A Cooperative Approach to Sequencing and Connection Problems

A Cooperative Approach to Sequencing and Connection Problems

Proefschrift

ter verkrijging van de graad van doctor aan de Universiteit van Tilburg, op gezag van de rector magnificus, prof.dr. Ph. Eijlander, in het openbaar te verdedigen ten overstaan van een door het college voor promoties aangewezen commissie in de aula van de Universiteit op vrijdag 4 december 2009 om 10.15 uur door

Barış Bekir Çiftçi,

geboren op 23 februari 1978 te Ankara, Turkije.

Promotores: Prof.Dr. P.E.M. Borm
Prof.Dr. H.J.M. Hamers

Acknowledgments

This thesis summarizes most of my research efforts of the past four years. I would like to express my gratitude to everyone who supported me for the completion of this work. I must begin with an expression of my debt to my advisors, Peter and Herbert, for their guidance and the valuable comments they gave during the preparation of this thesis. Peter's "I don't believe you" followed by Herbert's "Prove" will continue to stimulate my self criticism as long as I do research. I also thank Tilburg University, its staff and faculty members for providing me with a great environment to work on this thesis.

I would like to thank to Semih Koray for teaching me the basics of microeconomic theory and game theory, supervising me when I was a Ph.D. student at Bilkent University and most importantly introducing me the beauty of the challenge of explaining the choices and values of the society and the individual by intelligent handling of mathematics. I would also like to thank Tarık Kara and Taner Yiğit for their support and encouragement.

As far as the results of my research are concerned, I profited most from my co-authors Stef Tijs, Marco Slikker and Dinko Dimitrov. I want to thank them all for that. My special thanks go to Stef Tijs for affecting me with his enthusiasm and intellectuality. I would like to thank also to Rodica Branzei, Stefano Moretti, Yuan Ju and Manuel Mosquera for sharing their thoughts and suggesting the way forward at some points during my research.

My thanks also go to my committee members, Stef Tijs, Carles Rafels Pallarola, Flip Klijn and Jeroen Kuipers, for the time and effort spent on the manuscript and also for their insightful comments.

I am most grateful to my parents for their continuing love, support and encouragement. I would like to use this opportunity to express my gratitude and to dedicate this thesis to them. My thanks also go to my sisters Aslı and Bilge, my grandma Hakiye, my parents-in-law, my brother-in-law, Batur and his family and my cousins Umut and Utku for their care, encouragement and support.

My warmest thanks go to my friends and my fellow PhD students. Firstly, many thanks go to Tunga for being a "gerçek dost". I thank Gerwald for all the good time we had at the office. I thank Wouter, Teresa, Sebastian, Agnieszka, Hans, Vivian, Pilar, Ina, Pedro, Miguel, Mohammed, Anne, Tural, Arantza, John for their continuing care and friendship.

Finally, I would like to use this opportunity to express my gratitude and love to Esen, my wife, who continuously brighten up my life. This thesis would not have been possible without her encouragement and care. Seni çok seviyorum.

Barış Bekir Çiftçi
Rotterdam, 2009

Contents

Chapter 1

Introduction

1.1 Cost Allocation Problems and Game Theory

There are many economic settings in which a group of agents wishes to undertake a joint enterprise in order to save costs. Doctors, for example, look for colleagues to share an office in order to save from rent, equipment and secretarial help. Individuals or firms wish to jointly invest in common networks such as communication networks and distribution channels. Similarly, countries wish to collaborate in order to solve environmental or security problems. The success of such enterprises often relies on agreements on how to share the cost savings generated. That is, joint enterprises require a cost allocation mechanism that is efficient, fair and provides incentives to each agent (and also to each group of agents) involved to agree upon.

An economic approach one can take to analyze joint cost savings problems is by means of cooperative game theory.

Game theory is a branch of applied mathematics that provides tools for analyzing situations in which the actions of each involved party (called a player) have an effect on the outcome which is of interest to all. The competition between firms or political parties, the conflict between a boss and a worker, war and peace/disarmament/ environmental negotiations between countries, cooperation of producers sharing common production facilities and so on, all provide examples of these situations, i.e., games. As illustrated in the examples above, the mutual interdependence among the players is the essence of a game and what makes game theory a useful tool for the analysis of the behavior of interacting decision-makers. The book Theory of Games and Economic Behavior by von Neumann and Morgenstern (1944) is regarded by many as the starting point of game theory. As the title of the book of von Neumann and Morgenstern reveals the intention of the authors was to provide a new kind of mathematics for economic analysis. Indeed, starting from the late sixties, game theory is being applied to explain the behavior of firms and individuals in a wide range of economic situations. Nor is economics alone: operations research, accounting, finance, law, marketing, political science and sociology are beginning similar experiences.

Game Theory can be roughly divided into two broad areas: non-cooperative (or strategic) and cooperative (or coalitional) game theory. Generally speaking, non-cooperative game theory focuses on those situations where the competitive nature of interaction is dominant and players make choices which are based only on their perceived self-interest.

By contrast, the cooperative game theory models those situations in which cooperative behavior is the central feature of social or economic interaction. In a cooperative game, players can reach formal joint agreements, such as legally binding arrangements to act as a single entity. A cooperative game typically abstracts itself away from negotiations to reach agreements and mechanisms the players use to enforce agreements. Rather it describes only the physical outcomes that can jointly be achieved by each group of players. For this reason, cooperative game theory can be considered as a structural theory. The most commonly used model in the theory of cooperative games is the model of transferable utility games. A transferable utility game, or a TU-game, describes the monetary value that each group of players can achieve by most efficient means.

Generally, the focus in the analysis of a TU-game is on the allocation of the value that is achieved by the grand coalition of all players, where the values of the subcoalitions serve as a benchmark. The literature on TU-games has proposed many solution concepts each with its own appealing properties. Most important of these solution concepts are the core (Gillies, 1953), the Shapley value (Shapley, 1953), the nucleolus (Schmeidler, 1969) and the compromise value (Tijs, 1981).

The central issue of this monograph is to address cost allocation problems arising from sequencing problems and connection problems. Sequencing problems consider a group of agents who are waiting to be served in a facility and focuses on the problem of the allocation of the cost savings that can be obtained by switching from an initial service order to an optimal one. Connection problems consider the cost allocation problems arising from situations in which a group of agents wishes to collaborate and jointly invest in the construction or the maintenance of a common network. Both of these problems appear in a great number of diverse economic settings. Allocating patients to surgery rooms, scheduling of computer programs on servers or jobs in shop-floor on machines are common examples of sequencing problems while a group of villagers that has to construct and pay pipelines from their respective houses to a water supplier is a common example of connection problems.

The methods we use in this monograph to analyze the cost allocation problems arising from sequencing and connection problems mainly rely on models of TU-games. The following example illustrates how a cost allocation problem arising from a simple sequencing situation can be modeled and analyzed by a TU-game.

Example 1.1.1 Consider 3 companies each of which has a broken tool that has to be repaired by a maintenance service in order for the companies to restart production. The number of hours (processing time) required to repair each of these three tools is given in Table 1.1. For example, the processing time required to repair the tool

company	1	2	3
processing time in hours	3	1	2
cost coefficient in 1,000s of euros	3	2	6

Table 1.1: Processing times and cost coefficients in Example 1.1.1

of company 2 is 1 hour. Suppose that according to a first-come-first-serve principle, the tool of company 1 will be repaired first, followed by company 2, and, finally, company 3 (See Figure 1.1). Waiting until the reparation of the tool is costly for the companies because of the production that is lost meanwhile. Moreover, lost production in different companies may have different values per time unit. This is reflected by the cost coefficients in Table 1.1. For example, if the company that owns tool 2 has to wait 3 hours till it can restart production again, then the costs for this company will be $3 \cdot 2,000 = 6,000$ euros. Similarly, the total costs to repair the tools according to the initial order are $3 \cdot 3,000 + 4 \cdot 2,000 + 6 \cdot 6,000 = 53,000$ euros. It can be

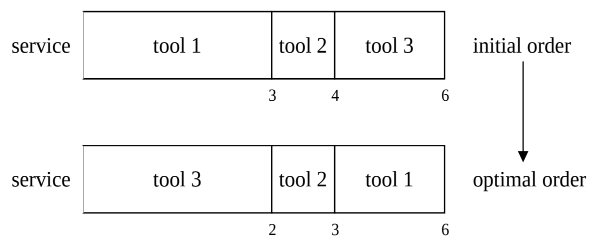

Figure 1.1: The initial order and the optimal order in Example 1.1.1

checked that this order is not optimal, i.e., it does not minimize the total costs. The unique optimal order in this situation is 3,2,1 (See Figure 1.1) and the associated total costs equal $2 \cdot 6,000 + 3 \cdot 2,000 + 6 \cdot 3,000 = 36,000$ euros. Hence, if all companies cooperate, they can decrease the total costs from 53,000 to 36,000 euros. Assume now that company 1 and 2 come together to discuss their possibilities to achieve cost savings without the help of company 3. Obviously, they could switch positions without changing the waiting time of company 3. It can be checked that if companies 1 and 2 switch places then their total costs will be $1 \cdot 2,000 + 4 \cdot 3,000 = 14,000$ euros and this is the minimum amount for companies 1 and 2. Now, consider companies 1 and 3. Since an initial processing order is established, company 2 might have the opportunity to prevent the switching between companies 1 and 3. Or, it might be that company 2 can only veto this switch if the processing time of company 3 is larger than the processing time of company 1. Hence, whether to allow the switching between companies 1 and 3 or under which conditions to allow the switching between companies 1 and 3 are important modeling decisions. For this situation, we take the perspective that companies 1 and 3 cannot switch positions without the cooperation of company 2. Hence, companies 1 and 3 cannot generate any cost savings on their own. The minimal costs for each group of companies are depicted in Table 1.2.

group	{1}	{2}	{3}	{1,2}	{1,3}	{2,3}	{1,2,3}
minimal total costs in 1,000s of euros	9	8	36	14	45	42	36

Table 1.2: The minimal total costs for each group of companies in Example 1.1.1

We can also consider the cost savings that each group of companies achieve when they cooperate. Consider again companies 1 and 2. Their total costs with respect to the initial order equal 17,000 euros. We showed that the minimum total costs these companies can achieve equal 14,000 euros. Hence, when they cooperate they save 3,000 euros. The maximal cost savings for each group of companies is depicted in Table 1.3.

group	{1}	{2}	{3}	{1,2}	{1,3}	{2,3}	{1,2,3}
maximal cost savings in 1,000s of euros	0	0	0	3	0	2	17

Table 1.3: The maximal cost savings for each group of companies in Example 1.1.1

Now we can start looking for a solution for the allocation problem at hand by taking the values of the subcoalitions as a benchmark. An important property of a cost allocation is that it has to provide incentives to each agent and to each group of agents involved to agree upon. For instance, the allocation 8,000 euros for company 1, 7,000 euros for company 2 and 21,000 euros for company 3 does not give companies 1 and 2 the required incentives to participate in the cost savings project. For, together companies 1 and 2 are charged 15,000 euros whereas they could achieve total costs of 14,000 euros on their own. In other words, this cost allocation is not stable, because companies 1 and 2 will not accept it. This notion of stability is the main idea behind the solution concept known as core.

A core allocation is the one that gives every group of players an incentive to participate by allocating them more than what the group could achieve on its own. The core of our cost game, which is represented in Figure 1.2, is the set of all allocations (x_1, x_2, x_3) satisfying the following constraints:

$$\begin{array}{ccccccc} x_1 & + & x_2 & + & x_3 & = & 36,000. \\ x_1 & + & x_2 & & & \leq & 14,000. \\ x_1 & & & + & x_3 & \leq & 45,000. \\ & & x_2 & + & x_3 & \leq & 42,000. \\ x_1 & & & & & \leq & 9,000. \\ & & x_2 & & & \leq & 8,000. \\ & & & & x_3 & \leq & 36,000. \end{array}$$

A solution satisfying the constraints above is (1500, 5500, 29000). ◇

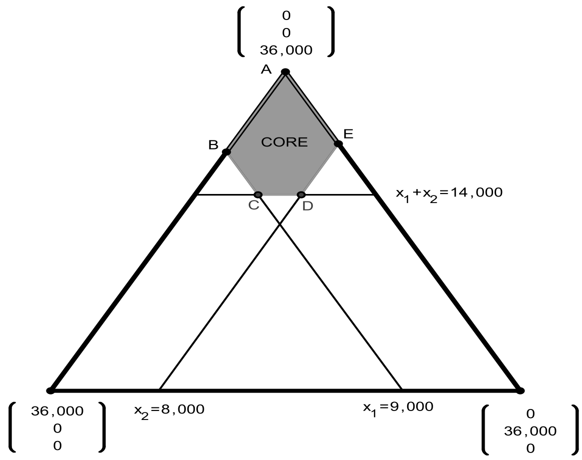

Figure 1.2: Core of the cost game in Example 1.1.1

Although the core of the game associated with the sequencing problem in Example 1.1.1 is non-empty, the core of TU-games need not be non-empty in general. Since core stability is a compulsory condition for the viability of a cost allocation, emptiness or non-emptiness of the core is an essential feature of the games associated with cost allocation problems. Hence, the first challenge in the cooperative analysis of cost allocation problems is to explore whether the core of the associated TU-games are non-empty in general. And if the core of the associated games can be empty, then the challenge is to explore for the conditions that guarantee the existence of allocations that satisfy core stability.

The sequencing problem in Example 1.1.1 also demonstrates the combinatorial nature of these problems. Given the cost functions of the players involved in a sequencing situation, the problem of finding an optimal order that minimizes the total costs of the grand coalition (or of sub-groups of players) is a combinatorial optimization problem. As examples of other studies that focus on the interplay between the optimization of costs of a project and the allocation of costs among the participants of the project we mention: minimum cost spanning tree games and spanning network games (cf. Claus and Kleitman, 1973; Bird, 1976; Megiddo, 1978; Granot and Huberman, 1981), linear programming games (cf. Owen, 1975), flow games (cf. Kalai and Zemel, 1982), traveling salesman games (cf. Potters et al., 1992), Chinese postman games (cf. Granot et al., 1999), sequencing games (cf. Curiel et al., 1989) and project games (cf. Estevez-Fernandez et al., 2007). This kind of problems have been baptized as operations research games in the survey Borm et al. (2001).

1.2 Overview

Chapter 2 is dedicated to the analysis of cost allocation problems arising from specific sequencing situations. In sequencing situations, there exists a set of clients each having one job to be processed on a (number of) machine(s). The clients incur costs that depend on the completion time of their jobs. Assuming that there is an initial order on the jobs, two problems arise in these situations: How to find an optimal processing order which minimizes the total costs and how to distribute these cost savings obtained by changing the initial order to an optimal order in a fair way. Game theoretic analysis of this "fairness" problem in sequencing situations was initiated by Curiel et al. (1989). This study considered one machine sequencing situations and tackled the problem of the distribution of the maximal cost savings by analyzing the corresponding cooperative sequencing games. It was shown that these games are convex and hence balanced, i.e, there are allocations of the cost savings which are stable in the sense that no coalition of clients can receive a larger payoff by rearranging the position of their own jobs in an admissible way. Moreover, Curiel et al. (1989) proposed an allocation rule, the so called equal gain splitting rule for these one-machine sequencing situations and provided an axiomatical characterization of this rule. The following studies in this strand of literature have extended the basic model by considering restrictions on the jobs (e.g., ready times, due dates), by allowing more general cost functions for the clients or by considering multiple-machine sequencing situations. A common feature of all these studies is that the analysis is restricted to manufacturing systems which consist of machines that can process only one job at a time. However, in many manufacturing/service systems, operations are carried out by batch machines which can simultaneously process multiple jobs.

In Section 2.1, we extend the game theoretical approach to the cost allocation problems arising from sequencing situations on systems that consist of batch machines. First, we consider batch sequencing situations: sequencing situations which consist of a single batch machine where the batch machine's capacity is defined as the maximum number of jobs that can be processed at any one time. It is shown that the cooperative games corresponding to batch sequencing situations are convex and an expression for the Shapley value of these games is provided. We also consider extensions of the equal gain splitting rule and the split core (Hamers et al., 1996) for these sequencing situations and provide axiomatic characterizations of these solution concepts. Second, we analyze relaxed batch sequencing games by considering relaxations in the notion of admissibility of reorderings and prove that these games are also balanced. Third, we analyze various aspects of flow-shop sequencing situations which consist of batch machines only. In particular, we provide two cases in which the cooperative game arising from the flow-shop sequencing situation is equal to the game arising from a sequencing situation that corresponds to one specific machine in the flow-shop.

Another common assumption in the analysis of sequencing games as initiated by Curiel et al. (1989) is that no set-up time (i.e, the delays required for reconfiguration or preparation of a machine or a system) is incurred prior to the processing of the jobs. This assumption requires that set-up times are negligible or can be included in processing times. However, in real industry, if a machine has to manufacture different types of products, then a significant set-up time is almost always required (See for example Pinedo, 2005). Moreover, recent studies in the scheduling literature show that explicit incorporation of set-up times in scheduling decisions results in tremendous cost savings (See Allahverdi et al., 2008).

In Section 2.2, we analyze the cost allocation problems arising from so-called family sequencing situations. In family sequencing situations, jobs are partitioned into families according to their similarities. A job does not require a set-up when following another job from the same family. A set-up time, known as a family set-up time, is required when a job follows a member of some other family. Family sequencing situations have gained considerable attention in the scheduling literature and we refer to Allahverdi et al. (2008) for a review of this literature. Following the ideas in Curiel et al. (1989), we associate cooperative games with family sequencing situations by defining the worth of a coalition as the maximum savings it can obtain by means of an admissible rearrangement. We show that the cooperative games associated with family sequencing problems have non-empty cores. This result is obtained by showing that a specific marginal vector of the game is a core element.

In Chapter 3 we analyze cost allocation problems arising from connection problems, i.e., economic settings in which a set of agents wishes to collaborate and jointly invest in the construction/maintenance of a common network. We look at two such problems in particular: minimum cost spanning tree problems, which consider a set of agents each of whom has to establish a connection to a source and highway problems which consider a set of agents each of whom has to establish a connection between a starting point and a destination point.

Imagine a group of villagers that has to construct and pay pipelines from their respective houses to a water supplier. Each villager could choose to build a direct link to the supplier, but such a decision would likely be highly inefficient. Instead, it may be cheaper for some villagers to connect directly to the supplier, whereas others could connect indirectly via links to neighboring villagers. Indeed, in these situations a configuration of links that minimizes the total cost of connection is provided by a minimum cost spanning tree (mcst) and hence these situations are called mcst situations. Once the agents in a mcst situation agree on which mcst to construct, the second problem they jointly face is the allocation of the costs of this mcst in a fair way. This type of cost allocation problems was first introduced in the economics literature by Claus and Kleitman (1973). The seminal paper by Bird (1976) provided the first game theoretical treatment of this problem by associating a coalitional game with transferable utility to mcst problems. Many division rules have been proposed in the literature as appropriate cost allocations for mcst problems. Recently, it has been shown that the equal remaining obligations rule (Feltkamp et al., 1994) for mcst problems satisfies many appealing properties (e.g., cost monotonicity, population monotonicity, equal treatment) and can be obtained with different approaches. Moreover, Bergantiños and Vidal-Puga (2007a) showed that other rules in the literature fail to satisfy some properties that are satisfied by the equal remaining obligations rule.

The original definition of equal remaining obligations rule by Feltkamp et al. (1994) consists of a step-by-step procedure: Kruskal's algorithm (Kruskal, 1956) is employed to construct an mcst and at each step of the algorithm the cost of the constructed edge is divided among agents who make use of the edge with respect to a prespecified scheme. In Section 3.2, we present a new approach to obtain the equal remaining obligations rule and hence provide yet another support for this important rule. For this aim, first, we define the vertex oriented construct and charge procedure which leads to an mcst for the problem and also a cost sharing allocation where each agent pays the edge which he chose to construct in the procedure. Then, we show that the equal remaining obligations rule can be obtained as the average of the cost allocations provided by a vertex oriented construct and charge procedure for each order of players. Moreover, in Sections 3.3 and 3.4, we investigate the extensions of the results obtained in Section 3.2 for minimum cost spanning forest situations (cf. Rosenthal, 1987) and mcst situations with two sources, respectively. For both of these situations, we first show that both Kruskal's algorithm and the vertex oriented construct and charge procedure can be defined in a way that they yield efficient algorithms. Second, we extend the definition of equal remaining obligations rule to these multi-source situations and prove that equal remaining obligations rule can again be obtained as the average of the cost allocations provided by vertex oriented construct and charge procedures.

Most of the current literature on the allocation of costs in connection problems focuses on the mcst problems or its variants. A common feature of these problems is that each agent in the problem has to establish a connection with a nonempty subset of the available sources in the network. However, in some connection situations, there is no particular point that every agent in the problem has to be connected to. For example, the users of a highway network need a connection only between their entry and exit points in the network.

Mosquera and Zarzuelo (2006) address the problem of fair allocation of the construction costs of a highway network. For this aim, they formally consider highway problems and analyze the corresponding cooperative cost games called highway games. In a highway problem, the possibilities regarding the construction of the highway network are determined by a connected graph. The set of vertices of the graph represents the potential entry and exit points and the edges in the graph represent the possible highway connections that can be constructed. Given a highway problem, a corresponding highway game is defined as a cooperative cost game which associates to each coalition of players the total cost of the cheapest selection of edges in the graph which connects the entry and exit point of every member of the coalition. Mosquera and Zarzuelo (2006) restricted attention to highway problems in which the underlying graph is a tree. In this setting, there is only one path between an entry and exit point.

In Section 3.5, we study highway problems in which the underlying graphs are weakly cyclic, i.e., connected graphs for which every edge in the graph is contained in at most one cycle. First, we focus on the question for which class of graphs the corresponding games are always concave. For this aim, a graph G is defined to be highway-game concave if for each highway problem in which G is the underlying graph the corresponding highway game is concave. We prove that a graph is highway-game concave if and only if it is weakly triangular. Then we focus on the balancedness of highway games induced by weakly cyclic graphs. It was shown by Kuipers (1997) that highway games induced by cyclic graphs need not be balanced in general. However, we prove that highway games on weakly cyclic graphs are balanced.

In Chapter 4, we consider the formation of coalitions through sequences of binding bilateral agreements.

Studies on coalition formation in many real-life contexts like voting situations and international trade negotiations suggest that cooperation of all related parties can hardly be achieved through simultaneous acceptance of an agreement. Hence, a basic policy recommendation in these situations is to broaden a coalition step by step through a sequence of binding bilateral agreements. For a coalition formed through binding bilateral agreements can grow larger by making use of the commitment/ synergy already attained by the coalition to facilitate the persuasion of one of the outsider parties to enter the coalition.

Chapter 4 focuses on the formation of coalitions through sequences of binding bilateral agreements in voting situations. In voting situations, voters' incentive to form coalitions arises from their will to increase their power to affect the outcome of the voting process. If we model these situations by simple transferable utility games and assume that each voter's voting power is predicted by an appropriate power index, then the sequences of binding bilateral agreements which result in the formation of the grand coalition boils down to the notion of population monotonic path schemes for simple games. A path scheme for a simple game is composed of a path, i.e., a sequence of coalitions that is formed during the coalition formation process and a scheme, i.e., a payoff vector for each coalition in the path. A path scheme is called population monotonic if a player's payoff does not decrease as the path coalition grows. First, we focus on Shapley path schemes of simple games in which for every path coalition the Shapley value of the associated subgame provides the allocation at hand. We show that the existence of veto players, i.e., a subgroup of voters whose unanimous agreement is necessary to pass a decision, is required for the existence of population monotonic Shapley path schemes and vice versa. Moreover, a Shapley path scheme is population monotonic if and only if the first winning coalition that is formed along the path contains every minimal winning coalition of the game. We also show that each Shapley path scheme of a game is population monotonic if and only if the set of veto players of the game is a winning coalition. We further show how to extend these results to the probabilistic values, generalizations of the Shapley value introduced by Weber (1988).

1.3 Preliminaries on Cooperative Games

This section provides a brief introduction to some basic concepts in cooperative game theory.

Let $N = \{1, 2, ..., n\}$ be a finite set of players. A *coalition* is a set of players $S \subset N$ and N is sometimes called *the grand coalition*.

A cooperative transferable utility game (a TU-game) in characteristic function form is a pair (N, v) where v is a mapping, $v : 2^N \to \mathbb{R}$ with $v(\emptyset) = 0$. For any coalition $S \subset N$, $v(S)$ is called the *worth* of coalition S and is interpreted as the optimal monetary amount the coalition S can jointly generate on itself without any help of players in $N \backslash S$. Given a TU-game (N, v) and $S \in 2^N$, the *restriction of* v *to* S (a *subgame* of v) is denoted by $v_{|S}$ and is defined by $v_{|S}(T) = v(T)$ for every $T \subset S$. We denote the set of TU-games with player set N by $\mathcal{G}^N$.

A TU-game can reflect rewards or costs. A reward game will be denoted by a map v and a cost game will be denoted by a map c. The following definitions and properties refer to reward games.

A game $v \in \mathcal{G}^N$ is *monotonic* if $v(T) \geq v(S)$ for every $S, T \in 2^N$ with $S \subset T$; it is called *superadditive* if $v(S) + v(T) \leq v(S \cup T)$ for every $S, T \in 2^N$ with $T \cap S = \emptyset$ and it is called *convex* if a player's marginal contribution does not decrease if he joins a larger coalition, i.e., $v(S \cup \{i\}) - v(S) \leq v(T \cup \{i\}) - v(T)$ for every $i \in N$ and $S, T \subset N \backslash \{i\}$ with $S \subset T$. Equivalently, a TU-game $v \in \mathcal{G}^N$ is convex if $v(T) + v(S) \leq v(T \cup S) + v(T \cap S)$ for every $S, T \subset N$.

An element $x \in \mathbb{R}^N$ is called an *allocation*. We say that $x \in \mathbb{R}^N$ is an *imputation* if it satisfies the following two properties:

(i) *Efficiency*: $\sum_{i \in N} x_i = v(N)$.

(ii) *Individual rationality*: $x_i \geq v(\{i\})$ for every $i \in N$.

The *core* (Gillies, 1953) of a TU-game $v \in \mathcal{G}^N$ is denoted by $Core(v)$ and is defined as the set of efficient payoff vectors for which no coalition has an incentive to split off from the grand coalition, i.e., $Core(v) = \{x \in \mathbb{R}^N | \sum_{i \in N} x_i = v(N)$ and $\sum_{i \in S} x_i \geq v(S)$ for all $S \in 2^N\}$.

A *balanced set* $\mathcal{B}$ is a collection of subsets S of N with the property that there exist positive numbers λ_S, $S \in \mathcal{B}$, called *weights*, such that for each $i \in N$, we have that

$$\sum_{S \in \mathcal{B}: i \in S} \lambda_S = 1.$$

A game $v \in \mathcal{G}^N$ is called *balanced* if

$$\sum_{S \in \mathcal{B}} \lambda_S v(S) \leq v(N),$$

for every balanced set $\mathcal{B}$ and for every corresponding set of weights $(\lambda_S)_{S \in \mathcal{B}}$. Bondereva (1963) and Shapley (1967) independently proved that a TU-game $v \in \mathcal{G}^N$ is balanced if and only if its core is nonempty. It is well-known that convex games are balanced and hence have nonempty cores. A game is said to be *totally balanced* if every subgame is balanced, i.e., every subgame has a nonempty core.

An *order on the set of players* is a bijection $\sigma : N \to \{1, ..., n\}$. We denote the set of all orders on N by $\Pi(N)$. Given an order $\sigma \in \Pi(N)$ the set of predecessors of a player $i \in N$ with respect to σ is defined as $P(\sigma, i) = \{j \in N | \sigma(j) < \sigma(i)\}$. Similarly, the set of successors of i with respect to σ is defined as $S(\sigma, i) = \{j \in N | \sigma(j) > \sigma(i)\}$. We denote by $\bar{P}(\sigma, i)$ and $\bar{S}(\sigma, i)$, the sets $P(\sigma, i) \cup \{i\}$ and $S(\sigma, i) \cup \{i\}$, respectively.

The *marginal vector* of (N, v) corresponding to order $\sigma \in \Pi(N)$, $m^\sigma(v)$ is defined to be the vector with i^{th} coordinate equal to $v(\bar{P}(\sigma, i)) - v(P(\sigma, i))$.

A coalition $S \subset N$ is called *connected with respect to* σ if for all $i, j \in S$ and $k \in N$ such that $\sigma(i) < \sigma(k) < \sigma(j)$ it holds that $k \in S$. We denote with $con(\sigma)$ the set of coalitions that are connected with respect to σ. For a coalition S, $S \backslash \sigma$ is the set of σ-components of S, a σ-component of S being a maximally connected subset of S with respect to σ.

Let $\sigma \in \Pi(N)$. We call a TU-game (N, v) *σ-component additive* if it satisfies the following three conditions:

(i) $v(i) = 0$ for all $i \in N$,

(ii) (N, v) is superadditive,

(iii) $v(S) = \sum_{T \in S \backslash \sigma} v(T)$.

Le Breton et al. (1992) showed that $\sigma-$component additive games are balanced.

For any coalition $S \in 2^N \backslash \{\emptyset\}$, the *unanimity game* u_S is defined by $u_S(T) = 1$ if $S \subset T$ and $u_S(T) = 0$ for all other coalitions T. It is well known that every cooperative TU-game (N, v) can be written as a unique linear combination of unanimity games by

$$v = \sum_{S \subset N} \lambda_S u_S, \tag{1.1}$$

where $\lambda_S = \sum_{T \subset S} (-1)^{|S| - |T|} v(T)$ for every $S \subset N$ (cf. Shapley, 1953).

A function $F : \mathcal{G}^N \to \mathbb{R}^N$ is called a *value*. A value F is *efficient* if for all $v \in \mathcal{G}^N$, $\sum_{i\in N} F_i(v) = v(N)$. A player $i \in N$ is a *null player* in (N, v) if $v(S \cup \{i\}) = v(S)$ for every $S \subset N\backslash\{i\}$. F is said to satisfy the *null player property* if for any $v \in \mathcal{G}^N$ and any null player $i \in N$ in v, $F_i(v) = 0$. F is said to satisfy the *null player out property* (cf. Derks and Haller, 1999) if elimination of a null player does not affect the value of the other players, i.e., $F_i(v) = F_i(v_{|N\backslash\{j\}})$ for all $i, j \in N$ and all $v \in \mathcal{G}^N$ such that j is a null player in v and $i \neq j$.

The Shapley value (Shapley, 1953) is one of the most important solution concepts in cooperative game theory and has been studied extensively. The Shapley value of a game can be calculated by making use of the decomposition of a cooperative game into unanimity games. More precisely, given a cooperative game (N, v) such that $v = \sum_{S\subset N} \lambda_S u_S$, the Shapley value Φ assigns to agent $i \in N$

$$\Phi_i(v) = \sum_{S\subset N: i\in S} \frac{\lambda_S}{|S|},$$

for every $i \in N$.

An *allocation scheme* specifies how to distribute the worth of every coalition among its members. That is an allocation scheme for the game $v \in \mathcal{G}^N$ is a vector $(x^S)_{S\in 2^N\backslash\{\emptyset\}}$ such that

$$\sum_{i\in S} x_i^S = v(S)$$

for every $S \in 2^N\backslash\{\emptyset\}$. Naturally, every efficient value for TU-games defines an allocation scheme where the allocation for every coalition is obtained by applying the value to the corresponding subgame. The allocation scheme in which the Shapley value is used as an allocation vector is called the *Shapley allocation scheme*.

Sprumont (1990) introduced the notion of *population monotonic allocation schemes*. The notion of population monotonicity requires that the share allocated to every player increases as the coalition to which he belongs grows larger. Formally, an allocation scheme $(x^S)_{S\in 2^N\backslash\{\emptyset\}}$ for the game $v \in \mathcal{G}^N$ is *population monotonic* if

$$x_i^S \leq x_i^T$$

for every $S, T \subset N$ such that $S \subset T$ and $i \in S$.

Observe that if $(x^S)_{S\in 2^N\backslash\{\emptyset\}}$ is a population monotonic allocation scheme (PMAS), then x^S is a core element of the corresponding subgame $v_{|S}$ (cf. Sprumont, 1990) for every $S \in 2^N\backslash\{\emptyset\}$.

Let (N, c) be a cooperative cost game. The corresponding cost savings game (N, v) is defined by

$$v(S) = \sum_{i\in S} c(\{i\}) - c(S),$$

for every $S \subset N$.

Consequently, the properties and solution concepts for cooperative cost games can be derived from the definitions given above. The equivalent of a superadditive game for a cost game is a *subadditive* game. A cost game is *concave* if and only if its corresponding cost savings game is convex.

Chapter 2

Sequencing Situations and Cooperation

In this chapter, which is based on Çiftçi et al. (2008) and Çiftçi et al. (2009a), we consider cost allocation problems arising from specific types of sequencing situations.

Game theoretic analysis of the cost allocation problems arising from sequencing situations is initiated by Curiel et al. (1989) . This study considered one machine sequencing situations in which a finite number of agents, each having one job, are queued in front of a machine waiting for their jobs to be processed. Agents have linear cost functions and each group of agents is allowed to obtain cost savings by reordering their jobs. The problem of the distribution of the maximal cost savings is tackled by analyzing corresponding cooperative sequencing games. It was shown that these games are convex and hence balanced. Curiel et al. (1989) also proposed the equal gain splitting ($\mathcal{EGS}$) rule for these one-machine sequencing situations and provided an axiomatical characterization of this rule.

The following studies in this strand of literature have extended the basic model by considering ready times (Hamers et al., 1995) , due dates (Borm et al., 2002) , precedence relations (Hamers et al., 2005) and controllable processing times (van Velzen, 2006). In each of these papers, convexity of the corresponding class of games or of some special sub-classes is established. Curiel et al. (1993) considered a larger class of sequencing situations by allowing more general cost functions for the agents. It was shown that these games are not convex in general but core elements do exist. The β-rule was proposed as an extension of the $\mathcal{EGS}$ rule. This rule was shown to yield outcomes in the core of the corresponding games. Hamers et al. (1996) introduced the split core, a generalization of the $\mathcal{EGS}$ rule and provided an axiomatical characterization. Other papers have investigated multiple-machine sequencing situations. van den Nouweland et al. (1992) considered sequencing situations in flow-shops while

Hamers et al. (1999) and Slikker (2005) studied sequencing situations with multiple parallel machines.

The manufacturing/service systems considered in all studies above consist only of machines which can process no more than one job at a time. Although these models are realistic for many existing manufacturing systems, there are also various systems which include batch machines: machines that can simultaneously process multiple jobs (a batch) subject to the capacity of the machine. Typically, the capacity of a batch machine is related to the number, weight or size of jobs placed in a batch. Transportation of the semi-finished jobs from one machine to another or the delivery of the finished jobs to the customers/warehouses (cf. Lee and Chen, 2001) constitute very common examples of batch machines in manufacturing systems since transporters, i.e., the machines in these operations usually carry a batch of jobs at the same time. Other well-known examples include heat-treat ovens which can process multiple jobs with the same processing requirement (temperature, processing time etc.) simultaneously in a batch (cf. Lee et al., 1992) and also numerically controlled (NC) routers which cut a stack of metal sheets simultaneously during the cutting operation (cf. Ahmadi et al., 1992). We refer to Webster and Baker (1995) and Potts and Kovalyov (2000) for a review of the scheduling literature on batch sequencing.

