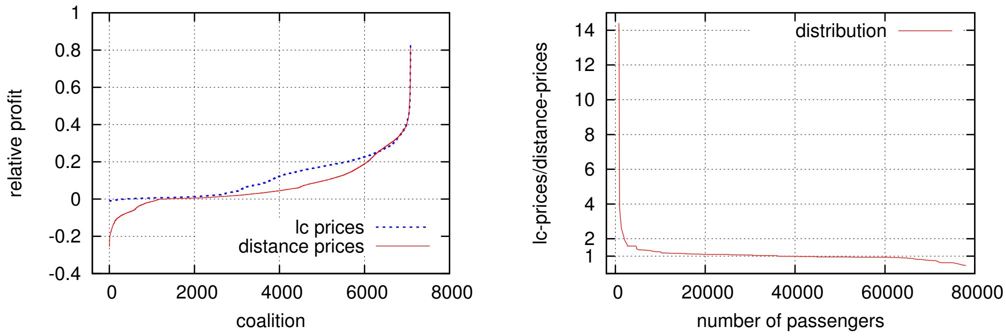

Figure 8.3: Distance vs. unbounded $(c, \bar{x})$-least core prices (2)

Roosendaal Grens – Rotterdam CS. Table 8.1 lists several major coalitions, which would earn a substantial benefit from shrinking the network. This table demonstrates the unfairness and instability of the distance price vector. Figure 8.2 shows that the lc-prices decrease significantly for all of the 100 worst coalitions. Many of them have even a relative profit of more than 10% with the lc-prices.

The picture on the right side of Figure 8.3 plots the distribution of the ratio between the lc-prices and the distance prices. A point (Π, ρ) in this graph means that there are exactly Π passengers who have to pay at least ρ times their distance prices. It can be seen that lc-prices are lower, equal, or slightly higher than the distance prices for most passengers. However, some passengers, mainly in the periphery of the country, pay much more to cover the costs that they produce. The increment factor is at most 3.775 except for two OD-pairs, which face very high price increases. The top of the list is the OD-pair Den Haag CS–Den Haag HS, which gets 14.4 times more expensive. The reason is that the travel path of this OD-pair consists

Coalition ID	Relative profit	Percentage of all passengers
0	-0.256723	15.34%
26	-0.183909	18.35%
56	-0.16356	30.81%
88	-0.141216	34.02%
133	-0.118969	57.34%
191	-0.104312	78.14%

Table 8.1: Unfairness of the distance price vector

of a single edge that is not used by any other travel route. The other two in the top three OD-pairs with a high increment factor are Hengelo–Oldenzaal Grens (factor 11.85) and Apeldoorn–Oldenzaal Grens (factor 3.775). The passengers of these OD-pairs travel in the periphery of the country.

From a game theoretical point of view, these (unbounded) lc-prices can be seen as fair. It would, however, be very difficult to implement such prices in practice. We therefore add property (8.3) in order to limit the difference in the prices for one traveling kilometer of passengers by a factor of K. Considering the results from the previous computation, we set $K = 3$. The $(c, \bar{x})$-least core prices with this constraint are called the bounded lc-prices. Figure 8.4 compares the relative profits of the coalitions in our coalition-pool with the bounded and unbounded $(c, \bar{x})$-least core prices. The picture on the left side presents the relative profits of 7050 coalitions, while the picture on the right side plots also these c-profits but their values are sorted in non-decreasing order. Here we do not plot the remaining 34