In Section 2.1, we present a first game theoretical analysis of sequencing situations with batch machines. From Section 2.1.1 to Section 2.1.4, we consider batch sequencing situations. In a batch sequencing situation, a machine's capacity is the maximum number of jobs that can be processed at any one time and it is assumed that the time required to process the jobs in any batch is fixed and independent of the number of the jobs in the batch. We introduce in Section 2.1.1 batch sequencing situations with a single batch machine. These situations give rise to the class of so-called batch sequencing games. It is shown in Section 2.1.2 that these games are convex. In particular, we show that these games can be written as a non-negative linear combination of unanimity games. This observation also leads to an expression for the Shapley value of these games. We consider extensions of the equal gain splitting rule and the split core and provide axiomatic characterizations of these solution concepts by using efficiency, symmetry and consistency axioms along the lines of Suijs et al. (1997) and Gerichhausen and Hamers (2008).

Our definition of batch sequencing games follows the standard approach of Curiel et al. (1989) and assumes that the worth of a coalition is the maximal cost savings that it can obtain by "admissible" rearrangements of its members' positions in the queue. In an admissible rearrangement, two members of a coalition who have a non-member between them may not change positions. The rationale behind this assumption is that, since an initial order of jobs is established, the non-members who are placed in between two members of a coalition have the right to object to these two players jumping over them. Curiel et al. (1993) argued that the set of admissible rearrangements is too restrictive because a player may not object other players jumping over him if he is not hurt by this change. For the classical sequencing situations involving machines that can process only one job at a time, van Velzen and Hamers (2003) and Slikker (2006) considered relaxed sequencing games by considering specific relaxations of the set of admissible rearrangements and focused on the balancedness of the corresponding sequencing games. Similarly, we consider in Section 2.1.3 relaxed batch sequencing games which allow for rearrangements in which two members of a coalition can switch places by jumping over non-members. The non-members are not hurt by such rearrangements since the completion time of a batch is independent of the jobs placed in the batch. We show that these games can be written as a sum of specific assignment games (cf. Shapley and Shubik, 1972) and hence are balanced. It is also seen that relaxed batch sequencing games are not convex in general.

In Section 2.1.4 we consider batch sequencing situations in flow-shops which consist of a sequence of finitely many batch machines. We show that when each batch machine has the same batch size or when each batch machine has the same batch processing time, the associated cooperative batch sequencing game is equal to the batch sequencing game corresponding one particular batch machine in the flow-shop. Hence, the games corresponding to these two special classes are convex. However, it is also shown that the games corresponding to batch sequencing situations in flow-shops are not convex in general.

Many practical scheduling/sequencing situations involve set-up times, i.e., the intermediate delays between processing of successive jobs required to prepare the machine for the following job (e.g., delays required to change/adjust tooling or to clean a machine). A common assumption in the analysis of sequencing games as initiated by Curiel et al. (1989) is that no set-up time is incurred prior to the processing of the jobs. This assumption requires that set-up times are negligible or can be included in processing times. However, recent studies in the scheduling literature show that explicit incorporation of set-up times in scheduling decisions results in tremendous cost savings in various real world manufacturing/service systems that are characterized by significant set-up times. We refer to Allahverdi et al. (1999) and Allahverdi et al. (2008) for a review of the scheduling literature with set-up considerations.

In particular, so-called family sequencing problems have received considerable attention in the scheduling literature with set-up considerations[1]. These problems consider situations where the jobs can be classified into distinct families with respect to their production requirements such as the required tooling or container size. Since the members of the same family have the same production requirements, a job does not require a set-up when following another job from the same family, but a "family set-up time" is required when it follows a member of another family. An example of a specific application of family sequencing problems is a production line of colored plastics (cf. Potts and Van Wassenhove, 1992). Customer orders can be divided into color groups. A set-up is required when switching from a job of one color to a job of another color.

In Section 2.2, we analyze the cost allocation problems arising from family sequencing problems. Similarly to Curiel et al. (1989), we assume that there exists an initial order on the jobs and associate cooperative games with family sequencing problems by defining the worth of a coalition as the maximum savings it can obtain by means of an admissible rearrangement. We show that family sequencing games are balanced. This result is obtained by showing that the marginal which corresponds to the initial order belongs to the core of the game. It is also seen that the corresponding games need not be convex in general.

2.1 Batch Sequencing and Cooperation

In this section, we consider the cost allocation problems arising from sequencing situations with batch machines. Section 2.1.1 describes batch sequencing situations with a single batch machine. Section 2.1.2 introduces and analyzes the corresponding batch sequencing games. Section 2.1.3 analyzes relaxed batch sequencing games. Section 2.1.4 introduces and analyzes flow-shop batch sequencing situations and corresponding games.

2.1.1 Batch Sequencing Situations

In a batch sequencing situation a finite number of agents, each having one job, are queued in front of a single batch machine, waiting for their jobs to be processed. The set of agents is denoted by $N = \{1, 2, ..., n\}$. The machine can process one batch of jobs at one time. At most $z \in \mathbb{Z}_{++}$ jobs can be placed in one batch. Each batch is processed in t time units which is independent of the number of jobs placed in

[1]See Potts and Van Wassenhove (1992), Webster and Baker (1995) and Liaee and Emmons (1997) for a review of scheduling literature on family sequencing problems.

the batch. We assume that there is an initial order $\sigma_0 \in \Pi(N)$ on the agents before the processing of the machine starts. Specifically, $\sigma_0(i) = j$ means that agent i is in position j. For each agent $i \in N$, the costs of spending time in the system is assumed to be linear and the corresponding cost function $c_i : \mathbb{R}_+ \to \mathbb{R}$ is defined by $c_i(k) = \alpha_i k$ with $\alpha_i > 0$.

A *batch sequencing situation* is denoted by $\Gamma(N) = (N, \sigma_0, \alpha, z, t)$ where $\sigma_0 \in \Pi(N)$, $\alpha = (\alpha_i)_{i \in N} \in \mathbb{R}^N_{++}$, $z \in \mathbb{Z}_{++}$ and $t \in \mathbb{R}_{++}$.

In a batch sequencing situation $\Gamma(N) = (N, \sigma_0, \alpha, z, t)$, it can easily be observed that as long as there is a sufficient number of jobs to fill up a batch, it is profitable to run full batches of size z on the machine. Hence, the first z jobs are placed in the first batch to be processed by the machine, the following z jobs are placed in the second batch and so on. So, if the jobs are processed according to the order σ, then $\left\lceil \frac{\sigma(i)}{z} \right\rceil$ gives the number of the batch that the job of agent i is placed in[2]. Hence the completion time $C(\sigma, i)$ of the job of agent i with respect to σ is given by $C(\sigma, i) = \left\lceil \frac{\sigma(i)}{z} \right\rceil t$.

The total costs of all agents if the jobs are processed according to the order σ equal $\sum_{i \in N} \alpha_i C(\sigma, i)$. By reordering the jobs with respect to σ_0 the total costs can be reduced. Since the number of possible orderings of jobs is finite, there exists an order for which total costs are minimized. We call such an order *optimal*. The following proposition establishes the optimality of an HWCF (highest waiting cost first) order: a processing order in which jobs are processed in nonincreasing order of cost parameters α_i.

Proposition 2.1.1 *An HWCF order is optimal for every batch sequencing situation.*

Proposition 2.1.1 is a direct consequence of the independence of the batch processing time from the composition of each batch and can be proved by using a straightforward argument based on adjacent pairwise interchanges. Notice that the optimal order is unique up to reorderings of the jobs in the same batch and up to reorderings of the jobs of the agents with the same cost parameter.

2.1.2 Batch Sequencing Games

For a batch sequencing situation $\Gamma(N) = (N, \sigma_0, \alpha, z, t)$, the costs of a coalition S with respect to a processing order σ equal $\sum_{i \in S} \alpha_i C(\sigma, i)$. We want to determine the maximal cost savings of a coalition S when its members decide to cooperate. For this aim, we have to define which reorderings of the jobs of coalition S are admissible

[2] $\lceil q \rceil$ denotes the smallest integer which is greater than or equal to q for any real number $q \in \mathbb{R}$.

with respect to the initial order. In this section, we follow the approach of Curiel et al. (1989) and assume that an order σ is *admissible* for S with respect to σ_0 if $P(\sigma, j) = P(\sigma_0, j)$ for all $j \in N \backslash S$. The set of admissible reorderings of a coalition S is denoted by $\mathcal{A}(S)$.

The value of a coalition S is defined as the maximum cost savings coalition S can achieve by means of an admissible reordering. Formally, the batch sequencing game (N, v) corresponding to $\Gamma(N)$ is defined by

$$v(S) = \max_{\sigma \in \mathcal{A}(S)} \left\{ \sum_{i \in S} \alpha_i \, t \left(\left\lceil \frac{\sigma_0(i)}{z} \right\rceil - \left\lceil \frac{\sigma(i)}{z} \right\rceil \right) \right\}, \tag{2.1}$$

for every $S \subset N$.

Clearly, batch sequencing games are monotonic and superadditive. Notice that, by definition of an admissible ordering, a coalition S can obtain cost savings only by changing positions within σ_0-components. Hence, the value of a coalition S is equal to the sum of the values of its σ_0-components, i.e., $v(S) = \sum_{T \in S \backslash \sigma_0} v(T)$. Notice further that one-person coalitions can not generate any cost savings, i.e., $v(\{i\}) = 0$ for every $i \in N$. So, batch sequencing games are σ_0-component additive and hence they are balanced.

In the following, we will denote by $\sigma_S \in \Pi(N)$ an ordering which is attained from σ_0 by reordering the members in each σ_0-component of a coalition S with respect to the HWCF rule, i.e., $\sigma_S(i) = \sigma_0(i)$ for every $i \in N \backslash S$ and $\sigma_S(i) < \sigma_S(j)$ for every $T \in S \backslash \sigma_0$ and every $i, j \in T$ such that $\alpha_i > \alpha_j$. Clearly, $\sigma_S \in \mathcal{A}(S)$. Moreover, it follows by Proposition 2.1.1 and the σ_0-component additivity of batch sequencing games that σ_S is optimal for S, i.e.,

$$v(S) = \sum_{i \in S} \alpha_i \, t \left(\left\lceil \frac{\sigma_0(i)}{z} \right\rceil - \left\lceil \frac{\sigma_S(i)}{z} \right\rceil \right),$$

for every $S \subset N$.

Example 2.1.1 Consider the batch sequencing situation $\Gamma(N) = (N, \sigma_0, \alpha, z, t)$ where $N = \{1, 2, ..., 7\}$, $z = 2$, $\alpha = (1, 3, 4, 6, 8, 9, 12)$ and $t = 1$. Assume that σ_0 is given by $\sigma_0(i) = i$ for every $i \in N$ and consider the coalition $S = \{2, 3, 5, 6, 7\}$. It can easily be observed that $\sigma_S = (1, 3, 2, 4, 7, 6, 5)$. The orders σ_0 and σ_S are depicted in Figure 2.1.

Hence

$$\begin{aligned} v(S) &= \sum_{i \in S} \alpha_i \, t \left(\left\lceil \tfrac{\sigma_0(i)}{z} \right\rceil - \left\lceil \tfrac{\sigma_S(i)}{z} \right\rceil \right) \\ &= \alpha_2(1-2) + \alpha_3(2-1) + \alpha_5(3-4) + \alpha_6(3-3) + \alpha_7(4-3) = 5. \end{aligned}$$

$\diamond$

$$\boxed{1}\boxed{2} \quad \boxed{3}\boxed{4} \quad \boxed{5}\boxed{6} \quad \boxed{7} \quad \sigma_0$$

$$\boxed{1}\boxed{3} \quad \boxed{2}\boxed{4} \quad \boxed{7}\boxed{6} \quad \boxed{5} \quad \sigma_S$$

Figure 2.1: The orders σ_0 and σ_S in Example 2.1.1

In the following, we will show that batch sequencing games can be written as a nonnegative linear combination of unanimity games. For this aim, we first need the following notation and two lemmas.

Let $\Gamma(N) = (N, \sigma_0, \alpha, z, t)$ be a batch sequencing situation and let (N, v) be the corresponding batch sequencing game. For any coalition $T \subset N$, we will denote the member of T which stands in front of the other members of T with respect to σ_0 by $f(T)$ and the member which stands behind the other members of T by $l(T)$, i.e.,

$$f(T) = argmin_{i \in T}\, \sigma_0(i) \quad \text{and} \quad l(T) = argmax_{i \in T}\, \sigma_0(i).$$

For every agent $i \in N$ and an order $\sigma \in \Pi(N)$, the number of the batch that the job of agent i is placed in with respect to σ is denoted by $b_\sigma(i)$, i.e., $b_\sigma(i) = k$ if and only if $\left\lceil \frac{\sigma(i)}{z} \right\rceil = k$. Also we denote by (N, v_{σ_S}) the batch sequencing game corresponding to the batch sequencing situation $(N, \sigma_S, \alpha, z, t)$ for any $S \subset N$.

Lemma 2.1.1 *Let $\Gamma(N) = (N, \sigma_0, \alpha, z, t)$ be a batch sequencing situation and $T, S \subset N$ such that $S \subset T$. Then,*

$$v(T) = v(S) + v_{\sigma_S}(T).$$

Proof.

$$\begin{aligned} v(S) + v_{\sigma_S}(T) =& \sum_{i \in S} \alpha_i \left(\left\lceil \tfrac{\sigma_0(i)}{z} \right\rceil - \left\lceil \tfrac{\sigma_S(i)}{z} \right\rceil \right) + \\ &+ \left(\sum_{i \in S} \alpha_i \left(\left\lceil \tfrac{\sigma_S(i)}{z} \right\rceil - \left\lceil \tfrac{\sigma_T(i)}{z} \right\rceil \right) + \sum_{i \in T \setminus S} \alpha_i \left(\left\lceil \tfrac{\sigma_S(i)}{z} \right\rceil - \left\lceil \tfrac{\sigma_T(i)}{z} \right\rceil \right) \right) \\ =& \sum_{i \in T} \alpha_i \left(\left\lceil \tfrac{\sigma_0(i)}{z} \right\rceil - \left\lceil \tfrac{\sigma_T(i)}{z} \right\rceil \right) \\ =& v(T), \end{aligned}$$

where the second equality follows from the fact that $\sigma_0(i) = \sigma_S(i)$ for every $i \in T \setminus S$. □

We will also make use of the following lemma, proved by Borm et al. (2002), which explicitly describes the coefficients in the unique linear decomposition of a σ_0-component additive game into unanimity games.

Lemma 2.1.2 (Borm et al., 2002, Proposition 1) *Let (N, v) be a σ_0-component additive game and let $\sum_{S \subset N} \lambda_S u_S$ be the linear decomposition of (N, v) into unanimity games. Then, for every coalition $S \subset N$*

$$\lambda_S = \begin{cases} v(S) - v(S \backslash \{f(S)\}) - v(S \backslash \{l(S)\}) + v(S \backslash \{f(S), l(S)\}), & \textit{if } S \in con(\sigma_0), \\ 0, & \textit{otherwise.} \end{cases}$$

In the following proposition, we show that in case of batch sequencing games the formula provided by Lemma 2.1.2 boils down to an expression in terms of the differences between certain players' waiting cost parameters. More specifically, Proposition 2.1.2 reveals that the coefficient of a connected coalition S is equal to the sum of differences between the weight of the last player of a batch with respect to σ_S and the weight of the first player of the subsequent batch with respect to σ_S over all batches that are crossed by both the first and the last players of S with respect to σ_0 when the order changes from σ_0 to σ_S.

Proposition 2.1.2 *Let $\Gamma(N) = (N, \sigma_0, \alpha, z, t)$ be a batch sequencing situation. Let (N, v) be the corresponding batch sequencing game and let $\sum_{S \subset N} \lambda_S u_S$ be the linear decomposition of (N, v) into unanimity games. Then, for every $S \in con(\sigma_0)$*

$$\lambda_S = \begin{cases} \displaystyle\sum_{k: b_{\sigma_S}(l(S)) \leq k < b_{\sigma_S}(f(S))} t\,(\alpha_{\sigma_S^{-1}(kz)} - \alpha_{\sigma_S^{-1}(kz+1)}), & \textit{if } b_{\sigma_S}(l(S)) < b_{\sigma_S}(f(S)), \\ 0, & \textit{otherwise.} \end{cases}$$

Proof. Let $S \subset N$ be a connected coalition with respect to σ_0. Let us denote $f(S)$ by i, $l(S)$ by j and $S \backslash \{f(S), l(S)\}$ by S'. By using Lemmas 2.1.1 and 2.1.2, we obtain

$$\begin{aligned} \lambda_S &= v(S) - v(S \backslash \{i\}) - v(S \backslash \{j\}) + v(S'), \\ &= (v_{\sigma_{S \backslash \{i\}}}(S) + v(S \backslash \{i\})) - v(S \backslash \{i\}) - (v_{\sigma_{S'}}(S \backslash \{j\}) + v(S')) + v(S'), \\ &= v_{\sigma_{S \backslash \{i\}}}(S) - v_{\sigma_{S'}}(S \backslash \{j\}). \end{aligned}$$

Moreover, since $S \backslash \{i\}$ is already ordered optimally in $\sigma_{S \backslash \{i\}}$,

$$v_{\sigma_{S \backslash \{i\}}}(S) = \sum_{k: b_{\sigma_0}(i) \leq k < b_{\sigma_S}(i)} t\,(\alpha_{\sigma^{-1}_{S \backslash \{i\}}(kz+1)} - \alpha_i).$$

Similarly,

$$v_{\sigma_{S'}}(S \backslash \{j\}) = \sum_{k: b_{\sigma_0}(i) \leq k < b_{\sigma_{S \backslash \{j\}}}(i)} t\,(\alpha_{\sigma^{-1}_{S'}(kz+1)} - \alpha_i).$$

Hence,

$$\lambda_S = \sum_{k:b_{\sigma_0}(i)\leq k<b_{\sigma_S}(i)} t\,(\alpha_{\sigma^{-1}_{S\setminus\{i\}}(kz+1)} - \alpha_i) - \sum_{k:b_{\sigma_0}(i)\leq k<b_{\sigma_{S\setminus\{j\}}}(i)} t\,(\alpha_{\sigma^{-1}_{S'}(kz+1)} - \alpha_i). \quad (2.2)$$

Observe also that $\alpha_{\sigma^{-1}_{S\setminus\{i\}}(kz+1)} = \alpha_{\sigma^{-1}_{S'}(kz+1)}$ for every k such that $b_{\sigma_0}(i) \leq k < b_{\sigma_S}(j)$. So, equation (2.2) can be rewritten as

$$\lambda_S = \sum_{k:b_{\sigma_S}(j)\leq k<b_{\sigma_S}(i)} t\,(\alpha_{\sigma^{-1}_{S\setminus\{i\}}(kz+1)} - \alpha_i) - \sum_{k:b_{\sigma_S}(j)\leq k<b_{\sigma_{S\setminus\{j\}}}(i)} t\,(\alpha_{\sigma^{-1}_{S'}(kz+1)} - \alpha_i). \quad (2.3)$$

First assume that $b_{\sigma_S}(j) \geq b_{\sigma_S}(i)$. Clearly, $b_{\sigma_S}(i) = b_{\sigma_{S\setminus\{j\}}}(i)$ and $\alpha_{\sigma^{-1}_{S\setminus\{i\}}(kz+1)} = \alpha_{\sigma^{-1}_{S'}(kz+1)}$ for all k such that $b_{\sigma_S}(j) \leq k < b_{\sigma_S}(i)$. But, then $\lambda_S = 0$.

Assume now that $b_{\sigma_S}(j) < b_{\sigma_S}(i)$. Observe that $b_{\sigma_S}(i)$ is either equal to $b_{\sigma_{S\setminus\{j\}}}(i)$ or equal to $b_{\sigma_{S\setminus\{j\}}}(i) + 1$. Assume first that $b_{\sigma_S}(i) = b_{\sigma_{S\setminus\{j\}}}(i)$. Then, (2.3) can be rewritten as

$$\begin{aligned}
\lambda_S &= \sum_{k:b_{\sigma_S}(j)\leq k<b_{\sigma_S}(i)} t\,(\alpha_{\sigma^{-1}_{S\setminus\{i\}}(kz+1)} - \alpha_{\sigma^{-1}_{S'}(kz+1)}), \\
&= \sum_{k:b_{\sigma_S}(j)\leq k<b_{\sigma_S}(i)} t\,(\alpha_{\sigma^{-1}_{S}(kz)} - \alpha_{\sigma^{-1}_{S}(kz+1)}),
\end{aligned}$$

where the second equality follows from the fact that $\alpha_{\sigma^{-1}_{S\setminus\{i\}}(kz+1)} = \alpha_{\sigma^{-1}_{S}(kz)}$ and $\alpha_{\sigma^{-1}_{S'}(kz+1)} = \alpha_{\sigma^{-1}_{S}(kz+1)}$ for every k such that $b_{\sigma_S}(j) \leq k < b_{\sigma_S}(i)$.

Assume lastly that $b_{\sigma_S}(i) = b_{\sigma_{S\setminus\{j\}}}(i) + 1$. Let's denote $b_{\sigma_{S\setminus\{j\}}}(i)$ by $\bar{k}$. Then, obviously, $\alpha_i = \alpha_{\sigma^{-1}_{S}(\bar{k}z+1)}$. By equation (2.3)

$$\begin{aligned}
\lambda_S &= \sum_{k:b_{\sigma_S}(j)\leq k<\bar{k}} t\,(\alpha_{\sigma^{-1}_{S\setminus\{i\}}(kz+1)} - \alpha_{\sigma^{-1}_{S'}(kz+1)}) + t\,(\alpha_{\sigma^{-1}_{S\setminus\{i\}}(\bar{k}z+1)} - \alpha_i), \\
&= \sum_{k:b_{\sigma_S}(j)\leq k<\bar{k}} t\,(\alpha_{\sigma^{-1}_{S}(kz)} - \alpha_{\sigma^{-1}_{S}(kz+1)}) + t\,(\alpha_{\sigma^{-1}_{S}(\bar{k}z)} - \alpha_{\sigma^{-1}_{S}(\bar{k}z+1)}), \\
&= \sum_{k:b_{\sigma_S}(j)\leq k<b_{\sigma_S}(i)} t\,(\alpha_{\sigma^{-1}_{S}(kz)} - \alpha_{\sigma^{-1}_{S}(kz+1)}).
\end{aligned}$$

$\square$

We illustrate the expression provided by Proposition 2.1.2 in the following example.

Example 2.1.2 Take the batch sequencing situation $\Gamma(N)$ considered in Example 2.1.1. Let (N, v) be the batch sequencing game corresponding to $\Gamma(N)$ and consider

$$\boxed{1}\boxed{2} \quad \boxed{3}\boxed{4} \quad \boxed{5}\boxed{6} \quad \boxed{7} \;\; \sigma_0$$

$$\boxed{6}\boxed{5} \quad \boxed{4}\boxed{3} \quad \boxed{2}\boxed{1} \quad \boxed{7} \;\; \sigma_S$$

Figure 2.2: The orders σ_0 and σ_S in Example 2.1.2

the coalition $S = \{1, 2, ..., 6\}$. We will calculate λ_S, the coefficient corresponding to S in the decomposition of (N, v) into unanimity games. Obviously, $f(S) = 1$, $l(S) = 6$ and $\sigma_S = (6, 5, 4, 3, 2, 1, 7)$ (See Figure 2.2).

Observe that the job of agent 1 is processed in the third batch and the job of agent 6 is processed in the first batch with respect to σ_S. That is $b_{\sigma_S}(l(S)) = 1$ and $b_{\sigma_S}(f(S)) = 3$. Since $b_{\sigma_S}(l(S)) < b_{\sigma_S}(f(S))$, by Proposition 2.1.2

$$\begin{aligned}\lambda_S &= (\alpha_{\sigma_S^{-1}(2)} - \alpha_{\sigma_S^{-1}(3)}) + (\alpha_{\sigma_S^{-1}(4)} - \alpha_{\sigma_S^{-1}(5)}) \\ &= (\alpha_5 - \alpha_4) + (\alpha_3 - \alpha_2) = (8 - 6) + (4 - 3) = 3.\end{aligned}$$

The whole decomposition of (N, v) into unanimity games is provided below:

$$\begin{aligned} v &= u_{\{2,3\}} + 2u_{\{4,5\}} + 3u_{\{6,7\}} + 2u_{\{1,2,3\}} + 2u_{\{2,3,4\}} + 2u_{\{3,4,5\}} + u_{\{4,5,6\}} + u_{\{5,6,7\}} \\ &\quad + u_{\{1,2,3,4\}} + 3u_{\{2,3,4,5\}} + 2u_{\{3,4,5,6\}} + 5u_{\{4,5,6,7\}} + 4u_{N\setminus\{6,7\}} + 3u_{N\setminus\{1,7\}} \\ &\quad + 3u_{N\setminus\{1,2\}} + 3u_{N\setminus\{7\}} + 6u_{N\setminus\{1\}} + 5u_N.\end{aligned}$$

$\diamond$

Proposition 2.1.2 reveals that batch sequencing games are nonnegative combinations of unanimity games. Since unanimity games are convex, this establishes the convexity of the batch sequencing games. As a side result, we also obtain an expression of the Shapley value of batch sequencing games.

Theorem 2.1.1 *Let $\Gamma(N) = (N, \sigma_0, \alpha, z, t)$ be a batch sequencing situation and let (N, v) be the corresponding batch sequencing game. Then,*

(i) (N, v) is convex.

(ii) For all $i \in N$

$$\Phi_i(v) = \sum_{S \in con(\sigma_0), i \in S} \;\; \sum_{k : b_{\sigma_S}(l(S)) \leq k < b_{\sigma_S}(f(S))} \frac{t(\alpha_{\sigma_S^{-1}(kz)} - \alpha_{\sigma_S^{-1}(kz+1)})}{|S|}.$$

In the remainder of this section, we introduce and characterize a non-aggregated equal gain splitting ($\mathcal{EGS}$) solution and a non-aggregated split core ($\mathcal{SPC}$) for batch sequencing situations. Non-aggregated solutions, which are first introduced by Suijs et al. (1997) for classical sequencing situations, can be considered as a specification of all components of the total reward an agent obtains. In our setting, a non-aggregated solution Ψ is a map assigning to each batch sequencing situation $\Gamma(N)$ a matrix $W \in \mathbb{R}_+^{N\times N}$, where an element w_{ij} of W represents the nonnegative gain assigned to agent i for cooperating with agent j. The aggregated solution corresponding to $W \in \Psi(\Gamma(N))$ can be found by multiplying W with the vector e^N of all ones in $\mathbb{R}^N$.

Let $\Gamma(N) = (N, \sigma_0, \alpha, z, t)$ be a batch sequencing situation and (N, v) be the corresponding batch sequencing game. Also let $\sum_{S\subset N} \lambda_S u_S$ be the linear decomposition of (N, v) into unanimity games. For any $i, j \in N$, let $[i, j]$ be the set of all players in between i and j with respect to σ_0, i.e.,

$$[i,j] = \{k \in N |\min\{\sigma_0(i), \sigma_0(j)\} \le \sigma_0(k) \le \max\{\sigma_0(i), \sigma_0(j)\}\}\,.$$

The non-aggregated equal gain splitting solution $\mathcal{EGS}$ assigns to each batch sequencing situation $\Gamma(N)$ a solution $\mathcal{EGS}(\Gamma(N)) \in \mathbb{R}_+^{N\times N}$ such that

$$\mathcal{EGS}(\Gamma(N))_{ij} = \frac{\lambda_{[i,j]}}{2}$$

for every $i, j \in N$.

Example 2.1.3 Consider the batch sequencing situation of Example 2.1.1. Recall that the decomposition of the corresponding batch sequencing game into unanimity games is provided in Example 2.1.2. By making use of this decomposition, the equal gain splitting solution $\mathcal{EGS}(\Gamma(N))$ equals to

$$\mathcal{EGS}(\Gamma(N)) = \begin{bmatrix} 0 & 0 & 1 & \frac{1}{2} & 2 & \frac{3}{2} & \frac{5}{2} \\ 0 & 0 & \frac{1}{2} & 1 & \frac{3}{2} & \frac{3}{2} & 3 \\ 1 & \frac{1}{2} & 0 & 0 & 1 & 1 & \frac{3}{2} \\ \frac{1}{2} & 1 & 0 & 0 & 1 & \frac{1}{2} & \frac{5}{2} \\ 2 & \frac{3}{2} & 1 & 1 & 0 & 0 & \frac{1}{2} \\ \frac{3}{2} & \frac{3}{2} & 1 & \frac{1}{2} & 0 & 0 & \frac{3}{2} \\ \frac{5}{2} & 3 & \frac{3}{2} & \frac{5}{2} & \frac{1}{2} & \frac{3}{2} & 0 \end{bmatrix}$$

$\diamond$

The non-aggregated split core $\mathcal{SPC}$ assigns to each batch sequencing situation $\Gamma(N)$ a nonempty subset $\mathcal{SPC}(\Gamma(N)) \subset \mathbb{R}_+^{N\times N}$ such that

$$\mathcal{GS}(\Gamma(N))_{ij} + \mathcal{GS}(\Gamma(N))_{ji} = \lambda_{[i,j]}$$

for each gain splitting matrix $\mathcal{GS}(\Gamma(N)) \in \mathcal{SPC}(\Gamma(N))$ and every $i, j \in N$.

Now we introduce the notions of dummy agents and reduced batch sequencing situations that will be used for the axiomatizations of the $\mathcal{EGS}$ solution and the $\mathcal{SPC}$ for batch sequencing situations.

Let $\Gamma(N) = (N, \sigma_0, \alpha, z, t)$ be a batch sequencing situation. An agent $i \in N$ is called *a dummy agent in* $\Gamma(N)$ if $b(i) = b_{\sigma_S}(i)$ for every $S \subset N$ with $i \in S$, i.e., job of agent i stays in its initial batch no matter which coalition of agents i cooperates with. Roughly speaking, a batch sequencing situation reduced to a connected coalition $S \in con(\sigma_0)$ is a batch sequencing situation obtained when the agents outside S are replaced with dummy agents. Formally, *a reduced batch sequencing situation with respect to* S is described by $\Gamma_{|S}(N) = (N, \sigma_0, \beta, z, t)$ where $\beta = (\beta_i)_{i \in N} \in \mathbb{R}^N_+$ is such that $\beta_i = \alpha_i$ for every $i \in S$ and

$$\beta_j = \begin{cases} max_{i \in N}\ 2\alpha_i, & \text{if } j < f(S), \\ min_{i \in N}\ \frac{\alpha_i}{2}, & \text{if } j > l(S), \end{cases}$$

for every agent $j \in N \backslash S$.

Let Ψ be a non-aggregated solution for batch sequencing situations. We consider the following three properties of Ψ.

- **Efficiency**: Ψ is efficient if

$$\sum_{i,j \in N} \Psi(\Gamma(N))_{ij} = \sum_{i \in N} \alpha_i \left(C(\sigma_0, i) - C(\sigma_N, i) \right),$$

  for all batch sequencing situations $\Gamma(N)$.

- **Symmetry**: Ψ is symmetric if $\Psi(\Gamma(N))$ is a symmetric matrix for all batch sequencing situations $\Gamma(N)$.

- **Consistency**: Ψ is consistent if for all batch sequencing situations $\Gamma(N)$ and all $S \in con(\sigma_0)$ the following is satisfied:

$$\Psi(\Gamma(N))_{ij} = \Psi(\Gamma_{|S}(N))_{ij} \text{ for every } i, j \in S.$$

The efficiency axiom states that the total amount allocated to the agents is equal to the maximal total cost savings that the agents can jointly obtain. Symmetry states that the (extra) gain two agents can obtain is equally divided among the two agents. Consistency states that connected coalitions obtain the same division if they renegotiate on the basis of the same solution concept to the reduced situation with outside dummies.

In Theorem 2.1.2 we will characterize the $\mathcal{EGS}$ solution with the three properties mentioned above. For the proof, we need the following lemma which states that, for non-aggregate solutions that satisfy both efficiency and consistency, the total amount allocated to connected coalitions must be equal to the maximal total cost savings that these coalitions can achieve.

Lemma 2.1.3 *Let $\Gamma(N) = (N, \sigma_0, \alpha, z, t)$ be a batch sequencing situation and let Ψ be a non-aggregated solution for batch sequencing situations. If Ψ is both efficient and consistent, then*

$$\sum_{i,j \in S} \Psi(\Gamma(N))_{ij} = \sum_{i \in S} \alpha_i \left(C(\sigma_0, i) - C(\sigma_S, i) \right),$$

for every $S \in con(\sigma_0)$.

Proof. Suppose first that there exists $S \in con(\sigma_0)$ such that

$$\sum_{i,j \in S} \Psi(\Gamma(N))_{ij} > \sum_{i \in S} \alpha_i \left(C(\sigma_0, i) - C(\sigma_S, i) \right).$$

Now on the one hand efficiency of Ψ implies that

$$\sum_{i,j \in N} \Psi(\Gamma_{|S}(N))_{ij} = \sum_{i \in S} \alpha_i \left(C(\sigma_0, i) - C(\sigma_S, i) \right),$$

since all agents outside S are dummy agents in $\Gamma_{|S}(N)$. On the other hand, consistency of Ψ implies that

$$\begin{aligned}\sum_{i,j \in S} \Psi(\Gamma_{|S}(N))_{ij} &= \sum_{i,j \in S} \Psi(\Gamma(N))_{ij} \\ &> \sum_{i \in S} \alpha_i \left(C(\sigma_0, i) - C(\sigma_S, i) \right) = \sum_{i,j \in N} \Psi(\Gamma_{|S}(N))_{ij}.\end{aligned}$$

A contradiction, because

$$\sum_{i,j \in N} \Psi(\Gamma_{|S}(N))_{ij} \geq \sum_{i,j \in S} \Psi(\Gamma_{|S}(N))_{ij}$$

since $\Psi(\Gamma_{|S}(N))_{ij} \geq 0$ for every $i, j \in N$.

Now suppose that there exists $S \in con(\sigma_0)$ such that

$$\sum_{i,j \in S} \Psi(\Gamma(N))_{ij} < \sum_{i \in S} \alpha_i \left(C(\sigma_0, i) - C(\sigma_S, i) \right).$$

Clearly, $S \neq N$ and there exists $S_1, S_2 \in con(\sigma_0) \cup \{\emptyset\}$ such that $S_1 \cap S_2 = \emptyset$ and $N \backslash S = S_1 \cup S_2$. Observe that

$$\begin{aligned}\sum_{i,j \in N} \Psi(\Gamma(N))_{ij} &= \sum_{i \in N} \alpha_i \left(C(\sigma_0, i) - C(\sigma_N, i) \right) \\ &\geq \sum_{T \in \{S, S_1, S_2\}} \sum_{i \in T} \alpha_i \left(C(\sigma_0, i) - C(\sigma_T, i) \right),\end{aligned}$$

where the equality follows from Ψ being efficient and the inequality follows from the fact that $N = S \cup S_1 \cup S_2$ and $S \cap S_1 = S \cap S_2 = S_1 \cap S_2 = \emptyset$.

Then, since $\sum_{i,j \in S} \Psi(\Gamma(N))_{ij} < \sum_{i \in S} \alpha_i \left(C(\sigma_0, i) - C(\sigma_S, i)\right)$, we have either

$$\sum_{i,j \in S_1} \Psi(\Gamma(N))_{ij} > \sum_{i \in S_1} \alpha_i \left(C(\sigma_0, i) - C(\sigma_{S_1}, i)\right)$$

or

$$\sum_{i,j \in S_2} \Psi(\Gamma(N))_{ij} > \sum_{i \in S_2} \alpha_i \left(C(\sigma_0, i) - C(\sigma_{S_2}, i)\right).$$

However, as we have already shown, the existence of a coalition $S \in con(\sigma_0)$ with $\sum_{i,j \in S} \Psi(\Gamma(N))_{ij} > \sum_{i \in S} \alpha_i \left(C(\sigma_0, i) - C(\sigma_S, i)\right)$ leads to a contradiction. □

Theorem 2.1.2 *The $\mathcal{EGS}$ solution is the unique non-aggregated solution satisfying efficiency, symmetry and consistency.*

Proof. Obviously, $\mathcal{EGS}$ satisfies efficiency, symmetry and consistency.

Now let Ψ be a nonempty solution which satisfies efficiency, symmetry and consistency. Pick a batch sequencing situation $\Gamma(N) = (N, \sigma_0, \alpha, z, t)$. We will show that $\Psi(\Gamma(N))_{ij}$ is uniquely determined for every $i, j \in N$ with induction to the number $|\sigma_0(j) - \sigma_0(i)|$.

Pick $i, j \in N$. Assume first that $|\sigma_0(j) - \sigma_0(i)| = 0$. Then, $i = j$ and $\Psi(\Gamma(N))_{ij} = 0 = \mathcal{EGS}(\Gamma(N))_{ij}$ by Lemma 2.1.3. Now, assume that $\Psi(\Gamma(N))_{ij} = \mathcal{EGS}(\Gamma(N))_{ij}$ for every $i, j \in N$ with $|\sigma_0(j) - \sigma_0(i)| \leq k$ for some $k \in \{0, ..., n-2\}$. Pick $i, j \in N$ such that $|\sigma_0(j) - \sigma_0(i)| = k + 1$. Let $S = [i, j]$. Then,

$$\begin{aligned}\Psi(\Gamma(N))_{ij} + \Psi(\Gamma(N))_{ji} &= \sum_{i \in S} \alpha_i \left(C(\sigma_0, i) - C(\sigma_S, i)\right) - \sum_{(p,r) \in S \times S \backslash \{(i,j),(j,i)\}} \Psi(\Gamma(N))_{pr}, \\ &= \sum_{i \in S} \alpha_i \left(C(\sigma_0, i) - C(\sigma_S, i)\right) - \sum_{(p,r) \in S \times S \backslash \{(i,j),(j,i)\}} \mathcal{EGS}(\Gamma(N))_{pr},\end{aligned}$$

where the first equality follows from Lemma 2.1.3 and the second equality follows from the fact that $|\sigma_0(p) - \sigma_0(r)| \leq k$ for every $(p, r) \in S \times S \backslash \{(i, j), (j, i)\}$ and the induction assumption.

Then $\Psi(\Gamma(N))_{ij} + \Psi(\Gamma(N))_{ji}$ is determined uniquely. Consequently, by symmetry, Ψ is uniquely determined and hence $\Psi(\Gamma(N)) = \mathcal{EGS}(\Gamma(N))$. □

We show in the following proposition that, for every batch sequencing situation $\Gamma(N) = (N, \sigma_0, \alpha, z, t)$, the aggregated $\mathcal{EGS}$ solution $\mathcal{EGS}(\Gamma(N)) \cdot e^N$ is equal to the average of the two marginal vectors of the batch sequencing game corresponding to $\Gamma(N)$, the one belonging to the initial order σ_0 and the one belonging to the inverse order of σ_0.

Proposition 2.1.3 *Let* $\Gamma(N) = (N, \sigma_0, \alpha, z, t)$ *be a batch sequencing situation and let* (N, v) *be the corresponding batch sequencing game. Then,*

$$\mathcal{EGS}(\Gamma(N)) \cdot e^N = \frac{1}{2}\left(m^{\sigma_0}(v) + m^{\sigma_0^{-1}}(v)\right).$$

Proof. Let $\sum_{S \subset N} \lambda_S u_S$ be the linear decomposition of (N, v) into unanimity games and $i \in N$. Then,

$$\begin{aligned}
\frac{1}{2}\left(m_i^{\sigma_0}(v) + m_i^{\sigma_0^{-1}}(v)\right) &= \frac{1}{2}\left(v(\bar{P}(\sigma_0, i)) - v(P(\sigma_0, i)) + v(\bar{S}(\sigma_0, i)) - v(S(\sigma_0, i))\right) \\
&= \frac{1}{2}\left(\sum_{j \in \bar{P}(\sigma_0, i)} \lambda_{[i,j]} + \sum_{j \in \bar{S}(\sigma_0, i)} \lambda_{[i,j]}\right) = \frac{1}{2}\sum_{j \in N} \lambda_{[i,j]} \\
&= \mathcal{EGS}(\Gamma(N))_i \cdot e^N,
\end{aligned}$$

where the second equality follows from the fact that $\lambda_T = 0$ for every $T \subset N$ which is not connected with respect to σ_0. □

Comparing the aggregated version of the $\mathcal{EGS}$ rule with the Shapley value of batch sequencing games, we see that while the Shapley value takes the average of all the marginal vectors of the game, the aggregated version of the $\mathcal{EGS}$ rule takes the average of only two marginals. Since batch sequencing games are convex, each marginal of these games belongs to the core. Hence, as we highlight in the following theorem, $\mathcal{EGS}(\Gamma(N)) \cdot e^N$ belongs to the core of batch sequencing games.

Theorem 2.1.3 *Let* $\Gamma(N) = (N, \sigma_0, \alpha, z, t)$ *be a batch sequencing situation and let* (N, v) *be the corresponding batch sequencing game. Then,* $\mathcal{EGS}(\Gamma(N)) \cdot e^N$ *belongs to the core of* (N, v).

Now we turn to a characterization of the non-aggregated split core $\mathcal{SPC}$. Let $D(\Gamma(N))$ be the set of dummy players in a batch sequencing situation $\Gamma(N)$. Let Ψ be a non-aggregated solution concept that assigns to each batch sequencing situation $\Gamma(N)$ a non-empty subset of $\mathbb{R}_+^{N \times N}$. Consider the following three axioms for Ψ.

- **Efficiency**: Ψ is efficient if

$$\sum_{i,j \in N} W_{ij} = \sum_{i \in N} \alpha_i \left(C(\sigma_0, i) - C(\sigma_N, i)\right),$$

  for all batch sequencing situations $\Gamma(N)$ and all $W \in \Psi(\Gamma(N))$.

- **Consistency**: Ψ is consistent if for all batch sequencing situations $\Gamma(N)$, all $W \in \Psi(\Gamma(N))$ and all $S \in \mathit{con}(\sigma_0)$ the following is satisfied:

  There exists $W' \in \Psi(\Gamma_{|S}(N))$ such that $W_{ij} = W'_{ij}$ for every $i, j \in S$.

- **Converse consistency**: Ψ is converse consistent if for all batch sequencing situations $\Gamma(N)$ and all $W \in \mathbb{R}_+^{N\times N}$ with $\sum_{i,j\in N} W_{ij} = \sum_{i\in N} \alpha_i \left(C(\sigma_0, i) - C(\sigma_N, i)\right)$ the following statement is true:

  If for every $S \in con(\sigma_0)\backslash N$ with $D(\Gamma(N)) \subsetneq D(\Gamma_{|S}(N))$ there exists $W' \in \Psi(\Gamma_{|S}(N))$ such that $W_{ij} = W'_{ij}$ for every $i, j \in S$, then $W \in \Psi(\Gamma(N))$.

The efficiency and the consistency properties are the extensions of their previous definitions to the case of a correspondence. Converse consistency means that if each restriction of a feasible gain splitting matrix (a matrix which allocates the maximal cost savings to the agents) to a connected coalition of agents coincides with an element of the solution of the corresponding reduced situation, then this gain splitting matrix must also be an element of the solution of the original situation. Note that for the converse consistency property only reductions to connected coalitions that enlarge the set of dummy players are allowed.

In the following we will characterize the $\mathcal{SPC}$ solution with the properties mentioned above. For this aim, we first need the following lemma.

Lemma 2.1.4 *Let $\Gamma(N)$ be a batch sequencing situation and Ψ be a multi-valued non-aggregate solution for batch sequencing situations. If Ψ is both efficient and consistent, then it satisfies the following for every $W \in \Psi(\Gamma(N))$ and for every $S \in con(\sigma_0)$:*

$$\sum_{i,j\in S} W_{ij} = \sum_{i\in S} \alpha_i \left(C(\sigma_0, i) - C(\sigma_S, i)\right).$$

The proof runs similar with the proof of Lemma 2.1.3 and hence it is omitted.

Theorem 2.1.4 *The non-aggregated split core $\mathcal{SPC}$ is the unique non-empty solution satisfying efficiency, consistency and converse consistency.*

Proof. Obviously, $\mathcal{SPC}$ satisfies efficiency and consistency. For converse consistency, take a batch sequencing situation $\Gamma(N)$. Let (N, v) be the batch sequencing game corresponding to $\Gamma(N)$ and $\sum_{S\subset N} \lambda_S u_S$ be the linear decomposition of (N, v) to unanimity games. Pick a matrix $W \in \mathbb{R}_+^{N\times N}$ such that $\sum_{i,j\in N} W_{ij} = \sum_{i\in N} \alpha_i \left(C(\sigma_0, i) - C(\sigma_N, i)\right)$ and there exists $W' \in \mathcal{SPC}(\Gamma_{|S}(N))$ such that $W_{ij} = W'_{ij}$ for every $i, j \in S$ for every $S \in con(\sigma_0)\backslash N$ with $D(\Gamma(N)) \subsetneq D(\Gamma_{|S}(N))$.

Let $\underline{i} = argmin_{i\in N\backslash D(\Gamma(N))}\sigma_0(i)$ and $\bar{i} = argmax_{i\in N\backslash D(\Gamma(N))}\sigma_0(i)$. Let $S = [\underline{i}, \bar{i}]$, $S_1 = S\backslash\{\underline{i}\}$ and $S_2 = S\backslash\{\bar{i}\}$. Firstly, by Lemma 2.1.4, $W_{ij} = W_{ji} = 0$ for every $i \notin S$ and every $j \in N$. Now, consider the reduced batching situations $\Gamma_{|S_1}(N)$ and $\Gamma_{|S_2}(N)$.

Observe that both $D(\Gamma(N)) \subsetneq D(\Gamma_{|S_1}(N))$ and $D(\Gamma(N) \subsetneq D(\Gamma_{|S_2}(N)$. Then, there exists $W^1 \in \mathcal{SPC}(\Gamma_{|S_1}(N))$ and $W^2 \in \mathcal{SPC}(\Gamma_{|S_2}(N))$ such that $W_{ij} = W^1_{ij}$ for every $i, j \in S_1$ and $W_{ij} = W^2_{ij}$ for every $i, j \in S_2$. By consistency of $\mathcal{SPC}$ we know that

$$W_{ij} + W_{ji} = \begin{cases} W^1_{ij} + W^1_{ji} = \lambda_{[i,j]}, & \text{if } i, j \in S_1, \\ W^2_{ij} + W^2_{ji} = \lambda_{[i,j]}, & \text{if } i, j \in S_2. \end{cases}$$

Hence, for all pairs (i, j) except $(\underline{i}, \bar{i})$ and $(\bar{i}, \underline{i})$, $W_{ij} + W_{ji} = \lambda_{[i,j]}$. Then, Lemma 2.1.4 implies that $W_{\underline{i},\bar{i}} + W_{\bar{i},\underline{i}} = \lambda_{[\underline{i},\bar{i}]}$. Hence, $W \in \mathcal{SPC}(\Gamma(N))$.

Now let Ψ be a nonempty solution which satisfies efficiency, consistency and converse consistency. Pick $W \in \Psi(\Gamma(N))$. We will show that $W \in \mathcal{SPC}(\Gamma(N))$.

Pick $i, j \in N$ such that $||\sigma_0(j) - \sigma_0(i)|| = 0$. Then, $i = j$ and by Lemma 2.1.4 $W_{ij} = W_{ji} = 0$ and hence $W_{ij} + W_{ji} = \lambda_{[i,j]}$. Now, assume that $W_{ij} + W_{ji} = \lambda_{[i,j]}$ for every $i, j \in N$ with $||\sigma_0(j) - \sigma_0(i)|| \leq k$ for some $k \in \{0, ..., n-2\}$. Pick $i, j \in N$ such that $||\sigma_0(j) - \sigma_0(i)|| = k + 1$. Let $S = [i, j]$. Then,

$$\begin{aligned} W_{ij} + W_{ji} &= \sum_{i \in S} \alpha_i \left(C(\sigma_0, i) - C(\sigma_S, i) \right) - \sum_{(p,r) \in S \times S \setminus \{(i,j),(j,i)\}} W_{pr}, \\ &= \sum_{i \in S} \alpha_i \left(C(\sigma_0, i) - C(\sigma_S, i) \right) - \sum_{T \subsetneq S : T \in con(\sigma_0)} \lambda_T = \lambda_S, \end{aligned}$$

where the first equality follows from Lemma 2.1.4; the second equality follows from the fact that $||\sigma_0(p) - \sigma_0(r)|| \leq k$ for every $(p, r) \in S \times S \setminus \{(i, j), (j, i)\}$ and the induction assumption. Therefore $W \in \mathcal{SPC}(\Gamma(N))$. This establishes that $\Psi(\Gamma(N)) \subset \mathcal{SPC}(\Gamma(N))$.

Now we show that $\mathcal{SPC}(\Gamma(N)) \subset \Psi(\Gamma(N))$. Pick a batch sequencing situation $\Gamma(N)$ and $W \in \mathcal{SPC}(\Gamma(N))$. Since $\mathcal{SPC}$ is consistent, for every $S \in con(\sigma_0)$, there exists $Y^S \in \mathcal{SPC}(\Gamma_{|S}(N))$ such that $Y^S_{ij} = W_{ij}$ for every $i, j \in S$. We are going to show that $Y^S \in \Psi(\Gamma_{|S}(N))$ by induction on $|S|$. Assume first that $|S| = 1$, i.e., $S = \{i\}$ for some $i \in N$. Then, $\Psi(\Gamma_{|S}(N)) = [0] = Y^S$ since both $\mathcal{SPC}$ and Ψ are efficient.

Now assume that $Y^S \in \Psi(\Gamma_{|S}(N))$ for every $S \in con(\sigma_0)$ with $|S| < k$ and pick $S \in con(\sigma_0)$ with $|S| = k$. Then, pick $T \in con(\sigma_0)$. We denote a reduction of $\Gamma_{|S}(N)$ to T by $\Gamma_{|S,T}(N)$. Assume that $T \in con(\sigma_0)$ is such that $D(\Gamma_{|S}(N)) \subsetneq D(\Gamma_{|S,T}(N))$. First consider the case $T \cap S \neq \emptyset$. Then, firstly, $T \not\supset S$ since $D(\Gamma_{|S}(N)) \subsetneq D(\Gamma_{|S,T}(N))$. Since all agents in $N \setminus S$ are dummy agents in $\Gamma_{|S,T}(N)$, there exists $\Gamma_{|T \cap S}(N)$ a reduction of $\Gamma(N)$ to $T \cap S$ which is equal to $\Gamma_{|S,T}(N)$. But, then, since $|T \cap S| < k$, $Y^{T \cap S} \in \Psi(\Gamma_{|T \cap S}(N)) = \Psi(\Gamma_{|S,T}(N))$.

Now consider the case $T \subset N \setminus S$. Again since $\mathcal{SPC}$ is consistent, for every $T \in con(\sigma_0)$ there exists $Z^S \in \mathcal{SPC}(\Gamma_{|S,T}(N))$ such that $Z^S_{ij} = Y^S_{ij}$ for every $i, j \in T$. Then, since all agents in $N \setminus S$ are dummy agents in $\Gamma_{|S,T}(N)$, $Z^S = [0] = \Psi(\Gamma_{|S,T}(N))$.

Then for every $T \in con(\sigma_0)$ with $D(\Gamma_{|S}(N)) \subsetneq D(\Gamma_{|S,T}(N))$, there exists $Z \in \Psi(\Gamma_{|S,T}(N))$ such that $Y^S_{ij} = Z_{ij}$ for every $i, j \in T$. Then, since Ψ satisfies converse consistency, $Y^S \in \Psi(\Gamma_{|S}(N))$.

So, we proved that for every $S \in con(\sigma_0)$ such that $D(\Gamma(N)) \subsetneq D(\Gamma_{|S}(N))$, $Y^S \in \Psi(\Gamma_{|S}(N))$. Then, $W \in \Psi(\Gamma(N))$ since Ψ satisfies converse consistency. So we can conclude that $\mathcal{SPC}(\Gamma(N)) \subset \Psi(\Gamma(N))$. □

2.1.3 Relaxed Batch Sequencing Games

In Section 2.1.2, we associated with a one machine batch sequencing situation a cooperative game. The value of a coalition S was defined as the maximal cost savings which the coalition can obtain by admissible rearrangements, i.e., the rearrangements which reorder the jobs in each component (with respect to the initial order) of S. In this section, we introduce and analyze relaxed batch sequencing games, a class of games arising from batch sequencing situations where a coalition can also obtain savings by switching positions of agents that belong to different components of the coalition. We show that relaxed batch sequencing games are equal to the sum of specific assignment games (cf. Shapley and Shubik, 1972). Hence, these games are balanced. However, it is shown that relaxed batch sequencing games are not convex in general.

Let $\Gamma(N) = (N, \sigma_0, \alpha, z, t)$ be a batch sequencing situation. While defining the associated batch sequencing game in the previous section, we adapted the common assumption in the sequencing games literature and defined the value of a coalition S as the maximal cost savings that it can obtain by switching places within σ_0-components. The rationale behind this assumption is that a player not in S has the right to object to jobs in S jumping over him. In classical sequencing situations where the machine can process only one job at a time and the agents may require different processing times, a player not in S will certainly object to jobs in S jumping over him if he will end up with jobs in front of him that have a longer processing time than he had in the initial order. In batch sequencing situations however, the completion time of a player outside S is not affected if two players in different σ_0-components of S switch places since the completion time of a batch is independent of the jobs placed in the batch. Hence, the players outside S may not object to such rearrangements by S.

In this section, we follow this "optimistic" assumption and introduce relaxed batch sequencing games which allow for rearrangements in which players that belong to different σ_0-components of a disconnected coalition can switch places. Formally, an order $\sigma \in \Pi(N)$ is called *$\mathcal{R}$-admissible for coalition S* if $\sigma(j) = \sigma_0(j)$ for all $j \in N \backslash S$.

We denote the set of $\mathcal{R}$-admissible reorderings for a coalition S by $\mathcal{A}_{\mathcal{R}}(S)$. Obviously, $\mathcal{A}(S) \subset \mathcal{A}_{\mathcal{R}}(S)$.

By defining the value of a coalition S as the maximum cost savings coalition S can achieve by means of an $\mathcal{R}$-admissible reordering, we define *the relaxed batch sequencing game* $(N, v_{\mathcal{R}})$ corresponding to the batch sequencing situation $\Gamma(N) = (N, \sigma_0, \alpha, z, t)$ as follows:

$$v_{\mathcal{R}}(S) = max_{\sigma \in \mathcal{A}_{\mathcal{R}}(S)} \left\{ \sum_{i \in S} \alpha_i \left(C(\sigma_0, i) - C(\sigma, i) \right) \right\}, \tag{2.4}$$

for every $S \subset N$.

Example 2.1.4 Consider the batch sequencing situation given in Example 2.1.1 and the coalition $S = \{2, 3, 5, 6, 7\}$. It can easily be observed that $\sigma_S^* = (1, 7, 6, 4, 5, 3, 2)$ is an optimal rearrangement for S in $\mathcal{A}_{\mathcal{R}}(S)$. The orders σ_0 and σ_S^* are depicted in Figure 2.3.

$$\boxed{1}\boxed{2} \quad \boxed{3}\boxed{4} \quad \boxed{5}\boxed{6} \quad \boxed{7} \quad \sigma_0$$

$$\boxed{1}\boxed{7} \quad \boxed{6}\boxed{4} \quad \boxed{5}\boxed{3} \quad \boxed{2} \quad \sigma_S^*$$

Figure 2.3: The orders σ_0 and σ_S^* in Example 2.1.4

Then, $v_{\mathcal{R}}(S)$ can easily be calculated as

$$\begin{aligned} v_{\mathcal{R}}(S) &= \sum_{i \in S} \alpha_i \left(C(\sigma_0, i) - C(\sigma_S^*, i) \right) \\ &= \alpha_2(1-4) + \alpha_3(2-3) + \alpha_5(3-3) + \alpha_6(3-2) + \alpha_7(4-1) = 32. \end{aligned}$$

$\diamond$

Clearly, $(N, v_{\mathcal{R}})$ is monotonic and superadditive. Also observe that, for every coalition $S \subset N$, an order in which the players in S are arranged in order of non-increasing α_i is an optimal admissible order for S. We denote such an order by σ_S^*. That is, for every $S \subset N$, $\sigma_S^* \in \Pi(N)$ is such that $\sigma_S^* \in \mathcal{A}_{\mathcal{R}}(S)$ and $\sigma_S^*(i) < \sigma_S^*(j)$ for every $i, j \in S$ such that $\alpha_i > \alpha_j$.

In the following, we show that every relaxed batch sequencing game can be written as a sum of specific assignment games. First, we provide a brief review of assignment games.

Assignment games arise from bipartitite matching situations and are introduced by Shapley and Shubik (1972) to model two sided markets with transferable utility. Let the player set N be the union of two nonempty disjoint sets M_1 and M_2. For each $i \in M_1$ and $j \in M_2$, the value (the joint profit) of a matched pair (i, j) is defined by $a_{ij} \geq 0$. From this situation, *an assignment game* $(M_1 \cup M_2, w)$ is defined in the following way. The worth of a coalition $S \subset N$ is defined to be the maximum value that S can obtain by making suitable pairs from its members and pooling the profit. If $S \cap M_1 = \emptyset$ or $S \cap M_2 = \emptyset$, no suitable pairs can be made and hence the worth in this situation is zero. Formally, an assignment game $(M_1 \cup M_2, w)$ is defined by

$$w(S) = \max \left\{ \sum_{(i,j)\in\mu} a_{ij} | \mu \in \mathcal{M}(S \cap M_1, S \cap M_2) \right\} \tag{2.5}$$

for all $S \subset N$, where $\mathcal{M}(S \cap M_1, S \cap M_2)$ denotes the set of matchings between $S \cap M_1$ and $S \cap M_2$. Also, a matching is called *optimal* for S if it maximizes the gain of the coalition S.

Shapley and Shubik (1972) proved that the core of an assignment game $(M_1 \cup M_2, w)$ is nonempty.

Theorem 2.1.5 (Shapley and Shubik, 1972, Theorem 2) *Let $(M_1 \cup M_2, w)$ be an assignment game. Then $Core(w) \neq \emptyset$.*

Let $(N, \sigma_0, \alpha, z, t)$ be a batch sequencing situation. For any coalition $S \subset N$ and batch number $k \in \{1, ..., \lceil \frac{n}{z} \rceil - 1\}$, we denote by $L(S, k)$ the set of players in S which are placed in batch k or in the batches in front of batch k with respect to the initial order σ_0, i.e., $L(S, k) = \{i \in S | b_{\sigma_0}(i) \leq k\}$. We denote by $U(S, k)$ the set of players in S which are placed in batches behind batch k, i.e., $U(S, k) = \{i \in S | b_{\sigma_0}(i) > k\}$. Also let $a_{ij} = max\{0, \alpha_j - \alpha_i\}$ for every $i \in L(N, k)$ and every $j \in U(N, k)$. For every batch number $k \in \{1, ..., \lceil \frac{n}{z} \rceil - 1\}$, we call the assignment game $(L(N, k) \cup U(N, k), w^k)$ *the k^{th} assignment game associated with the batch sequencing situation* $(N, \sigma_0, \alpha, z, t)$.

It can readily be observed that for each assignment game $(L(N, k) \cup U(N, k), w^k)$ associated with the batch sequencing situation $(N, \sigma_0, \alpha, z, t)$, the characteristic function w^k can be expressed as follows:

$$w^k(S) = \sum_{i \in S: b_{\sigma_0}(i) > k \text{ and } b_{\sigma_S^*}(i) \leq k} \alpha_i - \sum_{i \in S: b_{\sigma_0}(i) \leq k \text{ and } b_{\sigma_S^*}(i) > k} \alpha_i, \tag{2.6}$$

for every $S \subset N$ and $k \in \{1, ..., \lceil \frac{n}{z} \rceil - 1\}$.

In the following theorem, we show that the relaxed batch sequencing game $(N, v_{\mathcal{R}})$ corresponding to a batch sequencing situation $\Gamma(N)$ is equal to the sum of the assignment games associated with $\Gamma(N)$. Since assignment games are balanced, this result also establihes the balancedness of relaxed batch sequencing games.

Theorem 2.1.6 *Let $(N, \sigma_0, \alpha, z, t)$ be a batch sequencing situation and let $(N, v_{\mathcal{R}})$ be the corresponding relaxed batch sequencing game. Also let $(L(N,k) \cup U(N,k), w^k)$ be the k^{th} assignment game obtained from $(N, \sigma_0, \alpha, z, t)$ for every batch number $k \in \{1, ..., \lceil \frac{n}{z} \rceil - 1\}$. Then,*

$$v_{\mathcal{R}}(S) = \sum_{k=1}^{\lceil \frac{n}{z} \rceil - 1} w^k(S),$$

for every $S \subset N$ and hence $(N, v_{\mathcal{R}})$ is balanced.

Proof. Let $S \subset N$. Then, by using equation (2.6)

$$\begin{aligned} \sum_{k=1}^{\lceil \frac{n}{z} \rceil - 1} w^k(S) &= \sum_{k=1}^{\lceil \frac{n}{z} \rceil - 1} \left(\sum_{i \in S : b_{\sigma_0}(i) > k \text{ and } b_{\sigma_S^*}(i) \leq k} \alpha_i - \sum_{i \in S : b_{\sigma_0}(i) \leq k \text{ and } b_{\sigma_S^*}(i) > k} \alpha_i \right), \\ &= \sum_{i \in S} \alpha_i (b_{\sigma_0}(i) - b_{\sigma_S^*}(i)) = v_{\mathcal{R}}(S). \end{aligned}$$

□

However, a relaxed batch sequencing game need not be convex in general. This is illustrated in the following example.

Example 2.1.5 Consider the following batch sequencing situation $(N, \sigma_0, \alpha, z, t)$ where N={1,2,3,4}, $\alpha = (1, 3, 6, 9)$, $z = 2$, $t = 1$ and $\sigma_0(i) = i$ for every $i \in N$. It can easily be observed that $v_{\mathcal{R}}(N) = 11$, $v_{\mathcal{R}}(\{1,2,3\}) = 5$, $v_{\mathcal{R}}(\{1,2,4\}) = 8$ and $v_{\mathcal{R}}(\{1,2\}) = 0$. So, $(N, v_{\mathcal{R}})$ is not convex:

$$v_{\mathcal{R}}(N) - v_{\mathcal{R}}(\{1,2,3\}) < v_{\mathcal{R}}(\{1,2,4\}) - v_{\mathcal{R}}(\{1,2\}).$$

◇

2.1.4 Flow-Shop Batch Sequencing Games

Flow-shop batch sequencing (FSBS) situations consist of a sequence of finitely many batch machines $B_1, B_2, ..., B_m$ and a finite number of agents $N = \{1, 2, ..., n\}$ each having one job to be processed in the order $B_1, B_2, ..., B_m$. Each batch machine B_k has a batch size of z_k and processes a batch in t_k time units independent of the number of jobs placed in the batch. As it is the case in batch sequencing situations on a single batch machine, we assume that there is an initial order σ_0 on the jobs before the processing of the jobs on the flow-shop begins. That is if agent i is in front of agent j in the queue, then at all machines in the flow-shop, the job of agent

i has to be processed before the job of agent j is processed or together with the job of agent j in the same batch. An FSBS situation as described above is denoted by $\Gamma(N,M) = (N, M, \sigma_0, \alpha, z, t)$ where $M = \{1, 2, ..., m\}$ and $m \in \mathbb{Z}_{++}$ is the number of batch machines in the flow-shop, $\sigma_0 \in \Pi(N)$, $\alpha = (\alpha_i)_{i \in N} \in \mathbb{R}^N_{++}$, $z \in \mathbb{Z}^M_{++}$ and $t \in \mathbb{R}^M_{++}$.

For every FSBS situation, a *production schedule* τ fixes for every agent i and for every machine k a starting time $T_{i,k}(\tau)$ of the job of agent i at machine k. A production schedule is *feasible* if it conforms to the batch capacity constraints of the machines, to the order of the flow-shop and to the order on the jobs. Formally, we call a production schedule feasible with respect to the order $\sigma \in \Pi(N)$ if it satisfies the following:

(i) $T_{i,k}(\tau) \geq 0$ for all $i \in N$ and all $k \in M$.

(ii) $T_{i,k}(\tau) + t_k \leq T_{i,k+1}(\tau)$ for all $i \in N$ and all $k \in M$.

(iii) If $\sigma(i) \leq \sigma(j)$, then $T_{i,k}(\tau) \leq T_{j,k}(\tau)$ for all $i, j \in N$ and all $k \in M$.

(iv) $|\{i \in N | T_{i,k}(\tau) = s\}| \leq z_k$ for all $k \in M$ and $s \geq 0$.

(v) If $T_{i,k}(\tau) \neq T_{j,k}(\tau)$, then $|T_{i,k}(\tau) - T_{j,k}(\tau)| \geq t_k$ for all $i, j \in N$ and all $k \in M$.

We denote by $F_{i,k}(\tau)$ the time at which machine k finishes the processing of the batch in which job i is placed, i.e., $F_{i,k}(\tau) = T_{i,k}(\tau) + t_k$. Then, the completion time $C_i(\tau)$ of job i under production schedule τ is $F_{i,m}(\tau)$. The total costs of all agents if the jobs are processed according to the production schedule τ equal $\sum_{i \in N} \alpha_i F_{i,m}(\tau)$.

Example 2.1.6 Consider the following *FSBS* $\Gamma(N,M) = (N, M, \sigma_0, \alpha, z, t)$ where $N = \{1, ..., 5\}$, $M = \{1, 2\}$, $\sigma_0(i) = i$ for every $i \in N$, $\alpha = (1, 7, 1, 1, 1)$, $z = (1, 2)$ and $t = (1, 5)$. In Figure 2.4, we depict two feasible production schedules with respect to the initial order σ_0: τ_0 and τ_1.

It can easily be observed that with respect to both schedules, every job is immediately processed by B_1 as soon as B_1 is available. If the jobs are processed with respect to production schedule τ_0, then $F_{1,1}(\tau_0) = T_{1,2}(\tau_0) = 1$. That is job 1's processing at B_2 starts as soon as its processing at B_1 ends. If the jobs are processed with respect to production schedule τ_1, then $F_{1,1}(\tau_1) = 1$ and $T_{1,2}(\tau_1) = T_{2,2}(\tau_1) = 2$. That is, with respect to production schedule τ_1, although job 1 is available for processing at B_2 at time one, it waits for one time unit for job 2 to become available for B_2 and at time two their processing by B_2 starts together in the same batch. ◇

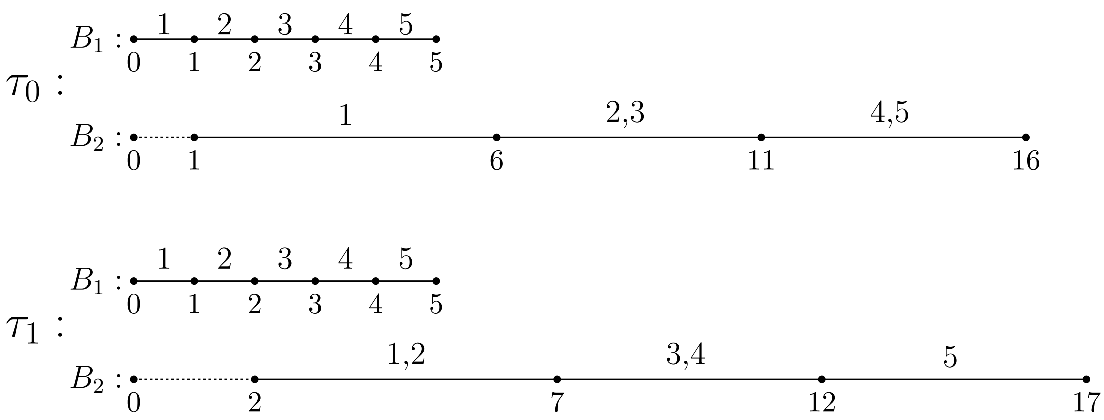

Figure 2.4: Two feasible schedules for the flow-shop situation in Example 2.1.6

Finding an optimal schedule for a general FSBS situation is still an open problem[3]. However, as we establish in the following proposition, there always exists an optimal schedule which processes the jobs according to the HWCF-sequence σ_N and runs full batches in the first batch machine.

Proposition 2.1.4 *For every FSBS situation* $\Gamma(N, M) = (N, M, \sigma_0, \alpha, z, t)$*, there exists an optimal schedule* τ *such that*

(i) τ *is a feasible schedule with respect to* σ_N*.*

(ii) τ *runs full batches of size* z_1 *in the first batch machine.*

We note that Proposition 2.1.4 can easily be proved by using a simple interchange argument.

Observe that there can be more than one production schedule which is feasible with respect to the initial order in an FSBS situation. We assume in this section that, in the initial case, i.e., when the agents do not cooperate, the corresponding *initial production schedule* is the one in which no agent waits for another because waiting for other jobs, without any further compensation, to get processed together in the same batch will only increase the costs of the agent. That is the initial production schedule is the one which is feasible with respect to σ_0 and satisfies the condition that every job which is ready to be processed at a machine is processed as soon as the machine is also available. We denote the initial production schedule in flow-shop

[3]For an FSBS situation with two machines and equal waiting time parameters, Ahmadi et al. (1992) showed that the problem of finding the optimal schedule can be solved within $O(N^3)$ operations.

batch sequencing situations by τ_0 (Note that the production schedule τ_0 in Example 2.1.6 indeed fits this description.).

Next, for a coalition of cooperating agents, we must decide on which production schedule rearrangements are admissible. We assume that a coalition S can choose any production schedule which is feasible with respect to an order $\sigma \in \mathcal{A}(S)$ as long as they do not harm the players outside S. Formally, a production schedule τ is admissible for coalition S if it satisfies the following two conditions:

(i) τ is a feasible production schedule with respect to an order in $\mathcal{A}(S)$.

(ii) $F_{i,m}(\tau) \leq F_{i,m}(\tau_0)$ for all $i \in N \backslash S$.[4]

The set of admissible production schedules for S is denoted by $\mathcal{APS}(S)$.

We define the value of a coalition S as the maximum cost savings coalition S can achieve by means of an admissible production schedule. Formally, a flow-shop batch sequencing game (N, w) corresponding to a flow-shop batch sequencing situation $\Gamma(N, M)$ is defined by

$$w(S) = \max_{\tau \in \mathcal{APS}(S)} \sum_{i \in S} \alpha_i (F_{i,m}(\tau_0) - F_{i,m}(\tau)), \tag{2.7}$$

for every $S \subset N$.

Using a simple interchange argument, it can be shown that there exists an optimal admissible production schedule for S which is feasible with respect to σ_S. Recall that σ_S is an ordering which is attained from σ_0 by reordering the members in each σ_0-component of a coalition S with respect to the HWCF rule.

Clearly, FSBS games are monotonic and superadditive. However, as illustrated by the following example, FSBS games are neither σ_0-component additive nor convex in general.