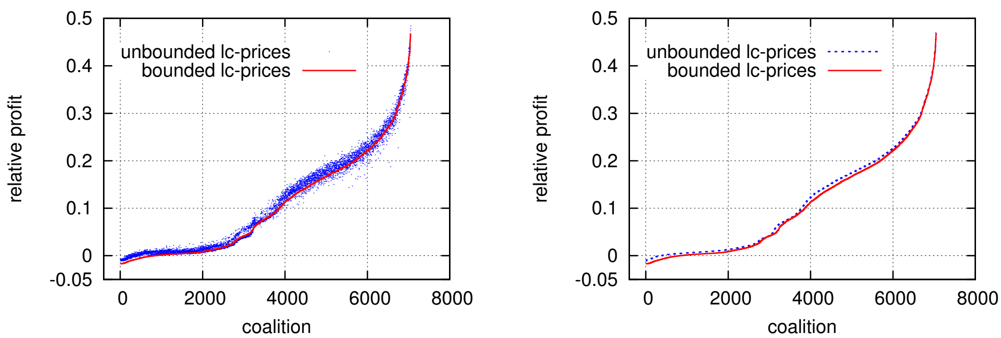

Figure 8.4: Unbounded vs. bounded lc-prices

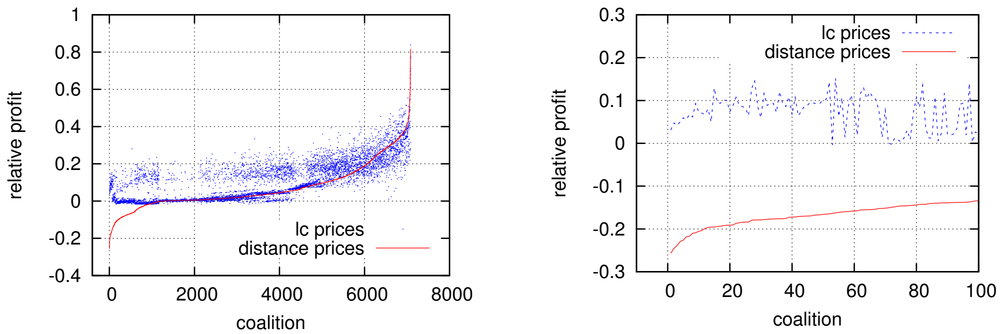

Figure 8.5: Distance vs. bounded $(c, \bar{x})$-least core prices (1)

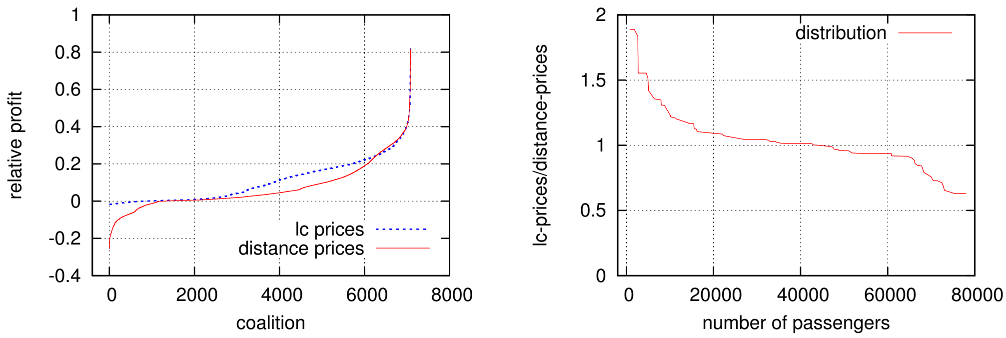

Figure 8.6: Distance vs. bounded $(c, \bar{x})$-least core prices (2)

coalitions in the pool which have larger c-profits in order to observe the differences better. These pictures show that the c-profits of every coalition with the two lc-prices are relatively close to each other. However, the bounded lc-prices are better than the unbounded ones. Figure 8.5 and Figure 8.6 give the same comparisons to the distance prices as Figure 8.2 and Figure 8.3 for the bounded lc-prices. The maximum c-loss of any coalition with the bounded lc-prices is 1.68%, which is slightly worse than before. But the price increments are significantly smaller as plotted in the picture on the right side of Figure 8.6. Again a point (Π, ρ) in this graph says that there are exactly Π passengers who have to pay at least ρ times their distance prices. Table 8.2 presents the 20 smallest and 15 largest ratios of lc-prices to distance prices. The set of coalitions with these ratios is the same for bounded and unbounded lc-prices. With the bounded lc-prices, nobody has to pay more than 1.89 times his distance price and nobody has to pay for one unit more than 3 times the unit price of another passenger. In this way, one can come up with price systems that constitute a good compromise between fairness and enforceability.

The computations were done on a PC with an Intel Core2 Quad 2.83GHz processor and 16GB RAM. CPLEX 11.2 was used as linear and integer program solver. It took in average 41048 and 65929 seconds respectively in order to calculate the bounded and unbounded lc-prices.