Example 2.1.7 Consider the flow-shop batch sequencing situation $\Gamma(N, M)$ given in Example 2.1.6 and the coalition $S = \{1, 3, 4, 5\}$. The initial production schedule, τ_0 is also given in Example 2.1.6. Notice that S can not create savings just by reordering

[4]Condition (ii) enables agents to wait for other agents' jobs in order to get processed together in the same batch. That is agents can create savings both by reordering their jobs and also by waiting for some other jobs. Another option for admissibility of a rearrangement could be to require that $F_{i,m}(\tau) = F_{i,m}(\tau_0)$ for all $i \in N \backslash S$. Under this more restrictive condition, waiting would no longer be possible in a flow-shop batch sequencing situation, i.e., the agents could create savings only by reordering their jobs. We want to remark that, under this more restrictive condition, the results obtained for single machine batch sequencing games can easily be extended to corresponding flow-shop batch sequencing games.

the jobs since $\alpha_i = \alpha_j$ for every $i, j \in S$. However, observe that τ_1 in Example 2.1.6 is an admissible production schedule for S (since $F_{2,2}(\tau_1) < F_{2,2}(\tau_0)$) and the cost savings obtained if S uses τ_1 is:

$$\begin{aligned}\sum_{i\in S}\alpha_i(F_{i,2}(\tau_0) - F_{i,2}(\tau_1)) &= \sum_{i\in S}(F_{i,2}(\tau_0) - F_{i,2}(\tau_1))\\ &= (6-7) + (11-12) + (16-12) + (16-17) = 1.\end{aligned}$$

So, when agent 1 waits for agent 2, the only agent outside S, agent 2 profits from an earlier completion time, agents 3 and 5 are harmed indirectly but agent 4 profits. As a result, S could obtain cost savings of 1. Actually, τ_1 is the optimal production schedule for S, i.e., $w(S) = 1$.

For $T \in 2^N \backslash \{S\}$ one finds that: $w(T) = 30$ for every $T \supset \{1, 2\}$ and $w(T) = 0$ otherwise.

Observe that this FSBS game is not convex:

$$0 = w(N) - w(\{1, 2, 3, 4\}) < w(\{1, 3, 4, 5\}) - w(\{1, 3, 4\}) = 1.$$

(N, w) is not σ_0-component additive either: $\{1\}$ and $\{3, 4, 5\}$ are the σ_0-components of $\{1, 3, 4, 5\}$, but

$$1 = w(\{1, 3, 4, 5\}) \neq w(\{1\}) + w(\{3, 4, 5\}) = 0.$$

$\diamond$

In the following, we will examine two particular FSBS situations: situations where all batch machines have the same batch size and situations where all batch machines have the same batch processing time. First, it is shown that, in both of these situations, an optimal order for a coalition of agents can be obtained by reordering the jobs of the agents. That is although waiting for other jobs to produce savings is allowed in our model, in these FSBS situations it need not be employed by the coalitions to obtain maximal cost savings. Second, it is shown that the FSBS games arising from these situations are equal to the game arising from the "bottleneck" machine in the flow-shop: the machine with the highest batch processing time when all machines have the same batch capacity and the machine with the minimum batch capacity when all machines have the same batch processing time.

We need the following notation in order to present our results. Consider an $FSBS$ situation $\Gamma(N, M) = (N, M, \sigma_0, \alpha, z, t)$. We denote by P_k the sum of the first k machines' batch processing times, by $\bar{t}$ the maximum batch processing time and by $\bar{z}$ the minimum batch size, i.e., $P_k = \sum_{p=1}^{k} t_p$, $\bar{t} = \max\{t_k | k \in M\}$ and $\bar{z} = \min\{z_k | k \in$

$M\}$. Lastly, we denote by τ_σ the production schedule which is obtained from τ_0 only by reordering the agents with respect to σ. That is

$$T_{\sigma^{-1}(p),k}(\tau_\sigma) = T_{\sigma_0^{-1}(p),k}(\tau_0)$$

for every $p \in \{1, 2, ..., n\}$ and $k \in M$.

Proposition 2.1.5 *Let $\Gamma(N, M) = (N, M, \sigma_0, \alpha, z, t)$ be an FSBS situation. If $z_k = z$ for every $k \in M$ or if $t_k = t$ for every $k \in M$, then τ_{σ_S} is an optimal production schedule for every $S \subset N$.*

Proof. First assume that $z_k = z$ for every $k \in M$ and consider the initial production schedule τ_0. Observe that with respect to τ_0 first z jobs in the initial order σ_0 are processed together in the first batch in each machine; the second z jobs in σ_0 are processed together in the second batch in each machine and so on. Since the initial production schedule τ_0 runs full batches in each machine in the flow-shop, waiting for other jobs is not an option for the players. We know that there exists an optimal production schedule for every coalition S which is feasible with respect to σ_S. Then, clearly, τ_{σ_S} is an optimal production schedule for any coalition $S \subset N$.

Now assume that $t_k = t$ for every $k \in M$ and consider the initial production schedule τ_0. Observe that τ_0 may run batches with less jobs than full capacity and the jobs in these batches have the option to wait for other jobs to get processed together in the same batch. However, observe that it is not possible to decrease the completion times of the other jobs by waiting for them, because the batch processing time of each machine is the same. That is the coalitions can not produce savings through waiting for other jobs. We know that there exists an optimal production schedule which is feasible with respect to σ_S for every coalition S. Then, since waiting is not profitable, τ_{σ_S} is an optimal production schedule for any coalition $S \subset N$. $\square$

In Theorem 2.1.7 we show that in both of these FSBS situations the corresponding FSBS games are equal to the batch sequencing game corresponding to the bottleneck machine in the flow-shop. For the proof, we need the following lemma which states that in both of these FSBS situations the completion time of a job is determined up to a constant by the bottleneck machine.

Lemma 2.1.5 *Let $\Gamma(N, M) = (N, M, \sigma_0, \alpha, z, t)$ be an FSBS situation.*

(i) If $z_k = z$ for every $k \in M$, then $F_{i,m}(\tau_\sigma) = P_m + (\left\lceil \frac{\sigma(i)}{z} \right\rceil - 1)\bar{t}$ for every $i \in N$ and every $\sigma \in \Pi(N)$.

(ii) If $t_k = t$ for every $k \in M$, then $F_{i,m}(\tau_\sigma) = mt + (\left\lceil \frac{\sigma(i)}{\bar{z}} \right\rceil - 1)t$ for every $i \in N$ and every $\sigma \in \Pi(N)$.

Proof. (i). Let $\sigma \in \Pi(N)$ and let $i \in N$ be such that $\left\lceil \frac{\sigma(i)}{z} \right\rceil = 1$. Clearly, i is processed in the first batch by each machine and $F_{i,k}(\tau_\sigma) = P_k$ for every $k \in M$. Now, pick $j \in N$ with $\left\lceil \frac{\sigma(j)}{z} \right\rceil = 2$, i.e., j is processed in the second batch by each machine. Let $\bar{t}_{[1,k]}$ be the maximum batch processing time among the first k machines, i.e., $\bar{t}_{[1,k]} = \max\{t_l | l \in \{1, 2, ..., k\}\}$. We will show, by induction on k, that

$$F_{j,k}(\tau_\sigma) = F_{i,k}(\tau_\sigma) + \left(\left\lceil \frac{\sigma(j)}{z} \right\rceil - 1\right) \bar{t}_{[1,k]} = P_k + \bar{t}_{[1,k]}.$$

When $k = 1$, the assertion holds trivially. So, assume that the assertion holds for every $k < l$. Then, $F_{j,l-1}(\tau_\sigma) = P_{l-1} + \bar{t}_{[1,l-1]}$.

Now, if $t_l < \bar{t}_{[1,l-1]}$, then $\bar{t}_{[1,l]} = \bar{t}_{[1,l-1]}$ and

$$F_{j,l-1}(\tau_\sigma) = P_{l-1} + \bar{t}_{[1,l-1]} > P_{l-1} + t_l = P_l = F_{i,l}(\tau_\sigma),$$

i.e., the second batch's processing at machine $l - 1$ finishes after the first batch's processing finishes at machine l. Hence, the second batch immediately starts to be processed by machine l at time $F_{j,l-1}(\tau_\sigma)$. So,

$$F_{j,l}(\tau_\sigma) = F_{j,l-1}(\tau_\sigma) + t_l = P_{l-1} + \bar{t}_{[1,l-1]} + t_l = P_l + \bar{t}_{[1,l]}.$$

If $t_l \geq \bar{t}_{[1,l-1]}$, then $\bar{t}_{[1,l]} = t_l$. Also we have that

$$F_{j,l-1}(\tau_\sigma) = P_{l-1} + \bar{t}_{[1,l-1]} \leq P_{l-1} + t_l = F_{i,l}(\tau_\sigma).$$

That is the second batch's processing at machine $l-1$ finishes before the first batch's processing finishes at machine l. Hence, the second batch starts to be processed by machine l at time $F_{i,l}(\tau_\sigma)$. So, $F_{j,l}(\tau_\sigma) = F_{i,l}(\tau_\sigma) + t_l = P_l + t_l = P_l + \bar{t}_{[1,l]}$.

Now, one can repeat the whole argument given above for a job which is processed in the third batch by each machine, then for the fourth batch and so on to prove that $F_{i,m}(\tau_\sigma) = P_m + (\left\lceil \frac{\sigma(i)}{z} \right\rceil - 1)\bar{t}$ for every $i \in N$.

(ii) can be proven similarly. □

Theorem 2.1.7 *Let $\Gamma(N, M) = (N, M, \sigma_0, \alpha, z, t)$ be an FSBS situation and (N, w) be the corresponding flow-shop batch sequencing game.*

(i) Let $z_k = z$ for every $k \in M$. Define $\Gamma(N) = (N, \sigma_0, \alpha, z, \bar{t})$ to be the batch sequencing situation corresponding to the bottleneck batch machine and let (N, v) be the corresponding batch sequencing game. Then, $v = w$.

(ii) Let $t_k = t$ for every $k \in M$. Define $\Gamma(N) = (N, \sigma_0, \alpha, \bar{z}, t)$ to be the batch sequencing situation corresponding to the bottleneck batch machine and let (N, v) be the corresponding batch sequencing game. Then, $v = w$.

Proof. (i). We know by Lemma 2.1.5 that

$$F_{i,m}(\tau_0) = P_m + (\left\lceil \frac{\sigma_0(i)}{z} \right\rceil - 1)\bar{t} \text{ and } F_{i,m}(\tau_{\sigma_S}) = P_m + (\left\lceil \frac{\sigma_S(i)}{z} \right\rceil - 1)\bar{t},$$

for every $S \subset N$ and every $i \in N$. Moreover, we know by Proposition 2.1.5 that in $\Gamma(N, M)$ τ_{σ_S} is an optimal schedule for every $S \subset N$. Then,

$$\begin{aligned} w(S) &= \sum_{i \in S} \alpha_i (F_{i,m}(\tau_0) - F_{i,m}(\tau_{\sigma_S})) \\ &= \sum_{i \in S} \alpha_i \left((P_m + (\left\lceil \frac{\sigma_0(i)}{z} \right\rceil - 1)\bar{t}) - (P_m + (\left\lceil \frac{\sigma_S(i)}{z} \right\rceil - 1)\bar{t}) \right) \\ &= \sum_{i \in S} \alpha_i \, \bar{t} \, \left(\left\lceil \frac{\sigma_0(i)}{z} \right\rceil - \left\lceil \frac{\sigma_S(i)}{z} \right\rceil \right) = v(S), \end{aligned}$$

for every $S \subset N$.

(ii) can be proven similarly. $\square$

2.2 Family Sequencing and Cooperation

In this section, we consider cost allocation problems arising from family sequencing situations. In a family sequencing situation, the jobs can be partitioned into distinct families. No set-up is required between the jobs of the same family. However, the family set-up time is required when a job is preceded by a job of a different family or if there is no preceding job. Section 2.2.1 formally describes family sequencing situations. Section 2.2.2 introduces and analyzes the corresponding family sequencing games. We prove that family sequencing games are balanced by showing that a specific marginal vector belongs to the core of these games. It is also seen that these games need not be convex even under specific restrictions of the model considered.

2.2.1 Family Sequencing Situations

In this section, we consider a one machine sequencing situation in which a finite number of agents, each having one job, are queued in front of a machine, waiting for their jobs to be processed. The machine in the situation is of classical type which can handle at most one job at a time. The set of agents is denoted by $N = \{1, 2, ..., n\}$.

The jobs can be partitioned into f families with respect to their production requirements. Let $F = \{1, 2, ..., f\}$ be the set of families. A family function $\mathcal{F} : N \to F$ associates to each agent $i \in N$ the family $\mathcal{F}(i)$ that his job belongs to. We denote with n_k the number of agents whose jobs are in family k. It is assumed that there is an initial processing order σ_0 on the agents before the processing of the machine starts. If a job in family k follows a job of the same family, then it does not require a set-up. However, the family set-up time s_k is required if it is preceded by a job of a different family or if there is no preceding job. Observe that the set-up times are *sequence independent*, i.e., they are independent of the family of the preceding job. We assume that each job of the same family requires the same processing time which is denoted by p_k for every family $k \in F$. For each agent $i \in N$, the costs of spending time in the system is assumed to be linear in the completion time of the job and the corresponding cost function $c_i : \mathbb{R}_+ \to \mathbb{R}$ is defined by $c_i(t) = \alpha_i t$ with $\alpha_i > 0$. Lastly, we assume that the agents having jobs of the same family have the same cost parameter, i.e., if $\mathcal{F}(i) = \mathcal{F}(j)$, then $\alpha_i = \alpha_j$ for every $i, j \in N$.

A one machine sequencing situation as described above is called a *family sequencing situation* and is denoted by $\Sigma(N) = (N, F, \mathcal{F}, \sigma_0, s, p, \alpha)$ where $\sigma_0 \in \Pi(N)$ and $s, p, \alpha \in \mathbb{R}^F_{++}$.

In a family sequencing situation $\Sigma(N) = (N, F, \mathcal{F}, \sigma_0, s, p, \alpha)$, the completion time $C(\sigma, i)$ of the job of agent i when the jobs are processed according to the order σ is given by

$$C(\sigma, i) = \sum_{j \in \bar{P}(\sigma,i)} \left(x_{\sigma,j} s_{\mathcal{F}(j)} + p_{\mathcal{F}(j)} \right),$$

where $x_{\sigma,j}$ equals 1 if the job j will require a set-up when jobs are processed with respect to σ and 0 otherwise. Observe that $C(\sigma, i)$ is the sum of processing times for the first $\sigma(i)$ jobs plus any set-ups that occurred.

The total costs of all agents if the jobs are processed according to the order σ equal $\sum_{i \in N} \alpha_{\mathcal{F}(i)} C(\sigma, i)$. By reordering the jobs with respect to σ_0 the total costs can be reduced. We call an order *optimal* if it minimizes the total costs. It was proven by Santos and Magazine (1985) and, independently, by Dobson et al. (1987) that a *highest urgency comes first* (HUCF) order, an order which processes the jobs of the same family together as a group (consecutively) and processes these family groups in nonincreasing order of the family-specific urgency index u_k defined by $u_k = \frac{n_k \alpha_k}{s_k + n_k p_k}$ is optimal for family sequencing situations.

Theorem 2.2.1 *For every family sequencing situation an HUCF order is optimal.*

2.2.2 Family Sequencing Games

For a family sequencing situation $\Sigma(N) = (N, F, \mathcal{F}, \sigma_0, s, p, \alpha)$, the costs of a coalition T with respect to a processing order σ equal $\sum_{i\in T} \alpha_{\mathcal{F}(i)} C(\sigma, i)$. We want to determine the maximal cost savings of a coalition T when its members decide to cooperate. For this aim, we have to define which reorderings of the jobs of coalition T are admissible with respect to the initial order. We assume that an order $\sigma \in \Pi(N)$ is admissible for a coalition T with respect to σ_0 if it satisfies the following two conditions:

(i) $P(\sigma, j) = P(\sigma_0, j)$ for all $j \in N\backslash T$.

(ii) $C(\sigma, i) \leq C(\sigma_0, i)$ for all $i \in N\backslash T$.

Condition (i) is the standard admissibility requirement that a coalition T can produce cost savings only by changing positions within σ_0-components. However, in a family sequencing situation, a coalition may hurt the players outside the coalition by reordering its players in a way that increases the total time required for set-ups. Hence, we also adopt condition (ii) which guarantees that T cannot not harm the players outside T. The set of admissible reorderings of a coalition T is denoted by $\mathcal{A}(T)$.

The value of a coalition T is defined as the maximum cost savings coalition T can achieve by means of an admissible reordering. Formally, the family sequencing game (N, v) corresponding to a family sequencing situation $\Sigma(N) = (N, F, \mathcal{F}, \sigma_0, s, p, \alpha)$ is defined by

$$v(T) = \max_{\sigma\in\mathcal{A}(T)} \left\{ \sum_{i\in T} \alpha_{\mathcal{F}(i)} \ (C(\sigma_0, i) - C(\sigma, i)) \right\}, \tag{2.8}$$

for every $T \subset N$.

The family sequencing games are illustrated in the following example.

Example 2.2.1 Consider the family sequencing situation $(N, F, \mathcal{F}, \sigma_0, s, p, \alpha)$ with $N = \{1, 2, 3, 4, 5\}$ and $F = \{1, 2\}$. Assume that $\mathcal{F}(1) = \mathcal{F}(4) = \mathcal{F}(5) = 1$ and $\mathcal{F}(2) = \mathcal{F}(3) = 2$. Assume also that σ_0 is given by $\sigma_0(i) = i$ for every $i \in N$, $s = (1, 5)$, $p = (4, 2)$ and $\alpha = (1, 1)$. Then the urgencies for the families are $u_1 = \frac{3}{13}$ and $u_2 = \frac{2}{9}$, respectively. Hence, an HUCF order processes first the jobs in family 1 and then the jobs in family 2. In Figure 2.5, we depict the production schedules corresponding to an HUCF order σ_N and σ_0.

Consider now the coalition $T = \{1, 2, 3\}$. Observe that the order $\sigma_T = (2, 3, 1, 4, 5)$ is an admissible order for T with respect to σ_0: $P(\sigma_T, j) = P(\sigma_0, j)$ for all $j \in N\backslash T$ and both $C(\sigma_T, 4) < C(\sigma_0, 4)$ and $C(\sigma_T, 5) < C(\sigma_0, 5)$. Figure 2.5 also depicts the

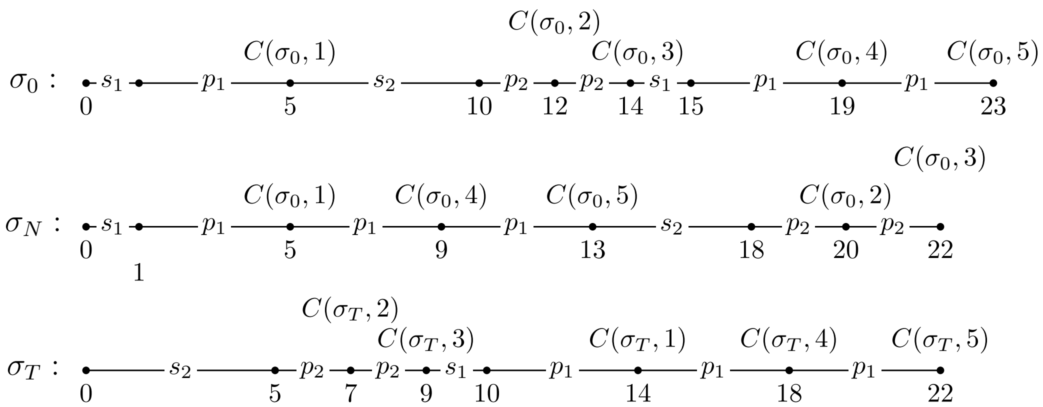

Figure 2.5: The production schedules corresponding to the orders σ_0, σ_N and σ_T in Example 2.2.1

production schedule corresponding to order σ_T. The cost savings obtained if T uses σ_T is:

$$\begin{aligned}\sum_{i\in T}\alpha_{\mathcal{F}(i)}(C(\sigma_0,i)-C(\sigma_T,i)) &= \sum_{i\in T}(C(\sigma_0,i)-C(\sigma_T,i))\\ &= (5-14)+(12-7)+(14-9)=1.\end{aligned}$$

Actually, σ_T is the optimal processing order for T, i.e., $v(T)=1$. Notice that σ_T processes the jobs in family 2 first although with respect to the optimal order for the grand coalition jobs in family 1 are processed first.

Consider now the coalition $Q=\{1,2,3,5\}$. Clearly, σ_T is an admissible order for Q. The cost savings obtained if Q uses σ_T is:

$$\begin{aligned}\sum_{i\in Q}\alpha_{\mathcal{F}(i)}(C(\sigma_0,i)-C(\sigma_T,i)) &= \sum_{i\in Q}(C(\sigma_0,i)-C(\sigma_T,i))\\ &= v(T)+(23-22)=2.\end{aligned}$$

That is when the agents 1,2 and 3 reorder themselves from σ_0 to σ_T, agent 5 profits from an earlier completion time. Actually, σ_T is also an optimal processing order for Q. Hence $v(Q)=2$. The complete family sequencing game (N,v) is given by: $v(N)=v(\{2,3,4,5\})=4$, $v(\{1,2,3,4\})=v(\{1,2,3,5\})=v(\{2,3,4\})=2$, $v(\{1,2,3\})=1$ and $v(T')=0$ for every remaining coalition $T'\in 2^N$.

Observe that this game is not convex:

$$v(N)-v(\{2,3,4,5\})=0<1=v(\{1,2,3\})-v(\{2,3\}).$$

(N,v) is not σ_0-component additive either: $\{1,2,3\}$ and $\{5\}$ are the σ_0-components of $\{1,2,3,5\}$, but $v(\{1,2,3,5\})=2\neq 1=v(\{1,2,3\})+v(\{5\})$. $\diamond$

Example 2.2.1 illustrated that family sequencing games need not be convex in general. The following example shows that a family sequencing game need not be convex even when the set-up times and the processing times are equal and the jobs of the same family are consequtive in the initial order.

Example 2.2.2 Let $\Sigma(N) = (N, F, \mathcal{F}, \sigma_0, s, p, \alpha)$ be a family sequencing situation with $N = \{1, 2, ..., 9\}$ and $F = \{1, 2, 3, 4\}$. Assume that $\mathcal{F}(1) = 1$, $\mathcal{F}(2) = \mathcal{F}(3) = 2$, $\mathcal{F}(4) = ... = \mathcal{F}(8) = 3$ and $\mathcal{F}(9) = 4$. Assume also that σ_0 is given by $\sigma_0(i) = i$ for every $i \in N$, $s = (1, 1, 1, 1)$, $p = (1, 1, 1, 1)$ and $\alpha = (100, 2, 4, 1)$. Finally, let (N, v) be the family sequencing game corresponding to $\Sigma(N)$.

Let $Q = \{1, 3, 4, 5, 6, 7, 8\}$, $R = Q \cup \{9\}$. It can be observed that $v(Q) = 0$, $v(R) = 5$ and $v(Q \cup \{2\}) = v(R \cup \{2\})$. Then, (N, v) is not convex since

$$v(Q \cup \{2\}) - v(Q) > v(R \cup \{2\}) - v(R).$$

$\diamond$

As illustrated in Example 2.2.1, family sequencing games are in general not σ_0-component additive in spite of the fact that a coalition can only change the positions of its members who belong to the same σ_0-component of the coalition. The reason that we lose σ_0-component additivity although we use a similar admissibility requirement to Curiel et al. (1989) is that the members of a coalition can affect the completion times of the members (and also the non-members) behind them by changing the total set-up time required to process the jobs. This property of family sequencing situations is in line with sequencing situations with controllable processing times (cf. van Velzen, 2006) where the players can affect the completion times of the players behind them by employing additional resources to reduce the time required to process their jobs.

In the following we will prove balancedness of family sequencing games by showing that the marginal corresponding to the initial order belongs to the core of these games.

Let $T \subset N$ and $\sigma \in \Pi(N)$. We will denote the member of T which stands in front of the other members of T in the order σ by $f(\sigma, T)$ and the member which stands behind the other members of T by $l(\sigma, T)$, i.e.,

$$f(\sigma, T) = \arg\min_{i \in T} \sigma(i) \quad \text{and} \quad l(\sigma, T) = \arg\max_{i \in T} \sigma(i).$$

We will call $f(\sigma, T)$ ($l(\sigma, T)$) the first player of T (the last player of T) with respect to σ. We will denote by $P(\sigma, T)$ ($S(\sigma, T)$) the set of players which stand in front of (behind) every member of T in the order σ, i.e.,

$$P(\sigma, T) = \{i \in N | \sigma(i) < \min_{j \in T} \sigma(j)\} \quad \text{and} \quad S(\sigma, T) = \{i \in N | \sigma(i) > \max_{j \in T} \sigma(j)\}.$$

Let $\Sigma(N) = (N, F, \mathcal{F}, \sigma_0, s, p, \alpha)$ be a family sequencing situation. For every $T \subset N$ and $\sigma \in \Pi(N)$, we denote by $g(\sigma, T)$ the cost savings obtained by T when the production order is changed from σ_0 to σ, i.e.,

$$g(\sigma, T) = \sum_{i \in T} \alpha_{\mathcal{F}(i)} \left(C(\sigma_0, i) - C(\sigma, i) \right).$$

Let $\sigma \in \Pi(N)$. We call a set of jobs R that are processed between two set-ups when the jobs are processed with respect to σ, a *run of* σ. Naturally, all jobs in the same run are of the same family. A run which consists of jobs of family k is called a run of family k. Given a connected coalition $T \in con(\sigma)$, a run R of σ is said to start in T (end in T) if $f(\sigma, R) \in T$ (if $l(\sigma, R) \in T$). Run R is said to start in front of T (end behind T) if $f(\sigma, R) \in P(\sigma, T)$ ($l(\sigma, R) \in S(\sigma, T)$). We call a run R of σ which starts in T the last run that starts in T with respect to σ if it includes $l(\sigma, T)$. With $\sigma' \in \Pi(N)$, σ' is said to split a run R of σ if there does not exist a run R' of σ' with $R \subset R'$.

Since a σ_0-component of a coalition can affect the completion times of the members of another σ_0-component behind it, it is generally not easy to find an optimal admissible processing order for coalitions. Nevertheless, there are useful properties regarding the structure of the optimal admissible orders. In the following proposition we show that, in every family sequencing situation, there exist optimal admissible orders for coalitions that process the jobs of the same family consecutively like the optimal order for the grand coalition.

Proposition 2.2.1 *Let $(N, F, \mathcal{F}, \sigma_0, s, p, \alpha)$ be a family sequencing situation and let $T \in 2^N \backslash \{\emptyset\}$. Then, there exists an optimal admissible order for T which processes all jobs of the same family within a σ_0-component of T consecutively.*

Proof. Let $T \backslash \sigma_0 = \{T_1, T_2, ..., T_l\}$ be such that $T_y \subset P(\sigma_0, T_{y+1})$ for every $y \in \{1, ..., l-1\}$. Let $\sigma \in \mathcal{A}(T)$ be optimal for T.

Assume that with respect to σ, family k jobs in T_y ($y \in \{1, 2, ..., l\}$) are processed in different runs. Let K_1 (K_2) be the set of family k jobs in T_y that belong to the first run (second run). Let M be the set of jobs (of other families) that are placed in between K_1 and K_2 with respect to σ. Also let $\tau = \sum_{i \in M} (x_{\sigma,i} s_{\mathcal{F}(i)} + p_{\mathcal{F}(i)})$. That is, τ is the time to process and set-up all jobs in M when they are processed with respect to σ. Let $i_1 = f(\sigma, K_1)$, $i_2 = f(\sigma, K_2)$ and $m = f(\sigma, M)$.

Now consider the processing order $\sigma' \in \Pi(N)$ which is obtained from σ by moving all jobs in K_1 to the head of K_2. Figure 2.6 depicts the orders σ and σ'. We will show that σ' is an optimal admissible order for T. Notice that we are done when we show that σ' is an optimal admissible order for T, since this proves that we can join two job groups of the same family that belong to the same σ_0-component of T but are processed in separate runs without decreasing the the total cost savings achieved by T.

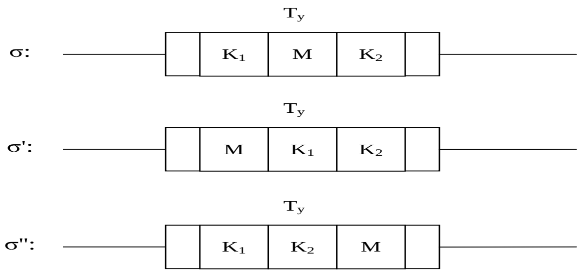

Figure 2.6: The orders σ, σ' and σ'' in the proof of Proposition 2.2.1

Let us first show that σ' is an admissible order for T. Clearly, $P(\sigma', j) = P(\sigma, j) = P(\sigma_0, j)$ for all $j \in N \backslash T$. So, we need to show that $C(\sigma', i) \leq C(\sigma_0, i)$ for every $i \in N \backslash T$. Since σ is an admissible order, it is sufficient to show that $C(\sigma', i) \leq C(\sigma, i)$ for every $i \in N \backslash T$. Observe that $x_{\sigma',j} = x_{\sigma,j}$ for every $j \in N \backslash \{i_1, i_2, m\}$. Hence, $C(\sigma', i) = C(\sigma, i)$ for every $i \in P(\sigma, K_1)$ and

$$\begin{aligned} C(\sigma, i) - C(\sigma', i) &= \sum_{j \in \bar{P}(\sigma,i)} \left(s_{\mathcal{F}(j)} x_{\sigma,j} + p_{\mathcal{F}(j)}\right) - \sum_{j \in \bar{P}(\sigma',i)} \left(s_{\mathcal{F}(i)} x_{\sigma',i} + p_{\mathcal{F}(i)}\right), \\ &= \sum_{j \in \{i_1, i_2, m\}} (x_{\sigma,j} - x_{\sigma',j}) s_{\mathcal{F}(j)}, \end{aligned}$$

for every $i \in S(\sigma, M)$.

Obviously, $x_{\sigma,m} = x_{\sigma,i_2} = 1$, $x_{\sigma',i_1} = 1$, $x_{\sigma',i_2} = 0$. Observe that x_{σ,i_1} and $x_{\sigma',m}$ can either be 0 or 1. Assume first that $x_{\sigma,i_1} = 0$, i.e., $l(\sigma, P(\sigma, K_1))$ is of family k. Then, clearly, $x_{\sigma',m} = 1$ and therefore

$$C(\sigma, i) - C(\sigma', i) = (0-1)s_{\mathcal{F}(i_1)} + (1-1)s_{\mathcal{F}(m)} + (1-0)s_{\mathcal{F}(i_2)} = 0,$$

for every $i \in S(\sigma, M)$.

Assume now that $x_{\sigma,i_1} = 1$, i.e., $l(\sigma, P(\sigma, K_1))$ is of a different family than k. Now, if $l(\sigma, P(\sigma, K_1))$ is in the same family with m then $x_{\sigma',m} = 0$. But, if $l(\sigma, P(\sigma, K_1))$ belongs to a different family than the one m belongs to, then $x_{\sigma',m} = 1$. Therefore,

$$C(\sigma, i) - C(\sigma', i) \geq (1-1)s_{\mathcal{F}(i_1)} + (1-1)s_{\mathcal{F}(m)} + (1-0)s_{\mathcal{F}(i_2)} = s_k,$$

for every $i \in S(\sigma, M)$. Hence, $\sigma' \in \mathcal{A}(T)$.

Now consider the cost savings $g(\sigma', T)$ obtained by T by switching from σ_0 to σ' and suppose that $g(\sigma', T)$ is strictly less than $v(T)$. We will show that this contradicts with the optimality of σ. Observe that

$$\begin{aligned}
0 > g(\sigma', T) - v(T) &= \sum_{i \in T} \alpha_{\mathcal{F}(i)} \left(C(\sigma_0, i) - C(\sigma', i)\right) - \sum_{i \in T} \alpha_{\mathcal{F}(i)} \left(C(\sigma_0, i) - C(\sigma, i)\right) \\
&= \sum_{i \in T} \alpha_{\mathcal{F}(i)} \left(C(\sigma, i) - C(\sigma', i)\right) \geq \sum_{i \in K_1 \cup M} \alpha_{\mathcal{F}(i)} \left(C(\sigma, i) - C(\sigma', i)\right),
\end{aligned} \tag{2.9}$$

where the last inequality follows from the fact that $C(\sigma', i) = C(\sigma, i)$ for every $i \in P(\sigma, K_1)$ and $C(\sigma, i) - C(\sigma', i) \geq 0$ for every $i \in S(\sigma, M)$.

It can also be observed that

$$\begin{aligned}
C(\sigma, i) - C(\sigma', i) &= \sum_{j \in K_1} (x_{\sigma,j} s_{\mathcal{F}(j)} + p_{\mathcal{F}(j)}) + (x_{\sigma,m} - x_{\sigma',m}) s_{\mathcal{F}(m)} \\
&= |K_1| p_k + x_{\sigma,i_1} s_k + (x_{\sigma,m} - x_{\sigma',m}) s_{\mathcal{F}(m)} \\
&= |K_1| p_k + x_{\sigma,i_1} s_k + (1 - x_{\sigma',m}) s_{\mathcal{F}(m)} \geq |K_1| p_k,
\end{aligned} \tag{2.10}$$

for every $i \in M$ and

$$C(\sigma, i) - C(\sigma', i) = -\sum_{j \in M} (x_{\sigma',j} s_{\mathcal{F}(j)} + p_{\mathcal{F}(j)}) - (x_{\sigma',i_1} - x_{\sigma,i_1}) s_k \geq -(\tau + s_k), \tag{2.11}$$

for every $i \in K_1$ since

$$\sum_{j \in M} (x_{\sigma',j} s_{\mathcal{F}(j)} + p_{\mathcal{F}(j)}) = \begin{cases} \tau, & \text{if } x_{\sigma',m} = 1, \\ \tau - s_{\mathcal{F}(m)}, & \text{if } x_{\sigma',m} = 0. \end{cases}$$

Then, by inequalities (2.9)-(2.11), we have that

$$0 > \sum_{i \in K_1 \cup M} \alpha_{\mathcal{F}(i)} \left(C(\sigma, i) - C(\sigma', i)\right) \geq |K_1| \left(p_k \sum_{i \in M} \alpha_{\mathcal{F}(i)} - (\tau + s_k) \alpha_k \right).$$

Hence,

$$p_k \sum_{i \in M} \alpha_{\mathcal{F}(i)} - (\tau + s_k) \alpha_k < 0. \tag{2.12}$$

Next consider the processing order $\sigma'' \in \Pi(N)$ which is obtained from σ by moving all jobs in K_2 to the tail of K_1. Figure 2.6 also depicts σ''. Let us first show that σ'' is an admissible order for T. Obviously, $P(\sigma'', i) = P(\sigma_0, i)$ for every $i \in N \backslash T$. Since σ is an admissible order, it is sufficient to show that $C(\sigma'', i) \leq C(\sigma, i)$ for every $i \in N \backslash T$. Let $h = f(\sigma, S(\sigma, K_2))$. It can be observed that $x_{\sigma'',i} = x_{\sigma,i}$ for every $i \in N \backslash \{i_2, h\}$. Hence, $C(\sigma'', i) = C(\sigma, i)$ for every $i \in P(\sigma, M)$ and

$$\begin{aligned}
C(\sigma, i) - C(\sigma'', i) &= \sum_{j \in \bar{P}(\sigma, i)} \left(s_{\mathcal{F}(j)} x_{\sigma,j} + p_{\mathcal{F}(j)}\right) - \sum_{j \in \bar{P}(\sigma'', i)} \left(s_{\mathcal{F}(i)} x_{\sigma'',i} + p_{\mathcal{F}(i)}\right), \\
&= \sum_{j \in \{i_2, h\}} (x_{\sigma,j} - x_{\sigma'',j}) s_{\mathcal{F}(j)},
\end{aligned}$$

for every $i \in S(\sigma, K_2)$.

Clearly, $x_{\sigma,i_2} = 1$ and $x_{\sigma'',i_2} = 0$. However, $x_{\sigma,h}$ can either be 0 or 1. Assume first that $x_{\sigma,h} = 0$, i.e., h is of family k. Then, it can be observed that $x_{\sigma'',h} = 1$ and hence,

$$\sum_{i\in\{i_2,h\}} (x_{\sigma,i} - x_{\sigma'',i})s_{\mathcal{F}(i)} = (1-0)s_{\mathcal{F}(i_2)} + (0-1)s_{\mathcal{F}(h)} = (1-0)s_k + (0-1)s_k = 0.$$

Assume now that $x_{\sigma,h} = 1$. Then,

$$\sum_{i\in\{i_2,h\}} (x_{\sigma,i} - x_{\sigma'',i})s_{\mathcal{F}(i)} \geq (1-0)s_k + (1-1)s_{\mathcal{F}(h)} = s_k$$

and we can conclude that $\sigma'' \in \mathcal{A}(T)$.

Observe now that

$$C(\sigma,i) - C(\sigma'',i) = \begin{cases} \tau + s_k, & \text{if } i \in K_2, \\ -|K_2|p_k, & \text{if } i \in M. \end{cases}$$

Then,

$$\begin{aligned} g(\sigma'',T) - v(T) &= \sum_{i\in T} \alpha_{\mathcal{F}(i)} \left(C(\sigma_0,i) - C(\sigma'',i)\right) - \sum_{i\in T} \alpha_{\mathcal{F}(i)} \left(C(\sigma_0,i) - C(\sigma,i)\right) \\ &= \sum_{i\in T} \alpha_{\mathcal{F}(i)} \left(C(\sigma,i) - C(\sigma'',i)\right) \geq \sum_{i\in K_2\cup M} \alpha_{\mathcal{F}(i)} \left(C(\sigma,i) - C(\sigma',i)\right) \\ &= |K_2| \left((\tau + s_k)\alpha_k - p_k \sum_{i\in M} \alpha_{\mathcal{F}(i)} \right) > 0, \end{aligned} \tag{2.13}$$

where the first inequality follows from the fact that $C(\sigma'',i) = C(\sigma,i)$ for every $i \in P(\sigma,M)$ and $C(\sigma,i) - C(\sigma'',i) \geq 0$ for every $i \in S(\sigma,K_2)$ while the last inequality is implied by inequality (2.12). However, this establishes a contradiction with the optimality of σ. □

Next we focus on the structure of the optimal admissible orders for specific coalitions that are connected with respect to the initial order.

Let $\Sigma(N) = (N, F, \mathcal{F}, \sigma_0, s, p, \alpha)$ be a family sequencing situation. For every $T \subset con(\sigma_0)$ and $k \in F$, the family urgency index $u_{T,k}$ for T is defined as

$$u_{T,k} = \frac{n_{T,k}\alpha_k}{s_k + n_{T,k}p_k},$$

where $n_{T,k}$ is the number of family k jobs in T.

Assume now that $\mathcal{F}(l(\sigma_0,T)) = \bar{k}$. The tail-adjusted family urgency index $u'_{T,k}$ for T is defined as

$$u'_{T,k} = \begin{cases} u_{T,k}, & \text{if } k \in F\backslash\{\bar{k}\}, \\ \min\limits_{k\in F} \dfrac{u_{T,k}}{2}, & \text{if } k = \bar{k}. \end{cases}$$

Then an order $\sigma \in \Pi(N)$ is called *an HUCF order for T* if

(i) $P(\sigma, i) = P(\sigma_0, i)$ for every $i \in N \backslash T$ and

(ii) it processes all jobs of T that belong to the same family consecutively and processes these family groups in non-increasing order of the family urgency index $u_{T,k}$.

Then an order $\sigma \in \Pi(N)$ is called *a tail-adjusted HUCF order for T* if

(i) $P(\sigma, i) = P(\sigma_0, i)$ for every $i \in N \backslash T$ and

(ii) it processes all jobs of T that belong to the same family consecutively and processes these family groups in non-increasing order of the tail-adjusted family urgency index $u'_{T,k}$.

Notice that a tail-adjusted HUCF order for T can be obtained from an HUCF order for T by taking the family $\bar{k}$ jobs behind all other family groups.

In the following proposition, we show that for coalitions that are connected with respect to σ_0 and contain the first player in the order σ_0, either an HUCF order or a tail-adjusted HUCF order is optimal.

Proposition 2.2.2 *Let $\Sigma(N) = (N, F, \mathcal{F}, \sigma_0, s, p, \alpha)$ be a family sequencing situation and $T \in con(\sigma_0)$ with $\sigma_0^{-1}(1) \in T$. Then,*

(i) If an HUCF order for T is admissible, then it is optimal for T.

(ii) If an HUCF order for T is not admissible, then a tail-adjusted HUCF order for T is optimal for T.

It can easily be observed that (i) immediately follows from Theorem 2.2.1 and Proposition 2.2.1. If an HUCF order σ is not admissible for T, then, by switching from σ_0 to σ, T must increase the total set-up time required to process the jobs. In other words, σ must split some runs of σ_0. It can be observed that the only run that can be splitted by switching from σ_0 to σ is the last run that starts in T with respect to σ_0. Moreover, since this run is splitted, it must end behind T with respect to σ_0. Then, a tail-adjusted HUCF order for T is an admissible order for T, since it does not split any runs of σ_0. The optimality of the tail-adjusted HUCF order is also implied by Theorem 2.2.1 and Proposition 2.2.1.

The following example illustrates the optimality of HUCF orders for coalitions that are connected with respect to σ_0 and contain the first player in the order σ_0.

Example 2.2.3 Consider the family sequencing situation $(N, F, \mathcal{F}, \sigma_0, s, p, \alpha)$ with $N = \{1,2,3,4,5\}$ and $F = \{1,2\}$. Assume that $\mathcal{F}(1) = \mathcal{F}(4) = \mathcal{F}(5) = 1$ and $\mathcal{F}(2) = \mathcal{F}(3) = 2$. Assume also that σ_0 is given by $\sigma_0(i) = i$ for every $i \in N$, $s = (1,5)$, $p = (4,2)$ and $\alpha = (1,2)$. Consider first the coalition $T = \{1,2,3\}$. The family urgency indices for this coalition can be calculated as $u_{T,1} = \frac{1}{5}$ and $u_{T,2} = \frac{4}{9}$. Then, an HUCF order for T is given by $\sigma_T = (2,3,1,4,5)$. σ_T is an admissible order for T with respect to σ_0 since $P(\sigma_T, i) = P(\sigma_0, i)$ for every $i \in N \backslash T$ and both $C(\sigma_T, 4) < C(\sigma_0, 4)$ and $C(\sigma_T, 5) < C(\sigma_0, 5)$. Then, by Proposition 2.2.2, σ_T is optimal for T.

Consider now the coalition $T' = \{1,2\}$. The family urgency indices for T' are $u_{T',1} = \frac{1}{5}$ and $u_{T',2} = \frac{2}{7}$. Then, $\sigma_{T'} = (2,1,3,4,5)$ is the HUCF order for T'. However, $\sigma_{T'}$ is not admissible for T': It splits the run of σ_0 that consists of jobs 2 and 3 and as a result $C(\sigma_{T'}, i) = C(\sigma_0, i) + 5$ for every $i \in \{3,4,5\}$. Observe that the initial order $\sigma_0 = (1,2,3,4,5)$ is the tail-adjusted HUCF order for T' and by Proposition 2.2.2, it is optimal for T'. $\diamond$

Now we are ready to prove that the marginal vector of a family sequencing game that corresponds to the initial order belongs to the core of the game.

Theorem 2.2.2 *Let $\Sigma(N) = (N, F, \mathcal{F}, \sigma_0, s, p, \alpha)$ be a family sequencing situation and (N, v) be the corresponding sequencing game. Then, $m^{\sigma_0}(v) \in Core(v)$.*

Proof. Let $T \backslash \sigma_0 = \{T_1, T_2, ..., T_l\}$ be such that $T_y \subset P(\sigma_0, T_{y+1})$ for every $y \in \{1, ..., l-1\}$. Let $\sigma \in \mathcal{A}(T)$ be optimal for T.

We want to show that $\sum_{i \in T} m_i^{\sigma_0}(v) \geq v(T)$. Since

$$v(T) = \sum_{i \in T} \alpha_{\mathcal{F}(i)} \left(C(\sigma_0, i) - C(\sigma, i) \right) = \sum_{y \in \{1,2,...,l\}} g(\sigma, T_y)$$

and

$$\sum_{i \in T} m_i^{\sigma_0}(v) = \sum_{y \in \{1,2,...,l\}} \sum_{i \in T_y} m_i^{\sigma_0}(v),$$

it is sufficient to show that $\sum_{i \in T_y} m_i^{\sigma_0}(v) \geq g(\sigma, T_y)$ for every $y \in \{1,2,...,l\}$.

Pick $y \in \{1,2,...,l\}$. Let $D = P(\sigma_0, T_y)$ and $E = D \cup T_y$. Notice that $\sigma_0^{-1}(1) \in D$ and $\sum_{i \in T_y} m_i^{\sigma_0}(v) = v(E) - v(D)$. It remains to show that

$$v(E) - v(D) - g(\sigma, T_y) \geq 0.$$

We denote by $\sigma_{|T_y}$ and by σ' the orders defined by

$$\sigma_{|T_y}(i) = \begin{cases} \sigma(i), & \text{if } i \in T_y, \\ \sigma_0(i), & \text{otherwise,} \end{cases}$$

$$\sigma'(i) = \begin{cases} \sigma(i), & \text{if } i \in \bigcup_{q=1}^{y} T_q, \\ \sigma_0(i), & \text{otherwise.} \end{cases}$$

Notice that σ' is an admissible order for E.

Let π be an optimal admissible order for D. Since $\sigma_0^{-1}(1) \in D$, we can assume that π is either an HUCF order for D or a tail-adjusted HUCF order for D. Let $\mu \in \Pi(N)$ be the order defined by

$$\mu(i) = \begin{cases} \pi(i), & \text{if } i \in D, \\ \sigma_{|T_y}(i), & \text{if } i \in T_y, \\ \sigma_0(i), & \text{otherwise.} \end{cases}$$

Observe that $P(\sigma_0, i) = P(\mu, i)$ for every $i \in N \backslash E$.

It can be observed that

$$\begin{aligned} g(\mu, E) - v(D) - g(\sigma, T_y) &= \sum_{i \in E} \alpha_{\mathcal{F}(i)} \left(C(\sigma_0, i) - C(\mu, i) \right) - \sum_{i \in D} \alpha_{\mathcal{F}(i)} \left(C(\sigma_0, i) - C(\pi, i) \right) \\ &\quad - \sum_{i \in T_y} \alpha_{\mathcal{F}(i)} \left(C(\sigma_0, i) - C(\sigma, i) \right) \\ &= \sum_{i \in T_y} \alpha_{\mathcal{F}(i)} \left(C(\sigma', i) - C(\mu, i) \right), \end{aligned} \tag{2.14}$$

where the last equality follows from the fact that $C(\sigma, i) = C(\sigma', i)$ for every $i \in T_y$ and $C(\pi, i) = C(\mu, i)$ for every $i \in D$.

Let i_y be the first player of T_y with respect to $\sigma_{|T_y}$. Assume that i_y is of family k. Observe that for every $i \in S(\sigma_0, D)$

$$\begin{aligned} C(\sigma', i) - C(\mu, i) &= \sum_{j \in \bar{P}(\sigma', i)} \left(s_{\mathcal{F}(j)} x_{\sigma', j} + p_{\mathcal{F}(j)} \right) - \sum_{j \in \bar{P}(\mu, i)} \left(s_{\mathcal{F}(j)} x_{\mu, j} + p_{\mathcal{F}(j)} \right), \\ &= \sum_{j \in D} s_{\mathcal{F}(j)} (x_{\sigma', j} - x_{\mu, j}) + s_k (x_{\sigma', i_y} - x_{\mu, i_y}) \\ &\geq s_k (x_{\sigma', i_y} - x_{\mu, i_y}), \end{aligned} \tag{2.15}$$

where the second equality follows from the fact that $\bar{P}(\sigma', i) = \bar{P}(\mu, i)$ for every $i \in S(\sigma_0, D)$ and $x_{\sigma', i} = x_{\mu, i}$ for every $i \in S(\sigma_0, D) \backslash \{i_y\}$ and the inequality follows from the fact that, with respect to μ, the members of D are processed with respect to π which is an HUCF or a tail-adjusted HUCF order for D and these orders require the minimum total set-up time to process the jobs in D.

Then by equation (2.14) and inequality (2.15),

$$g(\mu, E) - v(D) - g(\sigma, T_y) \geq s_k (x_{\sigma', i_y} - x_{\mu, i_y}) \sum_{i \in T_y} \alpha_{\mathcal{F}(i)}. \tag{2.16}$$

If $x_{\sigma',i_y} - x_{\mu,i_y} \geq 0$, then by inequality (2.15) and admissibility of σ', $C(\sigma_0, i) \geq C(\sigma', i) \geq C(\mu, i)$ for every $i \in S(\sigma_0, D)$. Hence, μ is an admissible order for E. Moreover, by inequality (2.16),

$$v(E) - v(D) - g(\sigma, T_y) \geq g(\mu, E) - v(D) - g(\sigma, T_y) \geq 0.$$

So, we can assume in the following that $x_{\sigma',i_y} = 0$ and $x_{\mu,i_y} = 1$.

First, by inequalities (2.15) and (2.16)

$$C(\sigma', i) - C(\mu, i) \geq -s_k, \qquad \text{for every } i \in S(\sigma_0, D) \qquad \text{and} \tag{2.17}$$

$$g(\mu, E) - v(D) - g(\sigma, T_y) \geq -s_k \sum_{i \in T_y} \alpha_{\mathcal{F}(i)}. \tag{2.18}$$

Since $x_{\sigma',i_y} = 0$, $\mathcal{F}(i_y) = \mathcal{F}(l(\sigma', D)) = k$. Observe that $l(\sigma', D) = l(\sigma_0, D)$. Moreover, since $x_{\mu,i_y} = 1$, $\mathcal{F}(l(\mu, D)) = \mathcal{F}(l(\pi, D)) \neq k$. Then π can not be a tail-adjusted HUCF order for D because the last player of D with respect to a tail-adjusted HUCF order for D must be of family k, the family of $l(\sigma_0, D)$. So, π is an HUCF order for D which splits the last run of D with respect to $\sigma_{|T_y}$.

Let K_1 (K_2) be the set of family k jobs in D (T_y) that belong to the last run of D with respect to $\sigma_{|T_y}$. Since π splits the last run of D with respect to $\sigma_{|T_y}$, K_1 and K_2 are processed in two different runs of μ. Let R_1 (R_2) be the run of μ that K_1 (K_2) belongs to. Also let M be the set of jobs (of other families) that are placed in between R_1 and K_2 with respect to μ. Figure 2.7 depicts the order μ. Let τ be the time to process and set-up all jobs in M when they are processed with respect to μ, i.e., $\tau = \sum_{i \in M}(x_{\mu,i} s_{\mathcal{F}(i)} + p_{\mathcal{F}(i)})$.

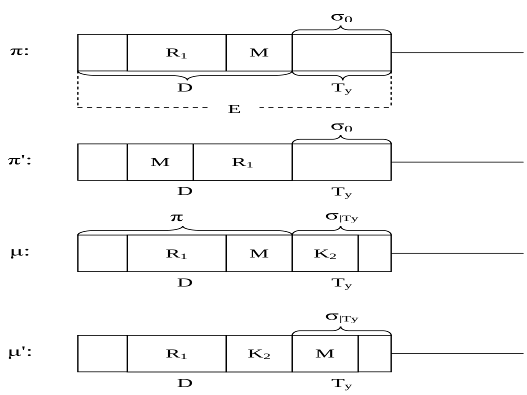

Figure 2.7: The orders π, π', μ and μ' in the proof of Theorem 2.2.2

Claim 1: $\quad \tau\alpha_k - p_k \sum_{i \in M} \alpha_{\mathcal{F}(i)} \geq 0.$

Proof of Claim 1. Consider π' the tail-adjusted HUCF order for D obtained from π by taking R_1 the group of family k jobs in D behind M. Figure 2.7 also depicts orders π and π'. Since π' is a tail-adjusted HUCF order for D, it is admissible for D.

Observe that

$$C(\pi', i) - C(\pi, i) = \begin{cases} 0, & \text{if } i \in D \backslash (M \cup R_1), \\ \tau, & \text{if } i \in R_1, \\ -(s_k + |R_1| p_k), & \text{if } i \in M. \end{cases}$$

Therefore,

$$\begin{aligned} v(D) - g(\pi', D) &= \sum_{i \in D} \alpha_{\mathcal{F}(i)} \big(C(\pi', i) - C(\pi, i)\big) = \sum_{i \in R_1 \cup M} \alpha_{\mathcal{F}(i)} \big(C(\pi', i) - C(\pi, i)\big) \\ &= |R_1| \tau \alpha_k - (s_k + |R_1| p_k) \sum_{i \in M} \alpha_{\mathcal{F}(i)} \\ &= |R_1| \left(\tau \alpha_k - p_k \sum_{i \in M} \alpha_{\mathcal{F}(i)} \right) - s_k \sum_{i \in M} \alpha_{\mathcal{F}(i)} \geq 0, \end{aligned}$$

where the last inequality follows from the fact that π is an optimal admissible order for D and π' is admissible for D. But then

$$\tau \alpha_k - p_k \sum_{i \in M} \alpha_{\mathcal{F}(i)} \geq \frac{s_k \sum_{i \in M} \alpha_{\mathcal{F}(i)}}{|R_1|} \geq 0.$$

Now consider the order μ' which is obtained from μ by moving all jobs in K_2 to the tail of R_1. Figure 2.7 also depicts the order μ'.

Claim 2: μ' is an admissible order for E and

$$g(\mu', E) - g(\mu, E) \geq s_k \sum_{i \in T_y} \alpha_{\mathcal{F}(i)}.$$

Observe that if Claim 2 is true, then we are done:

$$\begin{aligned} v(E) - v(D) - g(\sigma, T_y) &\geq g(\mu', E) - v(D) - g(\sigma, T_y) \\ &\geq s_k \sum_{i \in T_y} \alpha_{\mathcal{F}(i)} + g(\mu, E) - v(D) - g(\sigma, T_y) \geq 0, \end{aligned}$$

where the first inequality follows from the admissibility of μ' and the last inequality follows from inequality (2.18) and Claim 2.

Proof of Claim 2. Let j_y be the job which is preceded by $l(\mu, K_2)$ in the order μ. Observe that $x_{\mu',i} = x_{\mu,i}$ for every $i \in N \backslash \{i_y, j_y\}$. Hence, $C(\mu, i) = C(\mu', i)$ for every $i \in P(\mu, M)$ and

$$\begin{aligned} C(\mu,i) - C(\mu',i) &= \sum_{j\in\bar{P}(\mu,i)} (s_{\mathcal{F}(j)}x_{\mu,j} + p_{\mathcal{F}(j)}) - \sum_{j\in\bar{P}(\mu',i)} (s_{\mathcal{F}(j)}x_{\mu',j} + p_{\mathcal{F}(j)}), \\ &= \sum_{j\in\{i_y,j_y\}} (x_{\mu,j} - x_{\mu',j})s_{\mathcal{F}(j)}, \end{aligned} \tag{2.19}$$

for every $i \in S(\mu, K_2)$.

Obviously, $x_{\mu,i_y} = 1$, $x_{\mu',i_y} = 0$. Observe that x_{μ,j_y} can either be 0 or 1.

Case 1: $x_{\mu,j_y} = 1$. Then,

$$C(\mu,i) - C(\mu',i) \geq (1-0)s_{\mathcal{F}(i_y)} + (1-1)s_{\mathcal{F}(j_y)} = s_k,$$

for every $i \in S(\mu, K_2)$. Moreover, we know by inequality (2.17) that $C(\sigma', i) - C(\mu, i) \geq -s_k$ for every $i \in S(\mu', E)$ and σ' is an admissible order for E. Hence, μ' is also an admissible order for E since

$$C(\sigma_0,i) - C(\mu',i) \geq C(\sigma',i) - C(\mu',i) \geq C(\sigma',i) - C(\mu,i) + s_k \geq 0,$$

for every $i \in S(\sigma_0, E)$.

Observe also that

$$C(\mu,i) - C(\mu',i) = \begin{cases} \tau + s_k, & \text{if } i \in K_2, \\ |K_2|p_k, & \text{if } i \in M. \end{cases} \tag{2.20}$$

Therefore,

$$\begin{aligned} g(\mu',E) - g(\mu,E) &= \sum_{i\in E} \alpha_{\mathcal{F}(i)} \left(C(\mu,i) - C(\mu',i)\right) = \sum_{i\in M\cup T_y} \alpha_{\mathcal{F}(i)} \left(C(\mu,i) - C(\mu',i)\right) \\ &= \sum_{i\in K_2\cup M} \alpha_{\mathcal{F}(i)} \left(C(\mu,i) - C(\mu',i)\right) + \sum_{i\in T_y\backslash K_2} \alpha_{\mathcal{F}(i)} \left(C(\mu,i) - C(\mu',i)\right) \\ &\geq |K_2| \left((\tau + s_k)\alpha_k - p_k \sum_{i\in M} \alpha_{\mathcal{F}(i)}\right) + s_k \sum_{i\in T_y\backslash K_2} \alpha_{\mathcal{F}(i)} \\ &\geq |K_2|s_k\alpha_k + s_k \sum_{i\in T_y\backslash K_2} \alpha_{\mathcal{F}(i)} = s_k \sum_{i\in T_y} \alpha_{\mathcal{F}(i)}, \end{aligned} \tag{2.21}$$

where the second equality follows from the fact that $C(\mu, i) = C(\mu', i)$ for every $i \in P(\mu, M)$; the last but one inequality follows from equation (2.20) and the fact that $C(\mu, i) - C(\mu', i)$ is at least s_k for every $i \in T_y \backslash K_2$ and the last inequality follows by Claim 1.

Case 2: $x_{\mu,j_y} = 0$.

Observe first that $x_{\mu,j_y} = 0$ only when R_2 ends behind T_y. We know that the first job of R_2 with respect to μ is i_y, the first job of T_y with respect to $\sigma_{|T_y}$. Since R_2 ends behind T_y with respect to μ, every job in T_y must belong to R_2, i.e., each job in T_y is of family k and $K_2 = T_y$. Moreover, then j_y is the first player of $S(\sigma_0, T_y)$.

Observe also that since each job in T_y is of family k, $C(\pi, i) = C(\mu, i)$ for every $i \in S(\sigma_0, E)$ and hence $C(\mu, i) = C(\pi, i) \leq C(\sigma_0, i)$, i.e., μ is also an admissible order for E.

Consider now the order μ'. We know that $x_{\mu,i_y} = 1$, $x_{\mu',i_y} = 0$ and $x_{\mu,j_y} = 0$. Observe that $x_{\mu',j_y} = 1$ since the last job of M with respect to μ' belongs to a different family than k. Then by equation (2.19)

$$C(\mu, i) - C(\mu', i) = \sum_{j \in \{i_y, j_y\}} (x_{\mu,j} - x_{\mu',j}) s_{\mathcal{F}(j)} = 0, \tag{2.22}$$

for every $i \in S(\mu, E)$. And hence μ' is also an admissible order for E.

Moreover,

$$\begin{aligned} g(\mu', E) - g(\mu, E) &= \sum_{i \in M \cup T_y} \alpha_{\mathcal{F}(i)} \big(C(\mu, i) - C(\mu', i)\big) = |T_y| \left((\tau + s_k)\alpha_k - p_k \sum_{i \in M} \alpha_{\mathcal{F}(i)} \right) \\ &\geq s_k \sum_{i \in T_y} \alpha_{\mathcal{F}(i)}, \end{aligned} \tag{2.23}$$

where the equality follows by equation (2.20) and the inequality follows by Claim 1. □

As mentioned before, sequencing situations with controllable processing times (cf. van Velzen, 2006) are in the same spirit with family sequencing situations, because, in both types of sequencing situations, admissible reorderings of connected coalitions can affect the completion times of the players behind them. van Velzen (2006) analyzed the sequencing games corresponding to sequencing situations with controllable processing times and proved that many marginal vectors of these games are core elements. This result was obtained by showing that the sequencing games corresponding to sequencing situations with controllable processing times are *permutationally convex* with respect to many orders on the set of players. Permutational convexity (cf. Granot and Huberman, 1982) with respect to an order $\sigma \in \Pi(N)$ is a well-known sufficient condition for the marginal vector of a cooperative game that corresponds to σ to be a core element. Formally, a TU-game (N, v) is said to be *permutationally convex with respect to* $\sigma \in \Pi(N)$ if

$$v(\bar{P}(\sigma, i) \cup T) - v(\bar{P}(\sigma, i)) \leq v(\bar{P}(\sigma, j) \cup T) - v(\bar{P}(\sigma, j)),$$

for every $i, j \in N$ with $\sigma(i) < \sigma(j)$ and $T \subset S(\sigma, j)$.

In the following example, we show that family sequencing games need not be permutationally convex with respect to the initial order although the corresponding marginal is an element of the core.

Example 2.2.4 Consider the family sequencing situation $(N, F, \mathcal{F}, \sigma_0, s, p, \alpha)$ with $N = \{1,2,3,4,5,6\}$ and $F = \{1,2,3,4\}$. Assume that $\mathcal{F}(1) = \mathcal{F}(3) = 1$, $\mathcal{F}(2) = 2$, $\mathcal{F}(4) = \mathcal{F}(5) = 3$ and $\mathcal{F}(6) = 4$. Assume also that σ_0 is given by $\sigma_0(i) = i$ for every $i \in N$, $s = (2,2,1,5)$, $p = (1,2,2,5)$ and $\alpha = (10,10,10,1)$. Finally, let (N, v) be the family sequencing game corresponding to $\Sigma(N)$.

Consider the coalitions $Q = \{1,2,3\}$, $Q' = \{1,2,3,6\}$, $R = \{1,2,3,4\}$ and $R' = \{1,2,3,4,6\}$. The urgency indices for Q are $u_{Q,1} = \frac{20}{4}$ and $u_{Q,2} = \frac{10}{4}$. Hence, $\sigma_Q = (1,3,2,4,5,6)$ is an HUCF order for Q. It can easily be observed that σ_Q is admissible for Q. Then, by Proposition 2.2.2, σ_Q is optimal for Q.

The urgency indices for R are $u_{R,1} = \frac{20}{4}$, $u_{R,2} = \frac{10}{4}$ and $u_{R,3} = \frac{10}{3}$. Hence, $\sigma_R = (1,3,4,2,5,6)$ is an HUCF order for R. It can easily be observed that σ_R is admissible for R. Then, by Proposition 2.2.2, σ_R is optimal for R.

Observe also that σ_Q is an optimal admissible order for Q' and σ_R is an optimal admissible order for R'. Then, (N, v) is not permutationally convex with respect to σ_0 since

$$2 = v(Q') - v(Q) > v(R') - v(R) = 1.$$

◇

In several sequencing games considered in the literature, both the marginal vector corresponding to the initial order σ_0 and the marginal vector corresponding to the inverse order of σ_0 are core elements (e.g., Curiel et al., 1989; Çiftçi et al., 2008; Curiel et al., 1993). In the following example, we show that the marginal vector corresponding to the inverse order of σ_0 need not belong to the core of a family sequencing game.

Example 2.2.5 Consider the family sequencing situation $(N, F, \mathcal{F}, \sigma_0, s, p, \alpha)$ with $N = \{1,2,3,4\}$ and $F = \{1,2\}$. Assume that $\mathcal{F}(1) = \mathcal{F}(3) = 1$ and $\mathcal{F}(2) = \mathcal{F}(4) = 2$. Assume also that σ_0 is given by $\sigma_0(i) = i$ for every $i \in N$, $s = (2,1)$, $p = (\frac{1}{10}, 4)$ and $\alpha = (1,10)$. Finally, let (N, v) be the family sequencing game corresponding to $\Sigma(N)$.

The complete family sequencing game (N, v) is given by $v(N) = 62$, $v(\{1,2,3\}) = 18$, $v(\{1,2,4\}) = 36$, $v(\{1,3,4\}) = v(\{2,3,4\}) = v(\{3,4\}) = 27$, $v(\{1,2\}) = 16$ and $v(T) = 0$ for every remaining coalition $T \subset N$. Then, $m^{\sigma_0^{-1}}(v) \notin Core(v)$ since

$$35 = \sum_{i \in \{1,2,4\}} m_i^{\sigma_0^{-1}}(v) < v(\{1,2,4\}) = 36.$$

◇

Chapter 3

Connection Situations and Cooperation

Networks play an important role in modern economic life. Transportation networks (airlines, railroads, shipping services, postal services) provide us the means to commute and to deliver products. Communication networks (internet, telephone, broadcast television, radio) allow us to conduct the necessary transactions required for daily economic activities and energy networks (electric power transmission and distribution, water, natural gas and petroleum pipelines) help to provide the energy and resources required to maintain all of these activities. In this chapter, which is based on Çiftçi and Tijs (2009) and Çiftçi et al. (2007), we consider the problem of allocating the construction costs of networks in an interactive cooperative setting.

In Section 3.2, we focus on the problem of allocating the construction costs of networks arising in minimum cost spanning tree (mcst) problems. These problems consider a group of agents, each of whom has to be connected to a source, either directly or via other agents. One example would be a situation where villagers have to construct and pay pipelines from their respective houses to a water supplier. This type of cost allocation problems was first introduced in the economics literature by Claus and Kleitman (1973). The seminal paper by Bird (1976) provided the first game theoretical treatment of this problem by associating a coalitional game with transferable utility to mcst problems. Then, solution concepts of cooperative game theory were implemented and subsequently proposed as appropriate cost allocations for mcst problems by several studies: Granot and Huberman (1981, 1984) analyzed the core and the nucleolus; Kar (2002) studied the Shapley value.

Cost allocation rules for mcst problems can also be defined directly without considering the underlying cost game. In particular, one can make use of an algorithm to construct an mcst and allocate the cost of each edge constructed by the algorithm among the agents by following an appropriate method. Cost allocation rules which follow such a procedure are baptized construct and charge rules in Moretti et al. (2005). Construct and charge rules proposed in the literature mainly focus on the two well-known algorithms, Kruskal's algorithm (Kruskal, 1956) and Prim's algorithm (Prim, 1957), for constructing an mcst. In particular, the Bird rule (Bird, 1976) and the extended Bird rule (Dutta and Kar, 2004) rely on Prim's algorithm while the equal remaining obligations rule (ERO) (Feltkamp et al., 1994) and obligation rules (Tijs et al., 2006) rely on Kruskal's algorithm.

Recent contributions to the literature on cost allocation in mcst situations revealed the fact that ERO satisfies many appealing properties: Branzei et al. (2004) and Bergantiños and Vidal-Puga (2005a) obtained axiomatic characterizations of ERO which are based on properties such as additivity and equal treatment. Tijs et al. (2006) showed that obligation rules (and hence ERO) satisfy appealing population monotonicity and cost monotonicity properties. Norde et al. (2004) showed that an allocation scheme which is obtained by using ERO as an allocation vector is population monotonic. Moreover, Bergantiños and Vidal-Puga (2007a) showed that other rules in the literature fail to satisfy some properties that are satisfied by ERO. They also provided an axiomatic characterization of ERO based on monotonicity properties.

The original definition of ERO by Feltkamp et al. (1994) consists of a step-by-step procedure: Kruskal's algorithm is employed to construct an mcst and at each step of the algorithm the cost of the constructed edge is divided among agents who make use of the edge with respect to a prespecified scheme. Bergantiños and Vidal-Puga (2005b) provided two different approaches to obtain ERO. They showed that ERO can be obtained as the average of cost allocations provided by a procedure associated with the irreducible matrix of the mcst situation and also with a procedure which computes the part of the cost of edges in an mcst that every agent has to pay. Moreover, Bergantiños and Vidal-Puga (2005b, 2007a) showed that Shapley values of the so-called optimistic game associated with an mcst situation and of the coalitional game associated with the irreducible matrix are both equal to ERO.

In Section 3.2 we present a new approach to obtain ERO and hence provide yet another support for this important rule. For this aim, we consider a construct and charge procedure, which we call the vertex oriented construct and charge procedure (voccp). This procedure leads to an mcst and a cost sharing allocation where each agent pays the edge which she chose to construct in the procedure. Postponing a precise definition to the following sections, voccp can be explained as follows. Voccp works in steps. At each step of the procedure one agent constructs one edge and the agent who is going to construct an edge is determined by making use of an order on the set of agents. That is, at each step of the procedure, the first agent in the order who has not yet constructed an edge constructs and pays the cheapest edge which connects the component that the agent belongs to with another component. The main result of our study is that ERO can be obtained as the average of the cost allocations provided by voccp over all orders on the set of agents.

The vertex oriented approach differs from the original approach to obtain ERO in the following ways. First, while at each step of voccp an agent constructs the cheapest allowed edge, Kruskal's algorithm selects and adds edges to the spanning tree in increasing order of their costs. That is Kruskal's algorithm is an edge oriented algorithm. Second, ERO distributes the cost of an edge constructed by Kruskal's algorithm via a prespecified scheme. However, with our approach, the cost assigned to an agent is determined as the average of her own choices in each run of the voccp. Our new approach thus provides a new interpretation for ERO. Now ERO can be interpreted as an expected value of the cost allocations provided by voccp for each order on the set of agents and where the orders on the set of agents have equal probability.

In Section 3.3 we investigate extensions of the results obtained in Section 3.2 for mcst situations on a multisource extension of mcst situations, minimum cost spanning forest situations (cf. Rosenthal, 1987). A minimum cost spanning forest situation allows for more than one source, each source providing identical services. The users' objective is still to build a minimum-cost network (which is a forest in this case) which connects each of them to at least one source and to allocate its cost fairly. The availability of different identical sources may be interpreted as the existence of several suppliers which compete according to the costs of links connecting them to the users. We first show that both Kruskal's algorithm and voccp can be defined for minimum cost spanning forest situations in a way that they yield efficient algorithms. Second, we extend the definition of ERO to this multisource situation and prove that ERO can again be obtained as the average of the cost allocations provided by voccp.

Section 3.4 investigates the extensions of the results obtained in Section 3.2 for mcst situations with two sources. In a minimum cost spanning tree problem with two sources, there exist exactly two sources which may provide identical or different services and the users have to build a minimum-cost network which connects each of them to both of the sources. The optimal network in this multisource situation is again an mcst. Hence, Kruskal's algorithm is also an efficient algorithm for these mcst situations. We first propose an equal remaining obligations rule for mcst situations with two sources. Then we extend voccp to these multisource situations and prove that ERO can again be obtained as the average of the cost allocations provided by voccp.