OD-pair	$\frac{\text{bounded lc-price}}{\text{distance price}}$	$\frac{\text{unbounded lc-price}}{\text{distance price}}$
BEST 20		
Arnhem–Zevenaar Grens	0.63	0.52
Amsterdam CS–Roosendaal	0.63	0.62
Breda–Eindhoven	0.63	0.56
Roosendaal–Rotterdam CS	0.63	0.47
Roosendaal–Zwolle	0.65	0.66
Eindhoven–Den Haag CS	0.65	0.62
Roosendaal–Schiphol	0.69	0.69
Eindhoven–Roosendaal	0.70	0.61
Eindhoven–Rotterdam CS	0.71	0.63
Amsterdam CS–Eindhoven	0.72	0.74
Den Haag HS–Roosendaal	0.73	0.78
Roosendaal–Utrecht CS	0.73	0.70
Rotterdam CS–Zwolle	0.74	0.75
Amsterdam CS–Rotterdam CS	0.76	0.75
Amsterdam CS-Zevenaar	0.79	0.84
Utrecht CS–Zevenaar	0.84	0.81
Rotterdam CS–Zevenaar	0.85	0.82
Arnhem–Roosendaal	0.85	0.83
Arnhem–Eindhoven	0.86	0.62
Lelystad–Schiphol	0.86	0.51
WORST 15		
Breda–Roosendaal	1.30	1.28
Hengelo–Utrecht CS	1.30	1.37
Schiphol–Zwolle	1.30	1.42
Den Haag CS-Schiphol	1.35	1.31
Den Haag HS–Schiphol	1.36	1.35
Amsterdam Zuid–Zwolle	1.42	1.54
Amsterdam CS–Zwolle	1.52	1.26
Breda–Rotterdam CS	1.55	1.58
Lelystad–Utrecht CS	1.55	1.58
Amsterdam Zuid–Lelystad	1.84	1.88
Apeldoorn–Hengelo	1.89	1.67
Apeldoorn–Oldenzaal	1.89	3.77
Den Haag HS–Den Haag CS	1.89	14.41
Hengelo–Oldenzaal	1.89	11.85
Lelystad–Zwolle	1.89	2.62

Table 8.2: The ratios of lc-prices to distance prices

Chapter 9

Perspectives

We have seen that there does not exist a "perfect" cost allocation method which safisfies all of our axioms. For more specific families of cost allocation games, there may exist cost allocation methods which safisfy more desired properties simultaneously than for the general case. However, real-world applications often do not fulfil the required assumptions of such families. This negative result shows that total fairness is hard to define mathematically and often does not exist in practice. The presented concepts can model fairness in some way and reflect the real costs of coalitions. However, there are still several issues which must be answered for each application, namely, the choice of weight function and reference price vector. On one hand, this shows that mathematical concepts can help but in order to construct a cost allocation method which should be accepted by all users convincing and negotiating are still required. On the other hand, the freedom by choosing weight function and reference price vector provides the decision maker the possibility to have a pool of several cost allocations, which are fair in some way, for selecting.

The numerical results for the considered applications show that the game theoretical concepts provide much better prices in terms of coalitional stability than the traditional cost allocation approaches. These game theoretical prices minimize the incentive for users to form subcoalitions and increase satisfaction with the allocated prices amongst users. We are confident that these game theoretical concepts will bring advantages against traditional approaches for every real-world applications.

In this thesis, the difficulty by calculating the allocations which are based on the considered game theoretical concepts are overcome by using constraint generation approaches. Several general heuristics are considered in order to improve the solving time. Our approach works well for large applications like the ticket pricing problem for the Dutch IC network with 85 players and $2^{85}-1$ coalitions. In order to solve even larger applications or to decrease the running time, one may investigate more problem specific heuristics and valid cuts for the separation problem.

Bibliography

[1] R. Aumann and S. Hart, *Handbook of Game Theory with Economic Applications, Vol. 3*, Handbooks in Economics 11, Elsevier, Amsterdam, 2002.

[2] E. Bjørndal, H. Hamers, and M. Koster, *Cost allocation in a bank ATM network.*, Mathematical Methods of Operations Research, 59 (2004), pp. 405–418.

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

[4] R. Borndörfer, M. Grötschel, S. Lukac, M. Mitusch, T. Schlechte, S. Schultz, and A. Tanner, *An auctioning approach to railway slot allocation*, Tech. Rep. ZIB Report 05-45, Zuse-Institut Berlin, 2005.