In Section 3.5 we consider Highway Problems and the corresponding cooperative cost games called highway games which were introduced by Mosquera and Zarzuelo (2006) to address the problem of fair allocation of the construction costs of a highway network. In a highway problem, the possibilities regarding the construction of a highway network are determined by a connected graph. The set of vertices of the graph represents the potential entry and exit points and edges in the graph represent the possible highway connections that can be constructed. Each edge in the graph has an associated cost which in general will depend on its length or the geographical properties that may affect the construction costs of the highway. Each player in a highway problem has to establish a connection between two given vertices in the graph, i.e., between his entry and exit point. Given a highway problem, a corresponding highway game is defined as a cooperative cost game which associates to each coalition of players the total cost of the cheapest selection of edges in the graph which connects the entry and exit point of every member of the coalition. Mosquera and Zarzuelo (2006) restricted attention to highway problems in which the underlying graph is a tree. For complete graphs, highway problems are a special type of minimum cost forest (mcf) problems as introduced by Kuipers (1997). Mcf problems are generalizations of mcst problems which allow for more than one source, where each source offers a different type of service and each customer has to be connected with a given nonempty subset of the available sources.

In this study, we analyze highway problems in which the underlying graphs are weakly cyclic. A graph is called weakly cyclic if it is connected and every edge in the graph is contained in at most one cycle. In particular, these graphs may contain cycles and hence, there will exist multiple paths between some entry and exit points. Note that, in this setting, a coalition of players can further reduce the joint construction costs by an optimal coordination of paths to construct. That is, the joint minimal cost of a coalition is now obtained as a result of solving a combinatorial optimization problem. Hence highway games induced by weakly cyclic graphs belong to the research area of operations research games which focuses on the interplay between the optimization of costs of a project and the allocation of costs among the participants of the project. From among the numerous studies on this topic, we mention minimum cost spanning tree games (Granot and Huberman, 1981), traveling salesman games (Potters et al., 1992), Chinese postman games (Granot et al., 1999), sequencing games (Curiel et al., 1989) and project games (Estevez-Fernandez et al., 2007). An overview of operations research games can be found in Borm et al. (2001).

We start our analysis of highway games that are induced by weakly cyclic graphs by investigating their concavity properties in Section 3.5.2. We proceed as Herer and Penn (1995) in the setting of traveling salesman problems and Granot et al. (1999) for Chinese postman problems, and focus on the question for which class of graphs the corresponding games are always concave. We define a graph to be highway game-concave (HG-concave), if for every player set, for every choice of entry and exit points for the players and for every cost specification, the corresponding highway game is concave. The main result of this section is that a graph is HG-concave if and only if it is weakly triangular. Here, a graph is called weakly triangular if it is weakly cyclic and, moreover, if every cycle is a triangle, i.e., every cycle is composed of precisely three edges.

In Section 3.5.3 we investigate the core of the highway games. Highway games induced by trees are always balanced. For general highway games induced by graphs which allow for multiple paths between vertices, the core can be empty. We prove that highway games induced by weakly cyclic graphs are balanced.

3.1 Preliminaries

This section recalls basic notions from graph theory which will be used throughout this chapter.

A *graph* G is a pair (V, E), where V is a nonempty and finite set of *vertices* and $E \subset E_V = \{\{u, v\} | u, v \in V, u \neq v\}$ is a set of *edges*. (V, E_V) is called the *complete graph* on V. An edge $\{u, v\} \in E_V$ is said to *connect* vertices u and v and u, v are called the *end vertices* of the edge $\{u, v\}$. Also we say that an edge is *incident* with its end vertices (and vice versa).

A *subgraph* of $G = (V, E)$ is a graph $G' = (V', E')$ with $\emptyset \neq V' \subset V$ and $E' \subset E_{V'} \cap E$. If G' is a subgraph of G, then we say that G *contains* G'. If a subgraph $G' = (V', E')$ of $G = (V, E)$ is such that $E' = E_{V'} \cap E$, then we say that V' *induces* G' in G and write $G' = G[V']$.

A *path* P of length $k \geq 1$ is a graph (V, E) with

$$V = \{v_1, ..., v_{k+1}\}, |V| = k + 1 \text{ and } E = \{\{v_1, v_2\}, \{v_2, v_3\}, ..., \{v_k, v_{k+1}\}\},$$

and is often abbreviated by $P = v_1 v_2 ... v_{k+1}$; it is also referred as *a path from* v_1 to v_{k+1}.

A *cycle* C of length $k \geq 3$ is a graph (V, E) with

$$V = \{v_1, ..., v_k\}, |V| = k \text{ and } E = \{\{v_1, v_2\}, \{v_2, v_3\}, ..., \{v_{k-1}, v_k\}, \{v_k, v_1\}\}$$

and is often abbreviated by $C = v_1v_2...v_kv_1$.

Two vertices $u, v \in V$ are called *connected* in G if $u = v$ or if G contains a path between u and v as a subgraph. A graph G is called *connected* if every pair of distinct vertices of G is connected in G. If $G = (V, E)$ is connected, an edge $e \in E$ is called a *bridge* in G if the graph $(V, E\backslash\{e\})$ is not connected. The notion of connectedness induces an equivalence relation on the vertex set V. Thus there is a partition of V into nonempty subsets $V_1, V_2, ..., V_k$ such that two vertices u and v are connected if and only if both u and v belong to the same set V_i. The subgraphs $G[V_1], G[V_2], ..., G[V_k]$ are called *components* of G. We denote by G_u the component of G to which vertex u belongs.

An *acyclic* graph, i.e., a graph which does not contain cycles is called a *forest*. A connected forest is called a *tree*. Notice that the components of a forest are all trees, i.e., a forest is a disjoint union of trees. A subgraph $G' = (V', E')$ of $G = (V, E)$ is called a *spanning tree* of G if it is a tree with $V' = V$. A connected graph is called *weakly cyclic* if every edge in the graph is contained in at most one cycle [1]. A weakly cyclic graph is called *weakly triangular* if every cycle in G has length 3.

Given a graph $G = (V, E)$, an edge $e \in E_V\backslash E$ and an edge $e' \in E$, we denote by $G + e$ the graph $(V, E \cup \{e\})$ and by $G - e'$ the graph $(V, E\backslash\{e'\})$.

In the following we do not always distinguish strictly between a graph and its vertex or its edge set. For example, we may write $v \in G$ (rather than $v \in V$) or $e \in G$ (rather than $e \in E$).

3.1.1 Minimum cost spanning tree situations

Mcst situations involve a group of agents who have to be connected to a supplier of a service (source). Every mcst situation can be represented by a triple $(N, 0, w)$, where $N = \{1, ..., n\}$ is the agent set and 0 represents the source. In the following, we denote by N' the set $N \cup \{0\}$. The function $w : E_{N'} \to \mathbb{R}_+$ is called a *weight function* and associates with each edge $e \in E_{N'}$ the weight $w(e)$ which represents the cost of constructing e. If $w(e) \in \{0, 1\}$ for every $e \in E_{N'}$, then the weight function w is called a *simple weight function* and the mcst situation $(N, 0, w)$ is called a *simple mcst situation*. Obviously, the minimum cost network that would connect all agents to the source has to form a spanning tree of $(N', E_{N'})$. Therefore, given an mcst problem $(N, 0, w)$, we are interested in finding a minimum cost spanning tree (mcst) of $(N, 0, w)$. Formally, the cost of a spanning tree, Γ is given by $w(\Gamma) = \sum_{e \in E(\Gamma)} w(e)$ and

[1]The name "weakly cyclic" graph is not a standard graph theoretical term. It was first introduced in the context of Chinese postman games in Granot et al. (1999) and was maintained in subsequent related papers. Weakly cyclic graphs are also called *cactus graphs*.

Γ is called an mcst if it satisfies $w(\Gamma) = \min\{w(\Gamma')|\Gamma'$ is a spanning tree of $(N', E_{N'})\}$. Observe that an mcst situation with agent set N, $(N, 0, w)$, can be identified with the weight function, w. Hence, we will denote the set of mcst situations with agent set N as $\mathcal{W}^N = \mathbb{R}_+^{E_{N'}}$.

Example 3.1.1 Assume that three villages have to be connected to a water supplier directly or via the other villages. The cost of constructing water pipelines between the villages and between the villages and the water supplier are given in Figure 3.1. Here the villages are denoted by 1, 2 and 3 and the water supplier by 0.

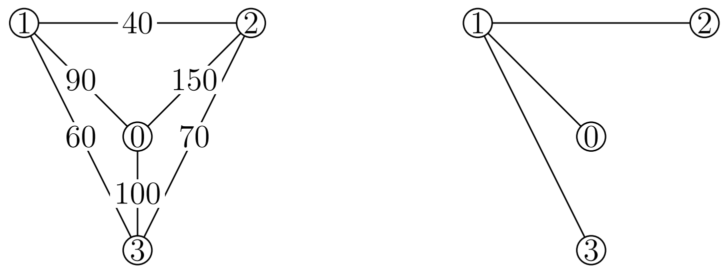

Figure 3.1: An mcst situation (left side) and a related mcst (right side)

An mcst in this mcst situation is the tree with edges $\{1, 2\}, \{1, 3\}, \{0, 1\}$ with cost 190.

We will use the following well-known results in graph theory:

Property 3.1.1 *Let $(N, 0, w)$ be an mcst situation and let Γ be a subgraph of $(N', E_{N'})$. Then,*

(1) Γ is a spanning tree of $(N', E_{N'})$ if and only if Γ has n edges and does not contain cycles.

(2) A spanning tree Γ of $(N', E_{N'})$ is an mcst of $(N, 0, w)$ if and only if $w(e) \geq w(e')$ for every $e \in E_{N'} \backslash E(\Gamma)$ and every $e' \in C$, where C is the unique cycle in $\Gamma + e$.

3.1.2 Algorithms for mcst situations

The first problem that arises in mcst situations is how to find an mcst. The operations research literature on mcst problems has provided many algorithmic solutions to the problem and has discussed the computational properties of these solutions. An historic overview of the algorithms provided for the mcst problem can be found in Graham and Hell (1985). In this section, we briefly introduce the two most famous algorithms, Kruskal's algorithm (Kruskal, 1956) and Prim's algorithm (Prim, 1957).

Let $(N, 0, w)$ be an mcst situation. A minimum cost spanning tree for $(N, 0, w)$ can be obtained in the following two ways.

> **Prim's Algorithm:** In the first iteration a cheapest edge which connects the source with an agent is constructed. In every subsequent iteration of Prim's algorithm, a cheapest edge which connects an agent who is not connected to the source yet with either the source or another agent which is already connected to the source in the previous iterations of the algorithm is constructed. In every step of the algorithm, precisely one agent gets connected with the source. So, the algorithm results in an mcst after $|N|$ steps.
>
> **Kruskal's Algorithm:** Kruskal's algorithm selects and adds edges to the spanning tree in increasing order of their costs such that an edge is added only if it does not create a cycle with the previously added edges.

Example 3.1.2 Consider the mcst situation $(N, 0, w)$ with $N = \{1, 2, 3\}$ and w as depicted in Figure 3.1.

Prim's algorithm first constructs the edge $\{0, 1\}$ since it is the cheapest of all edges that connect an agent with the source. Then the algorithm selects $\{1, 2\}$ and finally $\{1, 3\}$ resulting in the mcst with the set of edges $\{\{0, 1\}, \{1, 2\}, \{1, 3\}\}$.

Kruskal's algorithm first constructs the cheapest edge $\{1, 2\}$ and then the second cheapest edge $\{1, 3\}$. The third cheapest edge is $\{2, 3\}$. However, this edge is not constructed by the algorithm since it creates a cycle with the previously constructed edges. At the third step of the algorithm, the edge $\{0, 1\}$ is constructed resulting in the mcst with the set of edges $\{\{0, 1\}, \{1, 2\}, \{1, 3\}\}$. $\diamond$

3.2 A Vertex Oriented Approach to ERO for MCST Situations

In this section, we prove that ERO can be obtained as the average of the cost allocations provided by voccp for every order on the set of agents. Section 3.2.1 introduces ERO formally. Section 3.2.2 introduces another class of construct and charge procedures, the P^{σ}-rules (where σ is an order on the set of agents) introduced by Branzei et al. (2004). These procedures are also based on Kruskal's algorithm and they will be utilized while proving that ERO can be obtained as the average of the cost allocations provided by voccp. Section 3.2.3 introduces voccp and also analyzes its algorithmic properties. Section 3.2.4 proves our main results. Finally, Section 3.2.5 investigates the connections between voccp and the so-called optimistic game associated with an mcst situation (cf. Bergantiños and Vidal-Puga, 2007a).

3.2.1 ERO for cost sharing in mcst situations

Feltkamp et al. (1994) introduced ERO to solve the cost sharing problem related to mcst situations. In the following, we will provide the notation and the definitions required to introduce ERO.

Let $\Pi(E_{N'})$ stand for the set of all bijections $\pi : \{1, ..., |E_{N'}|\} \to E_{N'}$. Obviously, for each mcst situation $(N, 0, w)$, there exists a bijection $\pi \in \Pi(E_{N'})$ that orders edges in increasing order with respect to their costs, i.e., $w(\pi(1)) \leq w(\pi(2)) \leq ... \leq w(\pi(|E_{N'}|))$. The column vector $(w(\pi(1)), w(\pi(2)), ..., w(\pi(|E_{N'}|)))^t$ is denoted by w^π.

For any $\pi \in \Pi(E_{N'})$, one can define the set $K^\pi = \{w \in \mathcal{W}^N | w(\pi(1)) \leq w(\pi(2)) \leq ... \leq w(\pi(|E_{N'}|))\}$, i.e., the set of weight functions which result in the same increasing order on the set of edges with respect to their costs. It can easily be observed that K^π is a cone in $\mathcal{W}^N$. Obviously, $\bigcup_{\pi \in \Pi(E_{N'})} K^\pi = \mathcal{W}^N$. For each $\pi \in \Pi(E_{N'})$, the set of simple weight functions $e^{\pi,k} \in K^\pi$ defined by

$$e^{\pi,k}(\pi(1)) = e^{\pi,k}(\pi(2)) = ... = e^{\pi,k}(\pi(k-1)) = 0$$

$$e^{\pi,k}(\pi(k)) = e^{\pi,k}(\pi(k+1)) = ... = e^{\pi,k}(\pi(|E_{N'}|)) = 1,$$

for every $k \in \{1, 2, ..., |E_{N'}|\}$ forms a basis of K^π. That is each weight function $w \in K^\pi$ can be written as a unique linear combination of these simple weight functions as

$$w = w(\pi(1))e^{\pi,1} + \sum_{k=2}^{|E_{N'}|} \left((w(\pi(k)) - w(\pi(k-1)))e^{\pi,k} \right). \tag{3.1}$$

We introduce ERO in two steps. First a rule ERO^π is defined on each cone K^π and then it is proved that these ERO^π rules can be extended to the whole cone of mcst situations.

Let $(N, 0, w)$ be an mcst situation and $\pi \in \Pi(E_{N'})$ be such that $w \in K^\pi$. In order to define ERO^π on K^π, we will consider Kruskal's algorithm when it selects edges with respect to order π. The ERO^π-rule distributes the cost of edges that are constructed by Kruskal's algorithm among the agents whose *connectivity*, i.e., the number of nodes in N that an agent is connected to, increases with the construction of the edge. To do so, we will consider a sequence of $|E_{N'}|+1$ graphs: $(N', F^{\pi,0}), (N', F^{\pi,1}), ..., (N', F^{\pi,|E_{N'}|})$ such that $F^{\pi,0} = \emptyset$ and $F^{\pi,k} = F^{\pi,k-1} \cup \{\pi(k)\}$ for every $k \in \{1, ..., |E_{N'}|\}$. In the following, for the sake of simplicity, we will denote the graphs $(N', F^{\pi,k})$ shortly by $F^{\pi,k}$, i.e., by their edge sets. The connectivity of an agent i in graph $F^{\pi,k}$ is denoted by $n_i(F^{\pi,k})$. Note that $n_i(F^{\pi,k}) = 1$ when i is not connected to any other agent in N in $F^{\pi,k}$. The connection vectors $b^{\pi,k} \in \mathbb{R}^N$ are defined for each $k \in \{0, 1, ..., |E_{N'}|\}$ by

$$b_i^{\pi,k} = \begin{cases} 0 & \text{if } i \text{ is connected to } 0 \text{ in } F^{\pi,k} \\ \frac{1}{n_i(F^{\pi,k})} & \text{otherwise} \end{cases} \tag{3.2}$$

for each $i \in N$. ERO^π will distribute the cost of a Kruskal edge proportionally to the change in the *connection vectors* resulting from the introduction of the edge by the algorithm.

The contribution matrix with respect to $\pi \in \Pi(E_{N'})$ is the matrix $M^\pi \in \mathbb{R}^{N \times E_{N'}}$ where rows correspond to agents and columns to edges. It lists the change in the connectivity of the agents, i.e., the k-th column of M^π equals

$$M^\pi e^k = b^{\pi,k-1} - b^{\pi,k} \tag{3.3}$$

for each $k \in \{1, ..., |E_{N'}|\}$. Here e^k stands for the column vector defined as $e_i^k = 1$ if $i = k$ and $e_i^k = 0$ for each $i \in \{1, ..., |E_{N'}|\} \backslash \{k\}$.

Observe that the zero columns in M^π correspond to edges which are not constructed by Kruskal's algorithm. Moreover, each column $M^\pi e^k$ with $(M^\pi e^k)_i \neq 0$ for some $i \in N$ corresponds to the edge $\pi(k)$ constructed at stage k in the Kruskal algorithm. Notice that the sum of the elements of such a column equals 1. Then, $(M^\pi e^k)_i$ $(i \in N)$, the difference between i's connectivity resulting from the construction of $\pi(k)$, represents the fraction of the cost of the edge $\pi(k)$ to be paid by agent i. Notice also that the sum of elements of each row of M^π equals 1.

We are now ready to define the rule ERO^π on K^π.

Definition 3.2.1 *For each $\pi \in \Pi(E_{N'})$, ERO^π is defined as the map $ERO^\pi : K^\pi \to \mathbb{R}^N$, where $ERO^\pi(w) = M^\pi w^\pi$ for each mcst situation w in the cone K^π.*

If in an mcst $(N, 0, w)$, some edges have the same cost then we know that there exist multiple orders that are compatible with w, i.e., there exist $\pi, \pi' \in \Pi(E_{N'})$ such that $w \in K^\pi \cap K^{\pi'}$. Then, for all orders compatible with edges, the corresponding ERO^π rule can be utilized to distribute the cost of the mcst obtained. So, the important question that arises at this point is whether different ERO^π rules corresponding to different orders that are compatible with an mcst situation lead to the same cost allocation or not. The following lemma shows that although the allocation of the cost of a single edge by different ERO^π rules may change with respect to the order of edges under consideration, the allocation of the total cost of all edges with same cost is the same regardless of the order considered.

Lemma 3.2.1 *(Branzei et al., 2004, Lemma 1) Let $\pi \in \Pi(E_{N'})$, $w \in K^\pi$. Assume that $w^\pi_t = w^\pi_{t+1}$ for some $t \in \{1, ..., |E_{N'}|-1\}$. Then for the ordering $\pi' \in \Pi(E_{N'})$ such that $\pi'(t) = \pi(t+1), \pi'(t+1) = \pi(t)$ and $\pi'(i) = \pi(i)$ for every $i \in \{1, ..., |E_{N'}|\} \backslash \{t, t+1\}$, we have that $w \in K^{\pi'}$ and $ERO^\pi(w) = ERO^{\pi'}(w)$.*

Hence, for every order π that a weight function w is compatible with, $\mathrm{ERO}^\pi(w)$ results in the same allocation.

Proposition 3.2.1 *(Branzei et al., 2004, Proposition 1) Let $(N, 0, w)$ be an mcst situation. Then, $ERO^\pi(w) = ERO^{\pi'}(w)$ for every $\pi, \pi' \in \Pi(E_{N'})$ with $w \in K^\pi \cap K^{\pi'}$.*

Proposition 3.2.1 implies that ERO^π rules can be extended to the whole cone of mcst situations as one ERO rule.

Definition 3.2.2 *ERO is defined as the map $ERO : \mathcal{W}^N \to \mathbb{R}^N$, where*

$$ERO(w) = ERO^\pi(w) = M^\pi w^\pi \tag{3.4}$$

for every $w \in \mathcal{W}^N$ and $\pi \in \Pi(E_{N'})$ such that $w \in K^\pi$.

Example 3.2.1 Consider the mcst situation $(N, 0, w)$ with $N = \{1, 2, 3\}$ and w as depicted in Figure 3.1. Then $w \in K^\pi$, with $\pi(1) = \{1, 2\}$, $\pi(2) = \{1, 3\}$, $\pi(3) = \{2, 3\}$, $\pi(4) = \{0, 1\}$, $\pi(5) = \{0, 3\}$ and $\pi(6) = \{0, 2\}$.

The sequence of the graphs $F^{\pi,k}$ formed by Kruskal's algorithm and the corresponding connection vectors are given in Table 3.1.

$F^{\pi,k}$	$b^{\pi,k}$
$\emptyset$	$(1, 1, 1)^t$
$\{\{1, 2\}\}$	$(\frac{1}{2}, \frac{1}{2}, 1)^t$
$\{\{1, 2\}, \{1, 3\}\}$	$(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})^t$
$\{\{1, 2\}, \{1, 3\}, \{2, 3\}\}$	$(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})^t$
$\{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{0, 1\}\}$	$(0, 0, 0)^t$
$\{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{0, 1\}, \{0, 3\}\}$	$(0, 0, 0)^t$
$\{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{0, 1\}, \{0, 3\}, \{0, 2\}\}$	$(0, 0, 0)^t$

Table 3.1: Graphs formed in each step of Kruskal's algorithm and the corresponding connection vectors in Example 3.2.1.

Then the contribution matrix M^π is given by

$$M^\pi = \begin{pmatrix} \frac{1}{2} & \frac{1}{6} & 0 & \frac{1}{3} & 0 & 0 \\ \frac{1}{2} & \frac{1}{6} & 0 & \frac{1}{3} & 0 & 0 \\ 0 & \frac{2}{3} & 0 & \frac{1}{3} & 0 & 0 \end{pmatrix}.$$

Finally, $w^\pi = (40, 60, 70, 90, 100, 150)^t$. Hence, $ERO(w) = M^\pi w^\pi = (60, 60, 70)^t$. $\diamond$

It follows by the definition of ERO that it satisfies the *Cone-wise Positive Linearity* property (cf. Branzei et al., 2004). That is

$$ERO(\alpha w + \alpha' w') = \alpha ERO(w) + \alpha' ERO(w'),$$

for every $\pi \in \Pi(E_{N'})$, for every $w, w' \in K^\pi$ and for every $\alpha, \alpha' \geq 0$. Hence $ERO(w)$ can also be calculated by making use of the linear decomposition of w into simple weight functions as given in Equation 3.1. That is

$$ERO(w) = w(\pi(1))ERO(e^{\pi,1}) + \sum_{k=2}^{|E_{N'}|} \left((w(\pi(k)) - w(\pi(k-1)))ERO(e^{\pi,k})\right), \quad (3.5)$$

for every $w \in \mathcal{W}^N$ and $\pi \in \Pi(E_{N'})$ such that $w \in K^\pi$.

3.2.2 P^σ-rules for mcst situations

In this section we present a class of construct and charge rules, the P^σ-rules which are introduced by Branzei et al. (2004).

Let $(N, 0, w)$ be an mcst situation with $w \in K^\pi$. We denote by $\Pi(N)$ the set of all orders $\sigma : \{1, 2, ..., n\} \to N$, where $\sigma(k) = i$ means that agent i is in the k-th position with respect to σ. Moreover, for every $\sigma \in \Pi(N)$, we say that $i \in N$ is the last agent in $F_i^{\pi,t}$ with respect to σ, if $\sigma^{-1}(i) \geq \sigma^{-1}(j)$ for every agent $j \in F_i^{\pi,t}$. Recall here that $F^{\pi,t}$ denotes the graph obtained after adding the t cheapest edges (with respect to π) during Kruskal's algorithm and $F_i^{\pi,t}$ is the component of $F^{\pi,t}$ which contains agent i.

P^σ-rules ($\sigma \in \Pi(N)$) are closely related with ERO and to present this rule we follow the same plan used for ERO. Analogously to Definition 3.2.1, for each $\pi \in \Pi(E_{N'})$, we first define a rule $P^{\sigma,\pi}$ on cone K^π by $P^{\sigma,\pi}(w) = M^{\sigma,\pi} w^\pi$ for each mcst situation $w \in K^\pi$. Similarly to the definition of the contribution matrix M^π for ERO^π, $M^{\sigma,\pi} \in \mathbb{R}^{N \times E_{N'}}$ is defined as $M^{\sigma,\pi} e^k = b^\sigma(F^{\pi,k-1}) - b^\sigma(F^{\pi,k})$, where

$$b_i^\sigma(F^{\pi,k}) = \begin{cases} 1 & \text{if } i \text{ is the last agent in } F_i^{\pi,k} \text{ with respect to } \sigma \text{ and } 0 \notin F_i^{\pi,k}, \\ 0 & \text{otherwise,} \end{cases}$$

for every $k \in \{0, 1, ..., |E_{N'}|\}$ and every $i \in N$.

A variant of Lemma 3.2.1 holds also for rules $P^{\sigma,\pi}$. Hence, similarly to ERO^π-rules, for every order π that a weight function w is compatible with, $P^{\sigma,\pi}(w)$ also results in the same allocation. This enables us to define a P^σ-rule over all mcst situations.

Definition 3.2.3 *P^σ is defined as the map $P^\sigma : \mathcal{W}^N \to \mathbb{R}^N$, where*

$$P^\sigma(w) = P^{\sigma,\pi}(w) = M^{\sigma,\pi} w^\pi \quad (3.6)$$

for every $\sigma \in \Pi(N)$, $w \in \mathcal{W}^N$ and $\pi \in \Pi(E_{N'})$ such that $w \in K^\pi$.

Example 3.2.2 Consider the mcst situation $(N, 0, w)$ with $N = \{1, 2, 3\}$ and w as depicted in Figure 3.1. Let $\sigma \in \Pi(N)$ be such that $\sigma(i) = i$ for every $i \in N$. Example 3.2.1 provides an order $\pi \in \Pi(E_{N'})$ such that $w \in K^{\pi}$ and in Table 3.1 we already provided the sequence of graphs $F^{\pi,k}$ formed by Kruskal's algorithm. In Table 3.2, we provide the corresponding $b^{\sigma}(F^{\pi,k})$ vectors. Then the corresponding matrix $M^{\sigma,\pi}$

$F^{\pi,k}$	$b^{\sigma}(F^{\pi,k})$
$\emptyset$	$(1,1,1)^t$
$\{\{1,2\}\}$	$(0,1,1)^t$
$\{\{1,2\},\{1,3\}\}$	$(0,0,1)^t$
$\{\{1,2\},\{1,3\},\{2,3\}\}$	$(0,0,1)^t$
$\{\{1,2\},\{1,3\},\{2,3\},\{0,1\}\}$	$(0,0,0)^t$
$\{\{1,2\},\{1,3\},\{2,3\},\{0,1\},\{0,3\}\}$	$(0,0,0)^t$
$\{\{1,2\},\{1,3\},\{2,3\},\{0,1\},\{0,3\},\{0,2\}\}$	$(0,0,0)^t$

Table 3.2: Graphs formed in each step of Kruskal's algorithm and the corresponding $b^{\sigma}(F^{\pi,k})$ vectors in Example 3.2.2.

is given by

$$M^{\sigma,\pi} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

Recall that $w^{\pi} = (40, 60, 70, 90, 100, 150)^t$. Hence, $P^{\sigma}(w) = M^{\sigma,\pi}w^{\pi} = (40, 60, 90)^t$. $\diamond$

There is an important connection between ERO and P^{σ}-rules. Tijs et al. (2006) proved that ERO is the average of the P^{σ}-rules over all possible orderings of agents.

Proposition 3.2.2 *(Tijs et al., 2006, Proposition 7) Let $w \in \mathcal{W}^N$. Then*

$$ERO(w) = \sum_{\sigma \in \Pi(N)} \frac{P^{\sigma}(w)}{n!}.$$

In the following proposition we prove that if P^{σ} is utilized in an mcst situation, then every agent pays for one edge of the mcst constructed by Kruskal's algorithm.

Proposition 3.2.3 *Let $(N, 0, w)$ be an mcst situation, $\pi \in \Pi(E_{N'})$ with $w \in K^{\pi}$ and $\sigma \in \Pi(N)$. Then $P^{\sigma}(w)$ assigns each agent the cost of one of the edges of the mcst constructed by Kruskal's algorithm.*

Proof. First let us show that P^{σ} does not assign any cost for the edges not constructed by the algorithm. For this aim consider a column $M^{\sigma,\pi}e^k$ that correspond to an edge $\pi(k)$ which is not constructed by Kruskal's algorithm. Since $\pi(k)$ is not constructed it must create a cycle with the previously constructed edges in $F^{\pi,k}$ and hence does not affect vectors $b^\sigma(F^{\pi,k})$, i.e., $b^\sigma(F^{\pi,k-1}) = b^\sigma(F^{\pi,k})$ and hence $M^{\sigma,\pi}e^k = 0$. So, P^σ does not allocate any cost for the edges not constructed by Kruskal's algorithm.

Now consider a column $M^{\sigma,\pi}e^k$ that corresponds to an edge $\pi(k)$ constructed by Kruskal's algorithm. Since it is constructed it must connect two disjoint components in $F^{\pi,k-1}$, let's say F_1 and F_2. Let us denote the last agent in F_1 with respect to σ by i and the one in F_2 by j. Assume without loss of generality that $\sigma^{-1}(i) \geq \sigma^{-1}(j)$, i.e., j precedes i with respect to order σ.

Now assume that the source does not belong to these components. Then we have that $b^\sigma(F^{\pi,k-1})_i = b^\sigma(F^{\pi,k-1})_j = 1$ and $b^\sigma(F^{\pi,k-1})_r = 0$ for every other agent r in $F_1 \cup F_2$. Observe now that i is the last agent in $F_1 \cup F_2$ with respect to σ and $0 \notin F_1 \cup F_2$. Hence, $b^\sigma(F^{\pi,k})_i = 1$ and $b^\sigma(F^{\pi,k})_r = 0$ for every other agent r in $F_1 \cup F_2$. Hence, $(M^{\sigma,\pi}e^k)_j = 1$ and $(M^{\sigma,\pi}e^k)_r = 0$ for every other agent $r \in N\backslash\{j\}$. That is the cost of edge $\pi(k)$ is assigned totally to agent j. Observe that $b^\sigma(F^{\pi,l})_j = 1$ for every $l < k-1$ and $b^\sigma(F^{\pi,l})_j = 0$ for every $l > k$. So, agent j is not allocated any other cost, i.e., $P^\sigma(w)_j = w(\pi(k))$.

Now assume that F_1 contains the source. Then $b^\sigma(F^{\pi,k-1})_r = 0$ for every agent $r \in F_1 \cup F_2$ except j. $b^\sigma(F^{\pi,k-1})_j = 1$ since j is the last agent in F_2 and $0 \notin F_2$. Moreover, $b^\sigma(F^{\pi,k})_r = 0$ for every agent $r \in F_1 \cup F_2$ since $0 \in F_1 \cup F_2$. So, the cost of edge $\pi(k)$ is assigned totally to agent j. As discussed above, j is not allocated any other cost, so $P^\sigma(w)_j = w(\pi(k))$.

One can use the arguments given above to show that if F_2 contains the source, then the cost of edge $\pi(k)$ is assigned totally to agent i and i is not allocated any other cost. So, $P^\sigma(w)_i = w(\pi(k))$ and hence we are done. $\square$

In the following proposition we consider simple mcst situations and show which agents are assigned edges that cost zero and which agents are assigned edges that cost one by $P^\sigma(w)$.

Proposition 3.2.4 *Let $(N, 0, w)$ be a simple mcst situation, $\pi \in \Pi(E_{N'})$ with $w \in K^\pi$ and $\sigma \in \Pi(N)$. Assume that the number of edges in $E_{N'}$ that cost zero is t. Then*

$$P^\sigma(w)_i = \begin{cases} 1 & \textit{if } i \textit{ is the last agent in } F_i^{\pi,t} \textit{ with respect to } \sigma \textit{ and } 0 \notin F_i^{\pi,t}, \\ 0 & \textit{otherwise,} \end{cases}$$

for every $i \in N$.

We omit the proof of Proposition 3.2.4 since it is straightforward.

It follows by the definition of P^σ that it satisfies the cone-wise positive linearity property. Hence $P^\sigma(w)$ can also be calculated by making use of the linear decomposition of w into simple weight functions as given in Equation 3.1. Hence

$$P^\sigma(w) = w(\pi(1))P^\sigma(e^{\pi,1}) + \sum_{k=2}^{|E_{N'}|} \left((w(\pi(k)) - w(\pi(k-1)))P^\sigma(e^{\pi,k})\right), \tag{3.7}$$

for every $\sigma \in \Pi(N)$, $w \in \mathcal{W}^N$ and $\pi \in \Pi(E_{N'})$ such that $w \in K^\pi$.

3.2.3 The vertex oriented construct and charge procedure

In this section we introduce voccp and also analyze its algorithmic properties.

Let $G = (N', E)$ be a graph on N'. Recall that the component of G which contains agent i is denoted by G_i. Then the set of edges which connect a vertex of this component with a vertex of another component is given by $\{\{u, v\}|u \in G_i \text{ and } v \in N'\backslash G_i\}$. We call this set *the set of component reducing edges for agent i in G* and denote it by $\mathcal{A}_i(G)$.

Just like Kruskal's algorithm and Prim's algorithm, voccp is a greedy algorithm which works on the principle of gradual merging several trees in a forest in the cheapest possible way until an mcst is achieved. The procedure makes use of a predetermined order on the set of agents to determine the agent who is going to construct an edge. At each step of the procedure, the first agent in the order who has not yet constructed an edge constructs and pays the cheapest allowed edge, i.e., the cheapest edge which connects the tree to which he belongs to another tree of the forest obtained in the previous iterations of the procedure.

Let $(N, 0, w)$ be an mcst situation. Then voccp for mcst situations is defined formally as follows:

(**Step 1**) Pick $\sigma \in \Pi(N)$.
(**Step 2**) Set $G^{\sigma,0} = (N', \emptyset)$ and $v^{\sigma,0} = 0$ $(\in \mathbb{R}^N)$.
(**Step 3**) For $k = 1$ to n:

Let $\sigma(k) = i$. Choose an edge e_k^σ such that

$$e_k^\sigma = arg\min\{w(e)|e \in \mathcal{A}_i(G^{\sigma,k-1})\}.$$

Set $G^{\sigma,k} = G^{\sigma,k-1} + e_k^\sigma$ and $v^{\sigma,k} = v^{\sigma,k-1} + w(e_k^\sigma)e^{\{i\}}$.

(**Step 4**) Set $\Gamma^\sigma = G^{\sigma,n}$ and $v^\sigma = v^{\sigma,n}$.

Example 3.2.3 Consider the mcst situation $(N, 0, w)$ with $N = \{1, 2, 3\}$ and w as depicted in Figure 3.1.

Let $\sigma \in \Pi(N)$ be such that $\sigma(i) = i$ for every $i \in N$. The related voccp is described as follows:

(**Step 1**) Let $\sigma \in \Pi(N)$ be such that $\sigma(i) = i$ for every $i \in N$.

(**Step 2**) Set $G^{\sigma,0} = (N', \emptyset)$ and $v^{\sigma,0} = 0$.

(**Step 3**) Step 3 consists of the following three iterations:

- $k = 1$: $\sigma(1) = 1$. Obviously $\mathcal{A}_1(G^{\sigma,0}) = \{\{1, 0\}, \{1, 2\}, \{1, 3\}\}$. Then, $e_1^\sigma = \{1, 2\}$, $G^{\sigma,1} = (N', \{\{1, 2\}\})$ and $v^{\sigma,1} = (40, 0, 0)$.