[5] R. Borndörfer and N.-D. Hoang, *Determining Fair Ticket Prices in Public Transport by Solving a Cost Allocation Problem*, to appear in Modeling, Simulation and Optimization of Complex Processes, Proceedings of the Fourth International Conference on High Performance Scientific Computing, Springer, 2011.

[6] R. Borndörfer, M. Neumann, and M. E. Pfetsch, *Optimal fares for public transport*, Operations Research Proceedings 2005, (2006), pp. 591–596.

[7] ——, *Models for fare planning in public transport*, Tech. Rep. ZIB Report 08-16, Zuse-Institut Berlin, 2008.

[8] M. R. Bussieck, *Optimal Lines in Public Rail Transport*, PhD thesis, TU Braunschweig, 1998.

[9] E. Danna, E. Rothberg, and C. Le Pape, *Exploring relaxation induced neighborhoods to improve MIP solutions*, Mathematical Programming Series A, 102 (2005), pp. 71–90.

[10] G. B. Dantzig, *On the significance of solving linear programming problems with some integer variables*, Econometrica, 28 (1960), pp. 30–44.

[11] R. W. Day and S. Raghavan, *Fair payments for efficient allocations in public sector combinatorial auctions*, Management science, 53 (2007), pp. 1389–1406.

[12] N. Devanur, M. Mihail, and V. V. Vazirani, *Strategyproof cost-sharing mechanisms for set cover and facility location games*, Proceedings of the 4th ACM Conference on Electronic Commerce, (2003), pp. 108–114.

[13] S. Engevall, *Cost Allocation in Some Routing Problem - A Game Theoretic Approach*, PhD thesis, Linköping Institute of technology, 2002.

[14] S. Engevall, M. Göthe-Lundgren, and P. Värbrand, *The traveling salesman game: An application of cost allocation in a gas and oil company*, Annals of Operations Research, 82 (1998), pp. 453–471.

[15] ——, *The heterogenous vehicle-routing game*, Transportation science, 38 (2004), pp. 71–85.

[16] U. Faigle, W. Kern, and J. Kuipers, *On the computation of the nucleolus of a cooperative game*, Internat. J. Game Theory, 30 (2001), pp. 79–98.

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

[18] L. Fleischer, J. Könemann, S. Leonardi, and G. Schäfer, *Simple cost sharing schemes for multi-commodity rent-or-buy and stochastic steiner tree*, Proceedings of the 38th Annual ACM Symposium on Theory of Computing, (2006), pp. 663–670.

[19] V. Fragnelli, *Game theoretic analysis of transportation problems*, Proceedings of the 4th Twente Workshop on Cooperative Game Theory joint with 3rd Dutch-Russian Symposium, (2005), pp. 27–38.

[20] V. Fragnelli, I. Garcia-Jurado, H. Norde, F. Patrone, and S. Tijs, *How to Share Railway Infrastructure Costs?*, F.Patrone, I.Garcia-Jurado and S.Tijs, editors, *Game Practice: Contributions from Applied Game Theory*, Kluwer, Amsterdam, 1999, pp 91-101.

[21] V. Fragnelli and A. Iandolino, *A cost allocation problem in urban solid wastes collection and disposal.*, Mathematical Methods of Operations Research, 59 (2004), pp. 447–463.

[22] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, New York, 1979.

[23] D. B. Gillies, *Some Theorems on N-Person Games*, PhD thesis, Princeton University, 1953.

[24] ——, *Solutions to general non-zero-sum games*, Annals of Mathematics Studies 40, (1959), pp. 47–85.

[25] M. Goemans and M. Skutella, *Cooperative facility location games.*, Journal of Algorithms, 50 (2004), pp. 194–214.

[26] M. Grötschel, L. Lovász, and A. Schrijver, *The ellipsoid method and its consequences in combinatorial optimization*, Combinatorica, 1 (1981), pp. 169–197.

[27] ——, *Geometric Algorithms and Combinatorial Optimization, 2. corr. ed.*, Springer-Verlag, 1993.

[28] A. Gupta, A. Kumar, and T. Roughgarden, *Simpler and better approximation algorithms for network design*, Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing, (2003), pp. 365–372.

[29] A. Gupta, A. Srinivasan, and E. Tardos, *Cost-sharing mechanisms for network design*, Proceedings of the 7th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, (2004), pp. 139–150.

[30] Å. Hallefjord, R. Helming, and K. Jørnsten, *Computing the nucleolus when the characteristic function is given implicitly: A constraint generation approach*, International Journal of Game Theory, 24 (1995), pp. 357–372.

[31] S. S. Hamlen, W. A. Hamlen Jr., and J. T. Tschirhart, *The use of core theory in evaluating joint cost allocation schemes*, The Accounting Review, 52 (1977), pp. 616–627.

[32] S. Herzog, S. Shenker, and D. Estrin, *Sharing the cost of multicast trees: An axiomatic analysis*, IEEE/ACM Transactions on Networking, 5 (1997), pp. 847–860.

[33] N. Immorlica, M. Mahdian, and V. S. Mirrokni, *Limitations of cross-monotonic cost sharing schemes*, Proceedings of the Sixteenth Annual ACM-SIAM symposium on Discrete Algorithms, (2005), pp. 602–611.

[34] K. Jain and V. V. Vazirani, *Applications of approximation algorithms to cooperative games*, Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing, (2001), pp. 364–372.

[35] P. A. Kattuman, R. J. Green, and J. W. Bialek, *Allocating electricity transmission costs through tracing: A game-theoretic rationale*, Operations Research Letters, 32 (2004), pp. 114–120.

[36] L. G. Khachiyan, *A polynomial algorithm in linear programming (in Russian)*, Doklady Akademii Nauk SSSR, 244 (1979), pp. 1093–1096 (English translation: Soviet Mathematics Doklady 20 (1979) pp. 191–194).

[37] E. Kohlberg, *On the nucleolus of a characteristic function game*, SIAM Journal on Applied Mathematics, 20 (1971), pp. 62–66.

[38] J. Könemann, S. Leonardi, and G. Schäfer, *A group-strategyproof mechanism for Steiner forests*, Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, (2005), pp. 612–619.

[39] J. Könemann, S. Leonardi, G. Schäfer, and S. van Zwam, *From primal-dual to cost shares and back: A stronger lp relaxation for*

the Steiner forest problem, Proceedings of the 32nd International Colloquium on Automata, Languages and Programming, (2005), pp. 930–942.

[40] S. Leonardi and G. Schäfer, *Cross-monotonic cost sharing methods for connected facility location games*, Proceedings of the 5th ACM Conference on Electronic Commerce, (2004), pp. 242–243.

[41] F. Leveque, *Transport Pricing of Electricity Networks*, Kluwer Academic Publishers, Dordrecht, 2003.

[42] S. Littlechild, *A simple expression for the nucleolus in a special case*, International Journal of Game Theory, 3 (1971), pp. 21–29.

[43] S. Littlechild and G. Owen, *A simple expression for the Shapley value in a special case*, Management Science, 20 (1973), pp. 370–372.

[44] S. Littlechild and G. Thompson, *Aircraft landing fees: A game theory approach*, The Bell Journal of Economics and Management Science, 8 (1977), pp. 186–204.

[45] M. Maschler, B. Peleg, and L. S. Shapley, *Geometric properties of the kernel, nucleolus, and related solution concepts*, Mathematics of Operations Research, 4 (1979), pp. 303–338.

[46] N. Megiddo, *Cost allocation for steiner trees*, Networks, 8 (1978), pp. 1–6.

[47] H. Moulin and S. Shenker, *Strategyproof sharing of submodular costs: Budget balance versus efficiency*, http://www.aciri.org/shenker/cost.ps, (1997).

[48] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic Game Theory*, Cambridge University Press, Cambridge, 2007.

[49] M. Pal and E. Tardos, *Group strategyproof mechanisms via primal-dual algorithms*, Proceedings of the Annual IEEE Symposium on Foundations of Computer Science, (2003), pp. 584–593.

[50] T. Parker, *Allocation of the Tennessee Valley Authority projects*, Transaction of the American Society of Civil Engineers, 108 (1943), pp. 174–187.

[51] J. A. M. Potter, *An axiomatization of the nucleolus*, International Journal of Game Theory, 19 (1991), pp. 365–373.

[52] Ransmeier, *The Tennessee Valley Authority: A case study in the economics of multiple purpose stream planning*, Vanderbilt University Press; Nashville, Tennessee, (1942).