- $k = 2$: $\sigma(2) = 2$. Note that 2 and 1 are in the same component of $G^{\sigma,1}$ but 3 and 0 belong to different components. So $\mathcal{A}_2(G^{\sigma,1}) = \{\{1, 0\}, \{1, 3\}, \{2, 0\}, \{2, 3\}\}$. Then, $e_2^\sigma = \{1, 3\}$, $G^{\sigma,2} = (N', \{\{1, 2\}, \{1, 3\}\})$ and $v^{\sigma,1} = (40, 60, 0)$.

- $k = 3$: $\sigma(3) = 3$. Note that 3 is connected with 1 and 2 in $G^{\sigma,2}$ and hence these vertices belong to the same component. So $\mathcal{A}_3(G^{\sigma,2}) = \{\{1, 0\}, \{2, 0\}, \{3, 0\}\}$. Then, $e_3^\sigma = \{1, 0\}$, $G^{\sigma,3} = (N', \{\{1, 2\}, \{1, 3\}, \{1, 0\}\})$ and $v^{\sigma,3} = (40, 60, 90)$.

(**Step 4**) $\Gamma^\sigma = (N', \{\{1, 2\}, \{1, 3\}, \{1, 0\}\})$ and $v^\sigma = (40, 60, 90)$.

In Figure 3.2, we depict graphs $G^{\sigma,k}$ obtained in step 3 of voccp related to σ. Here, for each of these graphs, the agent who is going to construct an edge is indicated with a shaded vertex and edges that are allowed for the agent are indicated with dotted edges.

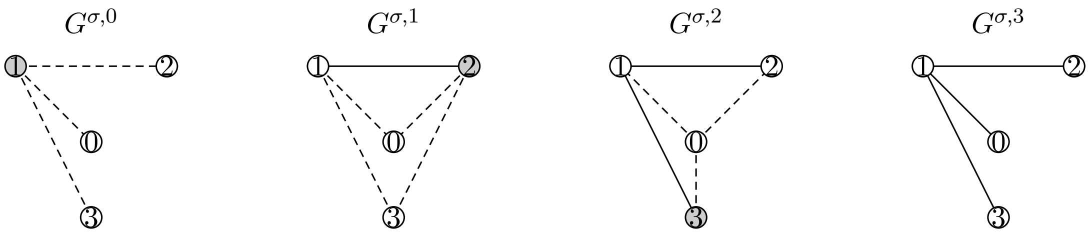

Figure 3.2: Graphs obtained during voccp in Example 3.2.3

In Table 3.3 we give the construct and charge results provided by voccp for all orderings of agents. $\diamond$

It's obvious that Γ^σ as obtained in Step 4 of voccp related to order σ has n edges and does not contain any cycle. Hence, it immediately follows from result (1) of Property 3.1.1 that Γ^σ is a spanning tree. Observe that voccp can lead to different

	Constructed edges by			Costs for		
σ	1	2	3	1	2	3
(1,2,3)	(1,2)	(1,3)	(1,0)	40	60	90
(1,3,2)	(1,2)	(1,0)	(1,3)	40	90	60
(2,1,3)	(1,3)	(1,2)	(1,0)	60	40	90
(2,3,1)	(1,0)	(1,2)	(1,3)	90	40	60
(3,1,2)	(1,2)	(1,0)	(1,3)	40	90	60
(3,2,1)	(1,0)	(1,2)	(1,3)	90	40	60

Table 3.3: Construct and charge results for the mcst situation in Figure 3.1

spanning trees. Because, at each iteration of the procedure in step 3, the agents are allowed to choose any cheapest allowed edge. So, if a cheapest allowed edge is not unique, the procedure may lead to different spanning trees depending on choices of agents.

We establish in the following theorem that voccp is an efficient algorithm for mcst problems, i.e., any spanning tree provided by voccp is an mcst of the mcst situation.

Theorem 3.2.1 *Let* $(N, 0, w)$ *be an mcst situation and* $\sigma \in \Pi(N)$. *Then,* Γ^σ *is an mcst of* $(N, 0, w)$.

Proof. For any spanning tree T of $(N', E_{N'})$ other than Γ^σ denote by $f(T)$ the smallest value of k such that e^σ_k is not in T. Now suppose that Γ^σ is not an mcst and let T be an mcst such that $f(T)$ is as large as possible.

Suppose that $f(T) = k$, i.e., $e^\sigma_1, ..., e^\sigma_{k-1}$ are in both T and Γ^σ, but that e^σ_k is not in T. Then, $T + e^\sigma_k$ contains a unique cycle C and obviously, C contains an edge e' different than e^σ_k such that $e' \in \mathcal{A}_{\sigma(k)}(G^{\sigma,k-1})$. Now, on the one hand, $w(e^\sigma_k) \leq w(e')$, since e' is allowed for $\sigma(k)$ in $G^{\sigma,k-1}$ and on the other hand, $w(e') \leq w(e^\sigma_k)$ by result (2) in Property 3.1.1. Hence, $w(e') = w(e^\sigma_k)$. Moreover, $T' = (T + e^\sigma_k) - e'$ is another spanning tree of $(N', E_{N'})$ and $w(T) = w(T')$. So, T', too, is an mcst. However, $f(T') > f(T)$, contradicting with the choice of T. Therefore, Γ^σ is indeed an mcst.□

Remark 3.2.1 It can be shown that voccp solves the mcst problem in $O\left((n+1)^3\right)$ time and hence has a greater (time) complexity than the well-known algorithms Prim's algorithm and Kruskal's algorithm which solve the problem in $O\left((n+1)^2\right)$ and $O\left((n+1)log(n+1)\right)$ time, respectively. The main reason behind the greater time requirement is that voccp has to manipulate more information compared to Kruskal's algorithm and Prim's algorithm, since it also provides a cost allocation besides an mcst of the problem.

We prove in Theorem 3.2.2 that every mcst in an mcst situation can be obtained by voccp for any order of agents followed.

Theorem 3.2.2 *Let $(N, 0, w)$ be an mcst situation and let T be an mcst. Then T can be the result of a vertex oriented construct and charge procedure for any permutation $\sigma \in \Pi(N)$.*

Proof. Suppose that the mcst T can not be constructed by voccp for a specific $\sigma \in \Pi(N)$. Starting with σ construct Γ by using voccp as far as possible. Then, there exists $k \in \{1, ..., n\}$ such that $e_1^\sigma, ..., e_{k-1}^\sigma$ are in T, but that e_k^σ is not in T. Then, $T + e_k^\sigma$ contains a unique cycle C and obviously, C contains an edge e' different than e_k^σ such that $e' \in \mathcal{A}_{\sigma(k)}(G^{\sigma,k-1})$. Since $\sigma(k)$ can not construct e' although it is allowed for $\sigma(k)$ in $G^{\sigma,k-1}$, it follows that $w(e_k^\sigma) < w(e')$. But, then $T' = T + e_k^\sigma - e'$ is a spanning tree of $(N', E_{N'})$ with total weight less than that of T, a contradiction. □

Remark 3.2.2 Theorem 3.2.2 shows that every mcst can be the result of voccp. Subsequently, the important question that arises is whether the cost allocation provided by voccp is independent of the mcst reached. In the next section, we will see that this is indeed the case.

3.2.4 A new approach to obtain ERO

In this section we prove the coincidence of the cost allocation provided by ERO with the average of the cost allocations provided by voccp over all orders of agents.

We show in the following lemma that, for simple mcst situations, the cost allocation v^σ provided by voccp related to order $\sigma \in \Pi(N)$ is equal to $P^\sigma(w)$.

Lemma 3.2.2 *Let $(N, 0, w)$ be a simple mcst situation. Then v^σ equals $P^\sigma(w)$ for every $\sigma \in \Pi(N)$.*

Proof. Let w be a simple mcst situation. Let us denote the number of edges in $E_{N'}$ that cost zero with t. Let $\pi \in \Pi(E_{N'})$ be such that $w \in K^\pi$ and let $\sigma \in \Pi(N)$.

By Proposition 3.2.4, we have to show that the cost v_i^σ assigned by voccp related to σ to an agent $i \in N$ is equal to 1 if i is the last agent in $F_i^{\pi,t}$ with respect to σ and $0 \notin F_i^{\pi,t}$, and is equal to 0 otherwise.

Pick an agent $i \in N$. Assume that i is the k^{th} agent with respect to order σ, i.e., $\sigma(k) = i$. Naturally, agent i constructs a zero-edge during voccp related to σ if there exist zero-edges that are component reducing for agent i in $G^{\sigma,k-1}$.

Now, assume that $0 \in F_i^{\pi,t}$. Observe then that the number of edges of a tree which connects the vertices of $F_i^{\pi,t}$ is equal to the number of agents in $F_i^{\pi,t}$. Hence, even if i is the last agent in $F_i^{\pi,t}$ and all agents in $F_i^{\pi,t}$ that precede i constructed a zero-edge in $F_i^{\pi,t}$, i will still not be connected to some agents in $F_i^{\pi,t}$, i.e., there will exist component reducing edges for agent i in $G^{\sigma,k-1}$ that cost 0. Hence, $v_i^\sigma = 0$.

Assume now that $0 \notin F_i^{\pi,t}$. Then the number of edges of a tree which connects the vertices of $F_i^{\pi,t}$ is one less than the number of agents in $F_i^{\pi,t}$. If agent i is the last agent in $F_i^{\pi,t}$ with respect to σ, then all other agents in $F_i^{\pi,t}$ that precede i with respect to σ will construct a zero-edge, and hence all vertices of $F_i^{\pi,t}$ will be connected in $G^{\sigma,k-1}$. Hence, all component reducing edges for agent i in $G^{\sigma,k-1}$ cost 1, i.e., $v_i^\sigma = 1$.□

Lemma 3.2.2 also proves that v^σ yields in a unique cost allocation independent of the mcst reached for simple mcst situations. Hence, for every simple mcst situation w, we can now denote by $v^\sigma(w)$ the unique cost allocation provided by voccp related to order $\sigma \in \Pi(N)$.

We show in the following proposition that v^σ equals $P^\sigma(w)$ for every mcst situation by making use of the linear decomposition of w into simple weight functions.

Proposition 3.2.5 *Let $(N, 0, w)$ be an mcst situation. Then v^σ equals $P^\sigma(w)$ for every $\sigma \in \Pi(N)$.*

Proof. Let $\pi \in \Pi(E_{N'})$ be such that $w \in K^\pi$. We know that

$$\begin{aligned} P^\sigma(w) &= w(\pi(1))P^\sigma(e^{\pi,1}) + \textstyle\sum_{k=2}^{|E_{N'}|} \left((w(\pi(k)) - w(\pi(k-1)))P^\sigma(e^{\pi,k})\right), \\ &= w(\pi(1))v^\sigma(e^{\pi,1}) + \textstyle\sum_{k=2}^{|E_{N'}|} \left((w(\pi(k)) - w(\pi(k-1)))v^\sigma(e^{\pi,k})\right), \end{aligned}$$

where the second equality follows by Lemma 3.2.2.

Now let Γ be a minimum cost spanning tree of w. Then by Theorem 3.2.2 Γ can be constructed by voccp related to order σ. Let v^σ be the cost allocation provided by voccp related to σ when it results in Γ.

It can be observed that Γ is also an mcst of every simple mcst situation $e^{\pi,k}$ ($k \in \{1, 2, ..., |E_{N'}|\}$). Then again by Theorem 3.2.2, Γ can be constructed by voocp related to σ in every simple mcst situation $e^{\pi,k}$. Now, consider voccp related to σ in w and in a simple mcst situation $e^{\pi,k}$. Assume that voccp resulted in Γ in both mcst situations. Then, obviously, for every iteration of the procedure $k \in \{1, ..., n\}$, the corresponding agent $\sigma(k)$ chooses to construct the same edge of Γ in both mcst situations w and $e^{\pi,k}$. But then voccp assigns the cost of the same edge to every agent in both situations. Since this observation is true for every simple mcst situation $e^{\pi,k}$, we can conclude that

$$v^\sigma = w(\pi(1))v^\sigma(e^{\pi,1}) + \sum_{k=2}^{|E_{N'}|} \Big((w(\pi(k)) - w(\pi(k-1)))v^\sigma(e^{\pi,k})\Big).$$

□

Since v^σ yields in a unique cost allocation independent of the mcst reached for any mcst situations, we can now denote by $v^\sigma(w)$ the unique cost allocation provided by voccp related to order $\sigma \in \Pi(N)$ for every mcst situation $w \in \mathcal{W}^N$.

We know by Proposition 3.2.2 that the average of P^σ-rules over all possible orders of agents is equal to ERO. But, since $v^\sigma(w) = P^\sigma(w)$ by Proposition 3.2.5, we can now conclude that ERO can be obtained an an average of the cost allocations provided by voccp over all possible orders of agents.

Theorem 3.2.3 *Let $w \in \mathcal{W}^N$. Then*

$$ERO(w) = \sum_{\sigma \in \Pi(N)} \frac{v^\sigma(w)}{n!}.$$

3.2.5 Voccp and the optimistic game in mcst situations

In this section, we investigate the connections between voccp and the optimistic transferable utility game for mcst problems (cf. Bergantiños and Vidal-Puga, 2007b). In an optimistic transferable utility game for mcst problems, the worth of a coalition is defined as the cost of connection, assuming that the rest of agents are already connected to the source. Bergantiños and Vidal-Puga (2005b) show that the Shapley value of this game is equal to the cost allocation provided by ERO. It is well-known that the Shapley value of a TU-game is equal to the average of its marginal vectors over all orders on the set of players. We will prove in the following that for every ordering σ of the agents, the v^σ value is equal to the marginal of the game for the same ordering.

The optimistic game for mcst situations is defined as follows. Given an mcst situation $(N, 0, w)$ and a nonempty coalition of agents $S \subset N$, we first obtain an optimistic mcst situation for S assuming that when the agents in S have to be connected to the source, the agents in $N \backslash S$ are already connected to the source and the agents in S can connect to the source via agents in $N \backslash S$. Formally, *the optimistic mcst situation for S* is the mcst situation $(S, 0, w_S)$, where $w_S(e) = w(e)$ for every $e \in E_S$ and $w_S(\{i, 0\}) = \min_{j \in N' \backslash S} w(\{i, j\})$ for every $i \in S$. That is the costs of edges that connect agents in S are not changed in the optimistic mcst situation. However, the cost of edges which connect an agent in S with the source are changed to be equal to the cost of the cheapest edge which connects the agent in S with an agent outside S or with the source in according with the assumptions of the optimistic setup.

Now the *optimistic TU-game* (N, v) associated with the mcst situation $(N, 0, w)$ is defined as $v(S) = w_S(\Gamma)$ where Γ is a mcst for $(S, 0, w_S)$ for every $S \subset N$ with $v(\emptyset) = 0$.

Example 3.2.4 Consider the mcst situation $(N, 0, w)$ with $N = \{1, 2, 3\}$ and w as depicted in Figure 3.1. Consider coalition $\{1, 2\}$. In Figure 3.3 we provide the optimistic mcst situation for $\{1, 2\}$ and a related mcst. Hence the worth of coalition $\{1, 2\}$ in the optimistic TU-game associated with $(N, 0, w)$ is 100. The complete

Figure 3.3: The optimistic mcst situation for $\{1, 2\}$ (left side) and a related mcst (right side)

optimistic TU-game associated with $(N, 0, w)$ is given in Table 3.4.

S	{1}	{2}	{3}	{1,2}	{1,3}	{2,3}	N
$v(S)$	40	40	60	100	100	100	190

Table 3.4: The optimistic TU-game in Example 3.2.4

$\diamond$

We first consider in Lemma 3.2.3 the structure of graphs $G^{\sigma,0}, G^{\sigma,1}, ..., G^{\sigma,n}$ that are formed during step 3 of voccp. In particular, we focus on the tree to which the agent that is going to construct an edge belongs and show that (i) the source 0 can not belong to this component and (ii) all other agents in the component must have constructed an edge in the previous iterations of the procedure.

Lemma 3.2.3 *Let $(N, 0, w)$ be an mcst, $\sigma \in \Pi(N)$, $k \in \{1, 2, ..., n\}$ and $\sigma(k) = i$. Then,*

(i) $0 \notin G_i^{\sigma,k-1}$.

(ii) $\sigma^{-1}(j) \leq \sigma^{-1}(i)$ *for every* $j \in G_i^{\sigma,k-1}$.

Proof. (i) Suppose $0 \in G_i^{\sigma,k-1}$. We know that $G_i^{\sigma,k-1}$ is a tree and hence the number of vertices of $G_i^{\sigma,k-1}$ is one more than its number of edges. Then, the number of agents that belong to this tree is equal to the number of edges of this tree and hence all agents that belong to this tree must have constructed an edge, a contradiction since agent i has not constructed an edge yet.
(ii) can be proven similarly. $\square$

In the following, we show that for every ordering of the agents, the cost allocation provided by the related voccp and the related marginal vector of the optimistic game are equal to each other.

Proposition 3.2.6 *Let $(N, 0, w)$ be an mcst. Then $v^\sigma(w) = m^\sigma(v)$ for every $\sigma \in \Pi(N)$.*

Proof. Let $\sigma \in \Pi(N)$. We denote by S_k the set of first k agents with respect to order σ, i.e., $S_k = \{\sigma(1), \sigma(2), ..., \sigma(k)\}$ for every $k \in \{1, 2, ..., n\}$. Also we denote an mcst of the optimistic mcst situation for S_k by Γ_k. We will show that the total cost of the first k edges constructed in voccp related to σ is equal to the cost of an mcst of $(S_k, 0, w_{S_k})$, i.e., $w(G^{\sigma,k}) = w_{S_k}(\Gamma_k)$ for every $k \in \{1, 2, ..., n\}$.

Pick $k \in \{1, 2, ..., n\}$. First, we know by Lemma 3.2.3 that at least one end vertex of every edge $e \in G^{\sigma,k}$ belongs to S_k. Second, if $\{u, v\} \in G^{\sigma,k}$ is such that $u \in S_k$ and $v \notin S_k$, then $\{u, v\}$ is the cheapest edge which connects u with a vertex outside S_k, i.e., $w(\{u, v\}) = \min_{w \in N' \setminus S_k} w(\{u, w\})$. Hence, $w(\{u, v\}) = w_{S_k}(\{u, 0\})$, i.e., the cost of $\{u, v\}$ equals to the cost of $\{u, 0\}$ the edge which connects u with the source in $(S_k, 0, w_{S_k})$. Now, we construct an mcst for $(S_k, 0, w_{S_k})$ by using the edges in $G^{\sigma,k}$ as follows: For every $\{u, v\} \in G^{\sigma,k}$, if $\{u, v\} \subset S_k$ then construct $\{u, v\}$ in $(S_k, 0, w_{S_k})$; if $u \in S_k$ and $v \notin S_k$ then construct $\{u, 0\}$ in $(S_k, 0, w_{S_k})$. Let us denote the set of edges we constructed in the mcst situation $(S_k, 0, w_{S_k})$ by Γ. Observe first that $w_{S_k}(\Gamma) = w(G^{\sigma,k})$. Observe second that Γ is a spanning tree because (i) it has k edges and (ii) it does not contain cycles since $G^{\sigma,k}$ does not contain cycles and every component of $G^{\sigma,k}$ contains exactly only one vertex outside S_k. Observe lastly that by construction of $G^{\sigma,k}$ by voccp Γ is an mcst of $(S_k, 0, w_{S_k})$.

So, we have that

$$\begin{aligned} m^\sigma(v)_{\sigma(k)} &= v(S_k) - v(S_{k-1}) = w_{S_k}(\Gamma_k) - w_{S_{k-1}}(\Gamma_{k-1}) = w(G^{\sigma,k}) - w(G^{\sigma,k-1}) \\ &= w(e_k^\sigma) = v^\sigma(w)_{\sigma(k)}, \end{aligned}$$

for every $k \in \{1, 2, ..., n\}$. Hence, we can conclude that $m^\sigma(v) = v^\sigma(w)$. □

3.3 ERO and Voccp for Minimum Cost Spanning Forest Situations

In this section we investigate extensions of the results obtained in Section 3.2 for mcst situations on minimum cost spanning forest situations (cf. Rosenthal, 1987). We first show that both Kruskal's algorithm and voccp can be defined for minimum cost spanning forest situations in a way that they yield efficient algorithms. Second, we extend the definition of ERO to this multisource situation and prove that ERO can again be obtained as the average of the cost allocations provided by voccp.

In an mcsf situation, there are finitely many identical sources and agents have to be connected to only one of them. These situations can be represented by a tuple (N, S, w) where $N = \{1, ..., n\}$ is the agent set, $S = \{0_1, 0_2, ..., 0_s\}$ ($s \in \mathbb{N}_{++}$) is the set of available identical sources. Let us denote $N \cup S$ by N'. Then $w : E_{N'} \to \mathbb{R}_{++}$ is a weight function on $E_{N'}$. Given an mcsf situation (N, S, w), a *spanning forest* of the graph $(N', E_{N'})$ is a set of trees that spans all vertices in N', in which each tree contains at least one source. A *minimum cost spanning forest* (mcsf) is a spanning forest with the minimum sum of weights of edges. Similar to the mcst situations with a unique source, mcsf situations can also be identified with their weight functions. Hence, we will denote the set of mcsf situations with agent set N by ${\mathcal{W}_{\mathcal{F}}}^N = \mathbb{R}_{++}^{E_{N'}}$.

In an mcsf situation, the availability of multiple identical sources enables agents to form several components which are connected to different sources. Hence, the optimal network in an mcsf situation is an mcsf. Moreover, since we consider mcsf situations where all edges have strictly positive weights, each tree in an mcsf contains exactly one source.

Kruskal's algorithm for mcst situations may not result in an optimal network if it is applied directly to mcsf situation. However, it can still be used to obtain an mcsf by applying it to a related mcst situation as follows. Let (N, S, w) be an mcsf situation. Now, add an additional vertex, denoted by 0, to N'; add edges $\{0, i\}$ for every $i \in V$ to $E_{N'}$. Also, define the weight function w_0 on $E_{N' \cup \{0\}}$ by

$$w_0(e) = \begin{cases} w(e), & \text{if } e \in E_{N'}, \\ 0, & \text{if } e = \{0, s\} \text{ for some } s \in S, \\ 2\max\{w(e) | e \in E_{N'}\}, & \text{if } e = \{0, i\} \text{ for some } i \in N. \end{cases}$$

We call the mcst situation $(N', 0, w_0)$ *the mcst situation associated with* (N, S, w). Since all edges in $E_{N'}$ have strictly positive costs, in any mcst of $(N', 0, w_0)$, the sources $0_1, ..., 0_s$ are connected to 0 through edges $\{0_k, 0\}$ that cost zero. Obviously if one removes these edges from an mcst of $(N', 0, w_0)$, then the graph obtained will be an mcsf for (N, S, w). Then one can obtain an mcsf for an mcsf situation, first by finding an mcst for the associated mcst situation and then by removing the edges that connect the sources with 0.

Example 3.3.1 Consider the mcsf situation (N, S, w) with $N = \{1, 2\}$, $S = \{0_1, 0_2\}$ and w as depicted in Figure 3.4. The associated mcst situation $(N', 0, w_0)$ is also depicted in Figure 3.4. By applying Kruskal's algorithm to $(N', 0, w_0)$, one obtains

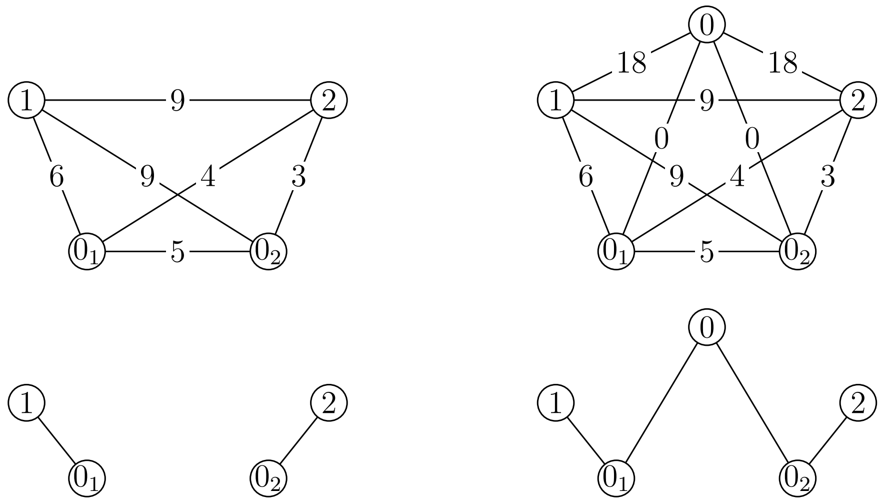

Figure 3.4: An mcsf situation with two agents and two sources (top-left); the associated mcst situation (top-right); the mcst Γ (bottom right) and the mcsf F (bottom left) considered in Example 3.3.1

the mcst Γ with $E(\Gamma) = \{\{0, 0_1\}, \{0, 0_2\}, \{1, 0_1\}, \{2, 0_2\}\}$. Then, by removing edges $\{0, 0_1\}$ and $\{0, 0_2\}$ from Γ, one obtains the mcsf F with $E(F) = \{\{1, 0_1\}, \{2, 0_2\}\}$. The mcst Γ and the mcsf F are also depicted in Figure 3.4. $\diamond$

Now we can define the equal remaining obligations rule for mcsf situations by making use of the cost allocations assigned by ERO to the associated mcst situations as follows. Let (N, S, w) be an mcsf situation and consider $ERO(w_0)$ the cost allocation assigned by ERO to the associated mcst situation $(N', 0, w_0)$. Observe that sources $0_1, ..., 0_s$ are treated as agents in the associated mcst situation. Hence, $ERO(w_0)$ also assigns costs to sources for the construction of the mcst. However, since sources are connected to 0 with edges that cost zero, $ERO(w_0)$ does not assign any cost to the sources, i.e, $ERO(w_0)_s = 0$ for every source $s \in S$. In particular, the cost of the mcst in the associated mcst situation is allocated completely to the agents.

Definition 3.3.1 *The equal remaining obligations rule for mcsf situations* $ERO_{\mathcal{F}}$ *is defined as the map* $ERO_{\mathcal{F}} : \mathcal{W_F}^N \to \mathbb{R}^N$*, where*

$$ERO_{\mathcal{F}}(w)_i = ERO(w_0)_i,$$

for every $i \in N$ *and* $w \in \mathcal{W_F}^N$*.*

Example 3.3.2 Consider the mcsf situation (N, S, w) with $N = \{1, 2\}$, $S = \{0_1, 0_2\}$ and w as depicted in Figure 3.4.

In order to calculate $ERO_{\mathcal{F}}(w)$, we simply have to calculate ERO for the related mcst situation $(N', 0, w_0)$. Observe that $w_0 \in K^\pi$, with $\pi(1) = \{0, 0_1\}$, $\pi(2) = \{0, 0_2\}$, $\pi(3) = \{2, 0_2\}$, $\pi(4) = \{0_1, 2\}$, $\pi(5) = \{0_1, 0_2\}$, $\pi(6) = \{1, 0_1\}$, $\pi(7) = \{1, 2\}$, $\pi(8) = \{0_2, 1\}$, $\pi(9) = \{0, 1\}$ and $\pi(10) = \{0, 2\}$. The sequence of the graphs $F^{\pi,k}$ formed by Kruskal's algorithm when it follows the order π and the corresponding connection vectors, where the first two indices correspond to 0_1 and 0_2 and the third (fourth) index correspond agent 1 (2), are given in Table 3.5 below. Hence, the

$F^{\pi,k}$	$b^{\pi,k}$
$\emptyset$	$(1,1,1,1)^t$
$\{\{0_1,0\}\}$	$(0,1,1,1)^t$
$\{\{0_1,0\},\{0_2,0\}\}$	$(0,0,1,1)^t$
$\{\{0_1,0\},\{0_2,0\},\{0_2,2\}\}$	$(0,0,1,0)^t$
$\{\{0_1,0\},\{0_2,0\},\{0_2,2\},\{0_1,2\}\}$	$(0,0,1,0)^t$
$\{\{0_1,0\},\{0_2,0\},\{0_2,2\},\{0_1,2\},\{0_1,0_2\}\}$	$(0,0,1,0)^t$
$\{\{0_1,0\},\{0_2,0\},\{0_2,2\},\{0_1,2\},\{0_1,0_2\},\{0_1,1\}\}$	$(0,0,0,0)^t$
$\vdots$	$\vdots$
$E_{N'\cup\{0\}}$	$(0,0,0,0)^t$

Table 3.5: Graphs formed in each step of Kruskal's algorithm and the corresponding connection vectors in Example 3.3.2

contribution matrix M^π is given by

$$M^\pi = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Observe that $w_0^\pi = (0, 0, 3, 4, 5, 6, 9, 9, 18, 18)$. Hence, $ERO(w_0) = M^\pi w_0^\pi = (0, 0, 6, 3)^t$ and $ERO_{\mathcal{F}}(w) = (6, 3)^t$. $\diamond$

In the following, we first define the vertex oriented construct and charge procedure for mcsf situations. Then we show that voccp is an efficient algorithm for mcsf situations. Finally, we prove that ERO for mcsf situations can be obtained as an average of the cost allocations provided by voccp over all orders of agents.

Voccp for mcsf situations is defined formally as follows:

Let (N, S, w) be an mcsf situation.
(**Step 1**) Pick $\sigma \in \Pi(N)$.
(**Step 2**) Set $G^{\sigma,0} = (N', \emptyset)$ and $v_{\mathcal{F}}^{\sigma,0} = 0$ ($\in \mathbb{R}^N$).
(**Step 3**) For $k = 1$ to n:

Let $\sigma(k) = i$. Choose an edge e_k^σ such that

$$e_k^\sigma = arg\min\{w(e)|e \in \mathcal{A}_i(G^{\sigma,k-1})\}.$$

Set $G^{\sigma,k} = G^{\sigma,k-1} + e_k^\sigma$ and $v_{\mathcal{F}}^{\sigma,k} = v_{\mathcal{F}}^{\sigma,k-1} + w(e_k^\sigma)e^{\{i\}}$.

(**Step 4**) Set $F^\sigma = G^{\sigma,n}$ and $v_{\mathcal{F}}^\sigma = v_{\mathcal{F}}^{\sigma,n}$.

Example 3.3.3 Consider the mcsf situation (N, S, w) with $N = \{1, 2\}$, $S = \{0_1, 0_2\}$ and w as depicted in Figure 3.4.

Let $\sigma \in \Pi(N)$ be such that $\sigma(i) = i$ for every $i \in N$. The related voccp is described as follows:
(**Step 1**) Let $\sigma \in \Pi(N)$ be such that $\sigma(i) = i$ for every $i \in N$.
(**Step 2**) Set $G^{\sigma,0} = (N', \emptyset)$ and $v_{\mathcal{F}}^{\sigma,0} = 0$.
(**Step 3**) Step 3 consists of the following two iterations:

- $k = 1$: $\sigma(1) = 1$. Obviously each edge incident with 1 is allowed for 1 in $G^{\sigma,0}$. Then, $e_1^\sigma = \{1, 0_1\}$; $G^{\sigma,1} = (N', \{\{1, 0_1\}\})$ and $v_{\mathcal{F}}^{\sigma,k} = (6, 0)$.
- $k = 2$: $\sigma(2) = 2$. Since 2 is not connected with any other vertex in $G^{\sigma,1}$, the allowed edges for 2 in $G^{\sigma,1}$ are the ones that are incident with 2. Then, $e_2 = \{2, 0_2\}$. $G^{\sigma,2} = (N', \{\{1, 0_1\}, \{2, 0_2\}\})$ and $v_{\mathcal{F}}^{\sigma,k} = (6, 3)$.

(**Step 4**) $F^\sigma = (N', \{\{1, 0_1\}, \{2, 0_2\}\})$ and $v_{\mathcal{F}}^\sigma = (6, 3)$.

Observe lastly that voccp related to the reverse order results in the same construct and charge result. $\diamond$

Obviously F^σ as obtained in Step 4 of voccp related to order σ does not contain any cycles and hence is a forest. It is well known that a forest with p vertices and k edges has $p - k$ components. So, since F^σ has n edges and $n + s$ vertices, it must contain s components which are trees. Recall here that s is the number of sources in an mcsf situation. One can also observe that each of these components must contain exactly one source. Hence, F^σ is a spanning forest. Similar to the mcst case, voccp for mcsf situations can lead to different spanning forests depending on choices of agents.

In the following theorem we first show that voccp is an efficient algorithm for mcsf problems, i.e., any spanning forest provided by voccp is an mcsf of the mcsf situation. Second, we show that the cost assigned by voccp to an agent in the mcsf situation is equal to the cost assigned by voccp to the same agent in the associated mcst situation. Let (N, S, w) be an mcsf situation. We say that an order $\pi \in \Pi(N')$ is *compatible* with the order $\sigma \in \Pi(N)$ if it satisfies the condition that $\sigma(i) < \sigma(j)$ if and only if $\pi(i) < \pi(j)$ for every $i, j \in N$.

Theorem 3.3.1 *Let (N, S, w) be an mcsf situation, $(N', 0, w_0)$ be the associated mcst situation and $\pi \in \Pi(N)$. Then*

(i) F^σ is an mcsf of (N, S, w) for every $\sigma \in \Pi(N)$;

(ii) $v^\sigma_{\mathcal{F},i} = v^\pi(w_0)_i$ for every order $\sigma \in \Pi(N)$, every $\pi \in \Pi(N')$ that is compatible with σ and every $i \in N$.

Proof. First recall that the sources $0_1, ..., 0_s$ are treated as agents in the associated mcst situation $(N', 0, w_0)$. Let $\sigma \in \Pi(N)$ and let $\pi \in \Pi(N')$ be compatible with σ. Consider the voccp related to order σ in the mcsf situation and the voccp related to π in the associated mcst situation. Observe that during voccp related to π in the associated mcst situation $(N', 0, w_0)$

- Each source $s \in S$ selects the edge $\{s, 0\}$ because $\{s, 0\}$ is always an allowed edge for s, it costs zero and all other allowed edges have strictly positive costs.
- None of the agents selects an edge $\{i, 0\}$ $(i \in N)$ since these are the most expensive edges with respect to w_0.

That's why in both procedures, voccp related to σ in mcsf situation and voccp related to π in the associated mcst situation, each agent $i \in N$ is in the same situation: They have the same options and hence they can select to construct the same edge in both procedures. And in this case the graph found by the voccp related to σ in the mcsf situation can also be obtained by first using voccp related to π to obtain an mcst for the associated mcst situation and then removing the arcs $(0, s)$, for every $s \in S$. So, voccp is an efficient algorithm for mcsf situations and moreover, since agents can select the same edges in both procedures $v^\sigma_{\mathcal{F},i} = v^\pi(w_0)_i$ for every $i \in N$. □

Theorem 3.3.1 also proves that the cost allocation provided by voccp for the mcsf situations is unique, i.e., is independent of the mcsf reached. Hence, we can now denote the cost allocation provided by voccp related to σ by $v^\sigma_{\mathcal{F}}(w)$.

Finally, we prove that $ERO_{\mathcal{F}}$ can be obtained as the average of the cost allocations provided by voccp for every order of the agents.