[53] D. Schmeidler, *The nucleolus of a characteristic function game*, SIAM Journal on Applied Mathematics, 17 (1969), pp. 1163–1170.

[54] L. S. Shapley, *A Value of n-person Games*, In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games*, Princeton University Press, 1953, pp. 307-317.

[55] ——, *On balanced sets and cores*, Naval Research Logistics Quarterly, 14 (1967), pp. 453–460.

[56] L. S. Shapley and M. Shubik, *Quasi-cores in a monetary economy with non-convex preferences*, Econometrica, (1966), pp. 805–827.

[57] M. Shubik, *Incentives, decentralized control, the assignment of joint costs and internal pricing*, Management Science, 8 (1962), pp. 325–343.

[58] D. Skorin-Kapov, *On the core of the minimum cost Steiner tree game in networks*, Annals of Operations Research, 57 (1995), pp. 233–249.

[59] C. Snijders, *Axiomatization of the nucleolus*, Mathematics of Operations Research, 20 (1995), pp. 189–196.

[60] A. I. Sobolev, *The characterization of optimality principles in cooperative games by functional equations*, Mat. Metody soc. Nauk., 6 (1975), pp. 94–151.

[61] P. Straffin and J. P. Heaney, *Game theory and the tennessee valley authority*, International Journal of Game Theory, 10 (1981), pp. 35–43.

[62] A. D. Taylor and A. M. Pacelli, *Mathematics and politics. Strategy, voting, power and proof. 2nd ed.*, Springer, 2008.

[63] H. P. Young, *Monotonic solutions of cooperative games*, International Journal of Game Theory, 1985 (14), pp. 65–72.

[64] ——, *Cost Allocation: Methods, Principles, Application*, North-Holland, Amsterdam, 1985.

[65] ——, *Cost Allocation*, In R. J. Aumann and S. Hart, editors, *Handbook of Game Theory*, vol. 2, North-Holland, Amsterdam, 1994, pp. 1193-1235.

[66] ——, *Cost allocation, demand revelation, and core implementation*, Mathematical Social Sciences, 36 (1998), pp. 213–228.

[67] H. P. Young, N. Okada, and T. Hashimoto, *Cost allocation in water resources development*, Water Resources Research, 18 (1982), pp. 463–475.

Notations

$\mathcal{C}(\Gamma)$	the core of Γ
$\mathcal{C}_{\varepsilon,f}(\Gamma)$	the (ε, f)-core of Γ
Γ_D	the dummy-free subgame of Γ
Γ_S	the subgame of Γ corresponding to the coalition S
$\mathcal{D}$	the dummy operator
$d(\Gamma)$	the set of all dummy players of Γ
dim	dimension of a vector space
$e_f(S, x)$	f-excess of coalition S at x
$\theta_{f,\lambda}$	f-excess function of Γ
$\lambda\Gamma$	scalar multiplication of λ and Γ
$\mathcal{LC}_f(\Gamma)$	the f-least core of Γ
$\mathcal{LC}_{f,r}(\Gamma)$	the (f, r)-least core of Γ
$\mathcal{N}_f(\Gamma)$	the f-nucleolus of Γ
$\mathbb{Q}_+$	$\{x \in \mathbb{Q} \mid x \geq 0\}$
$\mathbb{Q}_{>0}$	$\{x \in \mathbb{Q} \mid x > 0\}$
$\mathbb{R}_+$	$\{x \in \mathbb{R} \mid x \geq 0\}$
$\mathbb{R}_{>0}$	$\{x \in \mathbb{R} \mid x > 0\}$
R^N	$R^{\lvert N \rvert}$ for some given set N
Σ^+	$\Sigma \backslash \{\emptyset\}$
span(V)	the subspace of $\mathbb{R}^n$ spanned by $V = \{v_1, v_2, \ldots, v_m\}$, $v_i \in \mathbb{R}^n$, $i = 1, \ldots, m$
$x(S)$	$\sum_{i \in S} x_i$
$\mathcal{X}(N, c, P, \Sigma)$	imputation set $\{x \in P \mid x(N) = c(N)\}$
χ_S	incidence vector of the set S with respected to $N \supseteq S$
χ_Σ	$\{\chi_S \mid S \in \Sigma\}$ for $\Sigma \subseteq 2^N$
IP	integer program
LP	linear program
MIP	mixed integer program

Index