Theorem 3.3.2 *Let $w \in \mathcal{W_F}^N$. Then*

$$ERO_{\mathcal{F}}(w) = \sum_{\sigma \in \Pi(N)} \frac{v^\sigma_{\mathcal{F}}(w)}{n!}.$$

Proof. We know that $ERO_{\mathcal{F}}(w)_i = ERO_i(w_0)$. We also know by Theorem 3.3.1 that $v^{\sigma}_{\mathcal{F}}(w)_i = v^{\pi}(w_0)_i$ for every order $\sigma \in \Pi(N)$, $\pi \in \Pi(N)$ that is compatible with σ and $i \in N$. Observe that for every order $\sigma \in \Pi(N)$, there exists $\frac{(n+s)!}{n!}$ compatible orders in $\Pi(N')$. Then

$$
\begin{aligned}
ERO_{\mathcal{F}}(w)_i = ERO(w_0)_i &= \sum_{\pi \in \Pi(N')} \frac{v^{\pi}(w_0)_i}{(n+s)!} \\
&= \sum_{\sigma \in \Pi(N)} \frac{\frac{(n+s)!}{n!} v^{\sigma}_{\mathcal{F}}(w)_i}{(n+s)!} = \sum_{\sigma \in \Pi(N)} \frac{v^{\sigma}_{\mathcal{F}}(w)_i}{n!},
\end{aligned}
$$

for every $i \in N$. Hence, we can conclude that $ERO_{\mathcal{F}}(w)$ can be obtained as the average of the cost allocations provided by voccp over all orders of agents. □

3.4 ERO and Voccp for Mcst Situations with Two Sources

In this section we investigate the extensions of the results obtained in Section 3.2 for mcst situations in which the agents have to be connected to two sources. Section 3.4.1 proposes an equal remaining obligations rule for mcst situations with two sources. Then in Section 3.4.2 we extend voccp to these multisource situations in a way that it yields an efficient algorithm and prove that ERO can again be obtained as the average of the cost allocations provided by voccp. In Section 3.4.3 we define an optimistic transferable utility game for mcst problems with two sources and show that for every ordering of the agents, the cost allocation provided by the related voccp is equal to the marginal of the optimistic game for the same ordering.

In this section we consider mcst situations with two sources. In this type of mcst situations there exists a group of agents $N = \{1, 2, ..., n\}$ that has to be connected to two sources $\{0_1, 0_2\}$. These situations can be represented by a tuple $(N, \{0_1, 0_2\}, w)$. Let us denote $\{0_1, 0_2\} \cup N$ by N'. Here, as before $w : E_{N'} \to \mathbb{R}_{++}$ is a weight function on $E_{N'}$. Similar to mcst situations with a unique source, mcst situations with two sources can also be identified with their weight functions. Hence, we will denote the set of mcst situations with two sources with agent set N by $\mathcal{W_T}^N = \mathbb{R}^{E_{N'}}_{++}$.

The optimal network in an mcst situation with two sources is obviously an mcst of $(N', E_{N'})$. Hence, Kruskal's algorithm is an efficient algorithm for these multisource mcst situations. It can also easily be observed that any spanning tree of $(N', E_{N'})$ has $n + 1$ edges. That is the agents have to construct $n + 1$ edges in order to be connected to both sources.

Example 3.4.1 Consider the mcst situation with two sources $(N, \{0_1, 0_2\}, w)$ with $N = \{1, 2, 3\}$ and w as depicted in Figure 3.5.

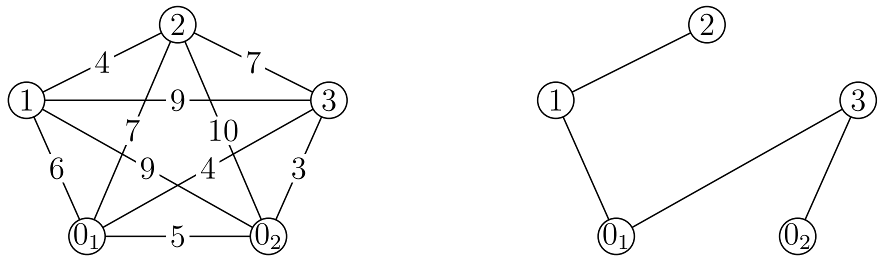

Figure 3.5: An mcst situation with two sources (left side) and a related mcst

Kruskal's algorithm first constructs the cheapest edge $\{0_2, 3\}$. The algorithm continues by constructing both of the two second cheapest edges $\{0_1, 3\}$ and $\{1, 2\}$ since neither of these edges creates a cycle with the previously constructed edges. Next the algorithm selects the edge $\{0_1, 0_2\}$ but it will not construct this edge since it creates a cycle with the previously constructed edges $\{0_2, 3\}$ and $\{0_1, 3\}$. Finally, the algorithm selects and constructs $\{0_1, 1\}$ resulting in the mcst with the set of edges $\{\{0_1, 1\}, \{1, 2\}, \{0_1, 3\}, \{0_2, 3\}\}$. $\diamond$

3.4.1 ERO for cost sharing in mcst situations with two sources

In the following we will propose an equal remaining obligations rule for mcst situations with two sources. In order to define this rule, we will follow a set-up analogous to the set-up used for *ERO* in Section 3.2.1.

Let $\Pi(E_{N'})$, K^π and w^π ($\pi \in \Pi(E_{N'})$) be defined analogously to their counterparts in Section 3.2.1. Let $(N, \{0_1, 0_2\}, w)$ be an mcst situation with two sources and $\pi \in \Pi(E_{N'})$ be such that $w \in K^\pi$. Let us first define a rule ERO_T^π on cone K^π. Similar to ERO's original definition in Section 3.2.1, we will again start with considering a sequence of $|E_{N'}| + 1$ graphs that are formed during Kruskal's algorithm when the algorithm follows the order π on the set of edges $E_{N'}$: $F^{\pi,0}, F^{\pi,1}, ..., F^{\pi,|E_{N'}|}$. But, this time we will define two different connection vectors and accordingly two connection matrices as follows. Connection vectors $b_1^{\pi,k} \in \mathbb{R}^N$ and $b_2^{\pi,k} \in \mathbb{R}^N$ are defined for each $k \in \{0, 1, ..., |E_{N'}|\}$ as

$$b_{1,i}^{\pi,k} = \begin{cases} 0 & \text{if } i \text{ is connected with } 0_1 \text{ or with } 0_2 \text{ in } F^{\pi,k} \\ \frac{1}{n_i(F^{\pi,k})} & \text{otherwise} \end{cases} \tag{3.8}$$

$$b_{2,i}^{\pi,k} = \begin{cases} 0 & \text{if } 0_1 \text{ and } 0_2 \text{ are connected in } F^{\pi,k} \\ \frac{1}{n} & \text{otherwise} \end{cases} \tag{3.9}$$

for each $i \in N$ and $w \in \mathcal{W}_{\mathcal{T}}{}^{N}$.

The contribution matrices with respect to $\pi \in \Pi(E_{N'})$ $M_1^\pi \in \mathbb{R}^{N \times E_{N'}}$ and $M_2^\pi \in \mathbb{R}^{N \times E_{N'}}$ are defined as follows. M_1^π is the matrix which lists the change in the connectivity of the agents as given by $b_1^{\pi,k}$, i.e., the k-th column of M_1^π equals

$$M_1^\pi e^k = (b_1^{\pi,k-1} - b_1^{\pi,k})$$

for each $k \in \{1, ..., |E_{N'}|\}$. Similarly, M_2^π lists the change in the connectivity of the agents as given by $b_2^{\pi,k}$, i.e., the k-th column of M_2^π equals

$$M_2^\pi e^k = (b_2^{\pi,k-1} - b_2^{\pi,k})$$

for each $k \in \{1, ..., |E_{N'}|\}$.

It can easily be observed that the connection vector $b_1^{\pi,k}$ and the corresponding contribution matrix M_1^π are straightforward extensions of their counterparts defined for ERO (see equations 3.2 and 3.3). Similar to the contribution matrix for ERO, each nonzero-column of M_1^π corresponds to an edge constructed by Kruskal's algorithm and represents the fraction of the cost of the edge paid by each player. However, in every mcst situation with two sources there exists one edge which is constructed by Kruskal's algorithm but the corresponding column in M_1^π is a zero-column. To see this, let $F^{\pi,k}$ be the first graph in the sequence $F^{\pi,0}, F^{\pi,1}, ..., F^{\pi,|E_{N'}|}$ in which 0_1 and 0_2 are connected. Observe that the corresponding edge $\pi(k)$ joins two separate components in $F^{\pi,k-1}$ one including 0_1 and the other including 0_2. Hence, $\pi(k)$ is constructed by Kruskal's algorithm since it does not create a cycle with the previously constructed edges. However, the column $(M_1^\pi e^k)$ is a zero column since $b_{1,i}^{\pi,k} = b_{1,i}^{\pi,k-1} = 0$ for every agent i that belongs to the components of $F^{\pi,k-1}$ joined by edge $\pi(k)$. So, according to contribution matrix M_1^π agents do not contribute for an edge which is constructed. In order to be able to distribute the cost of edge $\pi(k)$ among agents, we introduced connection vectors $b_2^{\pi,t}$ and the corresponding contribution matrix M_2^π. Observe that $(M_2^\pi e^k)_i = \frac{1}{n}$ and every other column of matrix M_2^π is a zero column. That is M_2^π distributes the cost of edge $\pi(k)$ equally among agents.

We are now ready to define the $ERO_{\mathcal{T}}^\pi$-value on K^π.

Definition 3.4.1 *For each $\pi \in \Pi(E_{N'})$, $ERO_{\mathcal{T}}^\pi$ is defined as the map $ERO_{\mathcal{T}}^\pi : K^\pi \rightarrow \mathbb{R}^N$, where $ERO_{\mathcal{T}}^\pi(w) = M_1^\pi w^\pi + M_2^\pi w^\pi$ for each mcst situation with two sources w in the cone K^π.*

A variant of Lemma 3.2.1 holds also for rules $ERO_{\mathcal{T}}^\pi$. Hence, similarly to ERO^π-rules, for every order π that a weight function w is compatible with, $ERO_{\mathcal{T}}^\pi$ also results in the same allocation.

Proposition 3.4.1 *Let $(N, \{0_1, 0_2\}, w)$ be an mcst situation with two sources. Then, $M_1^\pi w^\pi = M_1^{\pi'} w^{\pi'}$, $M_2^\pi w^\pi = M_2^{\pi'} w^{\pi'}$ and hence $ERO_\mathcal{T}^\pi(w) = ERO_\mathcal{T}^{\pi'}(w)$ for every $\pi, \pi' \in \Pi(E_{N'})$ with $w \in K^\pi \cap K^{\pi'}$.*

This enables us to define a $ERO_\mathcal{T}$-rule over all mcst situations with two sources.

Definition 3.4.2 *The equal remaining obligations rule for mcst situations with two sources is defined as the map $ERO_\mathcal{T} : \mathcal{W_T}^N \to \mathbb{R}^N$, where*

$$ERO_\mathcal{T}(w) = ERO_\mathcal{T}^\pi(w) = M_1^\pi w^\pi + M_2^\pi w^\pi \tag{3.10}$$

for every $w \in \mathcal{W_T}^N$ and $\pi \in \Pi(E_{N'})$ such that $w \in K^\pi$.

Remark 3.4.1 The method we followed to extend ERO to mcst situations with two sources can be summarized as: First use M_1^π the straightforward extension of a contribution matrix and then distribute the cost of the edge that M_1^π failed to handle equally among the agents. In the following sections we consider straightforward extensions of voccp and the optimistic game associated with mcst situations and show that (i) $ERO_\mathcal{T}$ can be obtained as an average of the cost allocations provided by voccp over all orders of agents and (ii) the Shapley value of the extended optimistic game is equal to $ERO_\mathcal{T}$. We believe that these two results provide strong support to $ERO_\mathcal{T}$ as an appropriate extension of equal remaining obligations rule for mcst situations with two sources.

Example 3.4.2 Consider the mcst situation with two sources $(N, \{0_1, 0_2\}, w)$ with $N = \{1, 2, 3\}$ and w as depicted in Figure 3.5. Observe that $w \in K^\pi$, with $\pi(1) = \{0_2, 3\}$, $\pi(2) = \{0_1, 3\}$, $\pi(3) = \{1, 2\}$, $\pi(4) = \{0_1, 0_2\}$ $\pi(5) = \{0_1, 1\}$, $\pi(6) = \{0_1, 2\}$, $\pi(7) = \{2, 3\}$, $\pi(8) = \{1, 3\}$, $\pi(9) = \{0_2, 1\}$ and $\pi(10) = \{0_2, 2\}$.

The sequence of graphs $F^{\pi,k}$ formed by Kruskal's algorithm when it follows the order π on the set of edges $E_{N'}$ and corresponding connection vectors are given in Table 3.6.

Then the contribution matrices M_1^π and M_2^π are given by

$$M_1^\pi = \begin{pmatrix} 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

and

$$M_2^\pi = \begin{pmatrix} 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Observe lastly that $w^\pi = (3, 4, 4, 5, 6, 7, 7, 9, 9, 10)^t$. Hence, $ERO_\mathcal{T}(w) = M_1^\pi w^\pi + M_2^\pi w^\pi = (5 + \frac{4}{3}, 5 + \frac{4}{3}, 3 + \frac{4}{3})^t$. ◇

$F^{\pi,k}$	$b_1^{\pi,k}$	$b_2^{\pi,k}$
$\emptyset$	$(1,1,1)^t$,	$(\frac{1}{3},\frac{1}{3},\frac{1}{3})^t$
$\{\{0_2,3\}\}$	$(1,1,0)^t$,	$(\frac{1}{3},\frac{1}{3},\frac{1}{3})^t$
$\{\{0_2,3\},\{0_1,3\}\}$	$(1,1,0)^t$,	$(0,0,0)^t$
$\{\{0_2,3\},\{0_1,3\},\{1,2\}\}$	$(\frac{1}{2},\frac{1}{2},0)^t$,	$(0,0,0)^t$
$\{\{0_2,3\},\{0_1,3\},\{1,2\},\{0_1,0_2\}\}$	$(\frac{1}{2},\frac{1}{2},0)^t$,	$(0,0,0)^t$
$\{\{0_2,3\},\{0_1,3\},\{1,2\},\{0_1,0_2\},\{0_1,1\}\}$	$(0,0,0)^t$,	$(0,0,0)^t$
$\{\{0_2,3\},\{0_1,3\},\{1,2\},\{0_1,0_2\},\{0_1,1\},\{0_1,2\}\}$	$(0,0,0)^t$,	$(0,0,0)^t$
$\vdots$	$\vdots$	$\vdots$
$E_{N'}$	$(0,0,0)^t$,	$(0,0,0)^t$

Table 3.6: Graphs formed in each step of Kruskal's algorithm and corresponding connection vectors in Example 3.4.2

In the following we define for every mcst situation with two sources a related mcst situation and show that there is a close relationship between the equal remaining obligations rule of the mcst situation with two sources and the equal remaining obligations rule of the related mcst situation. Let $(N,\{0_1,0_2\},w)$ be an mcst situation with two sources and let $\pi \in \Pi(E_{N'})$ such that $w \in K^\pi$. Remove the sources 0_1, 0_2 and add vertex 0. Define the weight function $w_0 : E_{N\cup\{0\}\to\mathbb{R}+}$ by

$$w_0(e) = \begin{cases} w(e), & \text{if } e \in E_N, \\ min\{w(\{u,0_1\}), w(\{u,0_2\})\}, & \text{if } e = \{u,0\} \text{ for some } u \in N. \end{cases}$$

The mcst situation $(N,0,w_0)$ as described above is called the mcst situation associated with $(N,\{0_1,0_2\},w)$.

Example 3.4.3 Consider the mcst situation with two sources $(N,\{0_1,0_2\},w)$ with $N=\{1,2,3\}$ and w as depicted in Figure 3.5. The associated mcst situation $(N,0,w_0)$ is depicted in Figure 3.6. $\diamond$

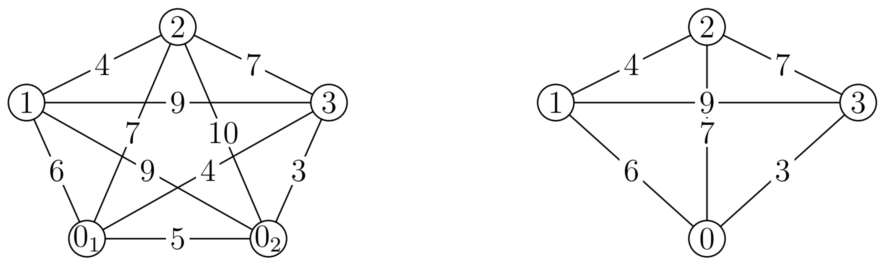

Figure 3.6: An mcst situation with two sources (left side) and the associated mcst situation

In the following lemma we show that the cost allocation provided by ERO for the associated mcst situation w_0 is equal to $M_1^\pi w^\pi$.

Lemma 3.4.1 *Let $w \in \mathcal{W}_T{}^N$ and $\pi \in \Pi(E_{N'})$ be such that $w \in K^\pi$. Then $M_1^\pi w^\pi = ERO(w_0)$.*

Proof. Let $\pi(t_1), \pi(t_2), ..., \pi(t_{n+1})$ be the $n+1$ edges of the mcst Γ constructed by Kruskal's algorithm corresponding to π in w. Let $\pi(t_k)$ be the first edge among them such that 0_1 and 0_2 are connected during Kruskal's algorithm. Then the edges in $\Gamma \backslash \{\pi(t_k)\} = \{\pi(t_1), ..., \pi(t_{k-1}), \pi(t_{k+1}), ..., \pi(t_{n+1})\}$ correspond to the non-zero columns of M_1^π.

Observe that the edge $\{0_1, 0_2\}$ and the more expensive of the edges $\{i, 0_1\}, \{i, 0_2\}$ $(i \in N)$ can be constructed by Kruskal's algorithm only if it is the first edge that 0_1 and 0_2 are connected. Then every edge $e \in \Gamma \backslash \{\pi(t_k)\}$ is either an edge that connects two agents or is the cheaper of the two edges that connect an agent with the source.

Let us now associate with each edge $\pi(t_r) \in \Gamma \backslash \{\pi(t_k)\}$ an edge $e_r \in E_{N \cup \{0\}}$. If $\pi(t_r) \subset N$, then $e_r = \pi(t_r)$; if $\pi(t_r)$ is the cheaper of the two edges that connect an agent i with a source, then $e_r = \{i, 0\}$. Obviously, $w(\pi(t_r)) = w_0(e_r)$. Now let $\Gamma' = \{e_1, ..., e_{k-1}, e_{k+1}, ... e_{n+1}\}$. Obviously, Γ' contains n edges and does not contain any cycles. Hence, Γ' is a spanning tree of $(N, E_{N \cup \{0\}})$. Moreover, it can easily be shown that Γ' is a mcst of the associated mcst situation w_0 by using the fact that Γ is a mcst for the mcst situation w.

Now, since Γ' is a mcst of the associated mcst situation w_0, there exists an order $\pi' \in \Pi(E_{N \cup \{0\}})$ such that $w_0 \in K^{\pi'}$ and such that the corresponding Kruskal's algorithm constructs first e_1, then e_2,... and lastly e_{n+1}. Clearly, $M^{\pi'}$ equals the matrix obtained from M_1^π by deleting its zero columns. Hence, $ERO(w_0) = M^{\pi'} w_0^{\pi'} = M_1^\pi w^\pi$. □

3.4.2 The vertex oriented construct and charge procedure for mcst situations with two sources

In this section we define voccp for mcst situations with two sources.

We know that since each agent has to be connected to both of the available sources, the mcst in an mcst situation with two sources has $n+1$ edges. Then the first n iterations of voccp for such an mcst problem will yield a forest which is composed of two disjoint trees and hence one more edge has to be constructed to connect these two disjoint trees. Therefore, we modify the voccp so that for any order of the agents $\sigma \in \Pi(N)$, after the first n iterations, the last agent in the order, $\sigma(n)$ is also responsible for the construction of the $(n+1)^{th}$ edge which connects the tree that he is involved in to the other one. In the following, abusing notation, $\sigma(n)$ the last agent in the order will also be denoted by $\sigma(n+1)$ for any $\sigma \in \Pi(N)$.

Voccp for mcst situations with two sources is defined formally as follows:

Let $(N, \{0_1, 0_2\}, w)$ be an mcst situation with two sources.

(**Step 1**) Pick $\sigma \in \Pi(N)$.

(**Step 2**) Set $G^{\sigma,0} = (N', \emptyset)$ and $v_T^{\sigma,0} = 0$ $(\in \mathbb{R}^N)$.

(**Step 3**) For $k = 1$ to $n+1$:

Let $\sigma(k) = i$. Choose an edge e_k^σ such that

$$e_k^\sigma = arg\min\{w(e)|e \in \mathcal{A}_i(G^{\sigma,k-1})\}.$$

Set $G^{\sigma,k} = G^{\sigma,k-1} + e_k^\sigma$ and $v_T^{\sigma,k} = v_T^{\sigma,k-1} + w(e_k^\sigma)e^{\{i\}}$.

(**Step 4**) Set $\Gamma^\sigma = G^{\sigma,n+1}$ and $v_T^\sigma = v^{\sigma,n+1}$.

We illustrate voccp in mcst situations with two sources in the following example.

Example 3.4.4 Consider the mcst situation with two sources $(N, \{0_1, 0_2\}, w)$ with $N = \{1, 2, 3\}$ and w as depicted in Figure 3.5. Let $\sigma \in \Pi(N)$ be such that $\sigma(i) = i$ for every $i \in N$. The related voccp is described as follows:

(**Step 1**) Let $\sigma \in \Pi(N)$ be such that $\sigma(i) = i$ for every $i \in N$.

(**Step 2**) Set $G^{\sigma,0} = (N', \emptyset)$ and $v^{\sigma,0} = 0$.

(**Step 3**) Step 3 consists of the following four iterations:

- $k = 1$: $\sigma(1) = 1$. Obviously $\mathcal{A}_1(G^{\sigma,0}) = \{\{1, 0_1\}, \{1, 0_2\}, \{1, 2\}, \{1, 3\}\}$. Then, $e_1^\sigma = \{1, 2\}$, $G^{\sigma,1} = (N', \{\{1, 2\}\})$ and $v^{\sigma,1} = (4, 0, 0)$.

- $k = 2$: $\sigma(2) = 2$. Note that 2 and 1 are in the same component of $G^{\sigma,1}$ but 3, 0_1 and 0_2 belong to different components. So,

$$\mathcal{A}_2(G^{\sigma,1}) = \{\{1, 3\}, \{1, 0_1\}, \{1, 0_2\}, \{2, 3\}, \{2, 0_1\}, \{2, 0_2\}\}.$$

  Obviously $e_2^\sigma = \{1, 0_1\}$, $G^{\sigma,2} = (N', \{\{1, 2\}, \{1, 0_1\}\})$ and $v^{\sigma,1} = (4, 6, 0)$.

- $k = 3$: $\sigma(3) = 3$. Note that 1, 2 and 0_1 are in the same component of $G^{\sigma,1}$ but 3 and 0_2 belong to different components. Obviously

$$\mathcal{A}_3(G^{\sigma,2}) = \{\{3, 1\}, \{3, 2\}, \{3, 0_1\}, \{3, 0_2\}\}.$$

  Then, $e_3^\sigma = \{3, 0_2\}$, $G^{\sigma,3} = (N', \{\{1, 2\}, \{1, 0_1\}, \{3, 0_2\}\})$ and $v^{\sigma,3} = (4, 6, 3)$.

- $k = 4$: $\sigma(4) = 3$. Note that there are two components in $G^{\sigma,3}$: 1, 2 and 0_1 are in one component and 3 and 0_2 belong to the other component. Then,

$$\mathcal{A}_3(G^{\sigma,3}) = \{\{3,1\},\{3,2\},\{3,0_1\},\{0_2,1\},\{0_2,2\},\{0_2,0_1\}\}.$$

So, $e_4^\sigma = \{0_1,3\}$, $G^{\sigma,4} = (N',\{\{1,2\},\{1,0_1\},\{3,0_2\},\{0_1,3\}\})$ and $v^{\sigma,3} = (4,6,3+4)$.

(**Step 4**) $\Gamma^\sigma = (N',\{\{1,2\},\{1,0_1\},\{3,0_2\},\{0_1,3\}\})$ and $v^\sigma = (8,6,3)$. In Table 3.7 we give the construct and charge results provided by voccp for all orderings of agents. ◇

	Constructed edges by			Costs for		
σ	1	2	3	1	2	3
(1,2,3)	$\{1,2\}$	$\{1,0_1\}$	$\{3,0_2\}$,$\{0_1,3\}$	4	6	7
(1,3,2)	$\{1,2\}$	$\{1,0_1\}$,$\{0_1,3\}$	$\{3,0_2\}$	4	10	3
(2,1,3)	$\{1,0_1\}$	$\{1,2\}$	$\{3,0_2\}$,$\{0_1,3\}$	6	4	7
(2,3,1)	$\{1,0_1\}$,$\{0_1,3\}$	$\{1,2\}$	$\{3,0_2\}$	10	4	3
(3,1,2)	$\{1,2\}$	$\{1,0_1\}$,$\{0_1,3\}$	$\{3,0_2\}$	4	10	3
(3,2,1)	$\{1,0_1\}$,$\{0_1,3\}$	$\{1,2\}$	$\{3,0_2\}$	10	4	3

Table 3.7: Construct and charge results for the mcst situation in Figure 3.5

We establish in the following theorem that (i) voccp is an efficient algorithm for mcst situations with two sources and (ii) that every mcst in an mcst situation with two sources can be obtained by voccp for any order of agents followed.

Theorem 3.4.1 *Let $(N,\{0_1,0_2\},w)$ be an mcst situation with two sources and $\sigma \in \Pi(N)$. Then,*
(i) Γ^σ is an mcst of $(N,\{0_1,0_2\},w)$.
(ii)If T is an mcst of $(N,\{0_1,0_2\},w)$, then T can be the result of a vertex oriented construct and charge procedure for any permutation $\sigma \in \Pi(N)$.

We omit the proof of Theorem 3.4.1 since it is similar to the proofs of theorems 3.2.1 and 3.2.2.

We consider in Lemma 3.2.3 the structure of graphs $G^{\sigma,0}, G^{\sigma,1}, ..., G^{\sigma,n+1}$ that are formed during step 3 of voccp and prove one preliminary result.

Lemma 3.4.2 *Let $(N,\{0_1,0_2\},w)$ be an mcst situation with two sources, $\sigma \in \Pi(N)$, $k \in \{1,2,...,n\}$ and $\sigma(k) = i$. Then, 0_1 and 0_2 can not belong to $G_i^{\sigma,k-1}$.*

The proof Lemma 3.4.2 is similar to its counterpart in Lemma 3.2.3 and hence omitted.

In the following lemma we consider voccp related to order $\sigma \in \Pi(N)$ in a mcst situation with two sources $(N, \{0_1, 0_2\}, w)$ and voccp related to same order in the associated mcst situation $(N, 0, w_0)$. Let us denote the k^{th} edge constructed by voccp related to σ in $(N, \{0_1, 0_2\}, w)$ by e_k^σ for every $k \in \{1, 2, ..., n+1\}$ and the k^{th} edge constructed by voccp related to σ in $(N, 0, w_0)$ by f_k^σ for every $k \in \{1, 2, ..., n\}$. We show that each of the first n edges constructed by the two algorithms cost the same, i.e., $w(e_k^\sigma) = w_0(f_k^\sigma)$ for every $k \in \{1, 2, ..., n\}$.

Lemma 3.4.3 *Let $(N, \{0_1, 0_2\}, w)$ be an mcst situation with two sources , $(N, 0, w_0)$ be the associated mcst situation and $\sigma \in \Pi(N)$. Then, $w(e_k^\sigma) = w_0(f_k^\sigma)$ for every $k \in \{1, 2, ..., n\}$.*

Proof. Let $\sigma \in \Pi(N)$. Consider first the voccp related to order σ in the mcst situation with two sources. Pick $k \in \{1, 2, ..., n\}$ and let $\sigma(k) = i$. We know by Lemma 3.4.2 that neither 0_1 nor 0_2 belongs to $G_i^{\sigma,k-1}$ the component that agent i belongs in $G^{\sigma,k-1}$. So, both edges $\{i, 0_1\}$ and $\{i, 0_2\}$ are component reducing for i in $G^{\sigma,k-1}$. Naturally, the more expensive of the two edges will never be constructed by i.

So, starting with the first step of both procedures, voccp related to σ in w and voccp related to σ in the associated mcst situation w_0, each agent can select to construct the same edge in both procedures in the following sense: if agent i chooses to construct an edge $e \subset N$ in voccp related to σ in w, then e is also a cheapest component reducing edge for agent i in voccp related to σ in w_0 and if agent i chooses to construct the cheaper of the two edges $\{i, 0_1\}$ and $\{i, 0_2\}$ in voccp related to σ in w_0 then the edge $\{i, 0\}$ is a cheapest component reducing edge for agent i in voccp related to σ in w_0. Since agents can select the same edges in the first n steps of both procedures $w(e_k^\sigma) = w_0(f_k^\sigma)$. □

We know by Lemma 3.4.1 that $ERO_{\mathcal{T}}(w) = ERO(w_0)$. Hence, Lemma 3.4.3 enables us to establish the relationship between the costs of the first n edges constructed by voccp and $ERO_{\mathcal{T}}(w)$. In the following lemma, we will establish the relationship between the cost of the last edge constructed by voccp and $ERO_{\mathcal{T}}(w)$.

Lemma 3.4.4 *Let $(N, \{0_1, 0_2\}, w)$ be an mcst situation with two sources , $\sigma \in \Pi(N)$ and $\pi \in \Pi(E_{N'})$ be such that $w \in K^\pi$. Then, $w(e_{n+1}^\sigma) = \sum_{i \in N} e^{\{i\}} M_2^\pi w^\pi$.*

Proof. Let Γ be an mcst for the mcst situation with two sources w and Γ' be an mcst for the associated mcst situation w_0. Now, on the one hand

$$\begin{aligned}
w(\Gamma) &= w(e^\sigma_{n+1}) + \sum_{k\in\{1,2,\ldots,n\}} w(e^\sigma_k) = w(e^\sigma_{n+1}) + \sum_{k\in\{1,2,\ldots,n\}} w(f^\sigma_k) \\
&= w(e^\sigma_{n+1}) + w_0(\Gamma') = w(e^\sigma_{n+1}) + \sum_{i\in N} ERO(w_0)_i \\
&= w(e^\sigma_{n+1}) + \sum_{i\in N} e^{\{i\}} M_1^\pi w^\pi, \qquad (3.11)
\end{aligned}$$

where the first equality follows from the efficiency of voccp in mcst situations with two sources, the second equality follows from Lemma 3.4.3, the third equality follows from the efficiency of voccp in mcst situations, the fourth equality follows from the efficiency of ERO in mcst situations and the last equality follows from Lemma 3.4.1.

On the other hand

$$w(\Gamma) = \sum_{i\in N} ERO_{\mathcal{T}}(w)_i = \sum_{i\in N} \left(e^{\{i\}} M_1^\pi w^\pi + e^{\{i\}} M_2^\pi w^\pi\right). \qquad (3.12)$$

Then by equations (3.11) and (3.12), $w(e^\sigma_{n+1}) = \sum_{i\in N} e^{\{i\}} M_2^\pi w^\pi$. $\square$

Finally, we prove that $ERO_{\mathcal{T}}$ can be obtained as the average of the cost allocations provided by voccp for every order of the agents.

Theorem 3.4.2 *Let $w \in \mathcal{W_T}^N$. Then*

$$ERO_{\mathcal{T}}(w) = \sum_{\sigma\in\Pi(N)} \frac{v^\sigma_{\mathcal{T}}(w)}{n!}.$$

Proof. Let w be an mcst with two sources and w_0 be the associated mcst situation, $\sigma \in \Pi(N)$, $k \in \{1,...,n\}$ and $\sigma(k) = i$. We know by Lemma 3.4.3 that if i is not the last agent with respect to σ then

$$v^\sigma_{\mathcal{T},i} = w(e^\sigma_k) = w_0(e^\sigma_k) = v^\sigma_i(w_0),$$

i.e., the cost $v^\sigma_{\mathcal{T},i}$ assigned by voccp related to σ to agent i in w is equal to the cost $v^\sigma_i(w_0)$ assigned by voccp related to σ to agent i in w_0.

We also know by lemmas 3.4.3 and 3.4.4 that if i is the last agent with respect to σ then

$$v^\sigma_{\mathcal{T},i} = w(e^\sigma_n) + w(e^\sigma_{n+1}) = w_0(e^\sigma_n) + \sum_{i\in N} e^{\{i\}} M_2^\pi w^\pi = v^\sigma_i(w_0) + \sum_{i\in N} e^{\{i\}} M_2^\pi w^\pi,$$

i.e., the cost $v^\sigma_{\mathcal{T},i}$ assigned by voccp related to σ to agent i in w is equal to the sum of the cost $v^\sigma_i(w_0)$ assigned by voccp related to σ to agent i in w_0 and $\sum_{i\in N} e^{\{i\}} M_2^\pi w^\pi$.

Then

$$\begin{aligned}
\sum_{\sigma\in\Pi(N)} \frac{v^\sigma_{\mathcal{T}}(w)_i}{n!} &= \frac{1}{n!}\left(\sum_{\sigma\in\Pi(N):i\neq\sigma(n)} v^\sigma(w_0)_i + \sum_{\sigma\in\Pi(N):i=\sigma(n)} v^\sigma(w_0)_i + \sum_{j\in N} e^{\{j\}} M_2^\pi w^\pi\right),\\
&= \frac{1}{n!}\sum_{\sigma\in\Pi(N)} v^\sigma(w_0)_i + \frac{1}{n!}\sum_{\sigma\in\Pi(N):i=\sigma(n)}\sum_{j\in N} e^{\{j\}} M_2^\pi w^\pi,\\
&= ERO(w_0)_i + \frac{1}{n!}\frac{n!}{n}\sum_{j\in N} e^{\{j\}} M_2^\pi w^\pi,\\
&= e^{\{i\}} M_1^\pi w^\pi + \frac{1}{n}\sum_{j\in N} e^{\{j\}} M_2^\pi w^\pi,\\
&= e^{\{i\}} M_1^\pi w^\pi + e^{\{i\}} M_2^\pi w^\pi = ERO_{\mathcal{T}}(w)_i,
\end{aligned}$$

where the first equality follows from Lemmas 3.4.3 and 3.4.4, the third equality follows from the equality of ERO and the average of the cost allocations provided by voccp in mcst situations and the fact that there are $\frac{n!}{n}$ orders at which i is the last player, the fourth equality follows from Lemma 3.4.1 and the last equality follows from the fact that $e^{\{j\}} M_2^\pi w^\pi = e^{\{j'\}} M_2^\pi w^\pi$ for every $j, j' \in N$. □

3.4.3 Voccp and the optimistic game in mcst situations with two sources

In this section, we define an optimistic transferable utility game for mcst problems with two sources and show that for every ordering σ of the agents, the $v^\sigma_{\mathcal{T}}$ value is equal to the marginal of the optimistic game for the same ordering.

The optimistic game for mcst situations with two sources is defined as follows. Given an mcst situation with two sources $(N, \{0_1, 0_2\}, w)$ and a nonempty coalition of agents $S \subsetneq N$, we first obtain an optimistic mcst situation for S assuming that when the agents in S have to be connected to both of the sources, the agents in $N\backslash S$ are already connected to sources and the agents in S can connect to sources via agents in $N\backslash S$. Formally, for every $S \subsetneq N$, *the optimistic mcst situation for* S is the mcst situation $(S, 0, w_S)_{\mathcal{T}}$, where $w_S(e) = w(e)$ for every $e \in E_S$ and $w_S(\{i, 0\}) = \min_{j\in N'\backslash S} w(\{i, j\})$ for every $i \in S$. Naturally, the optimistic mcst situation for N is the original mcst situation with two sources $(N, \{0_1, 0_2\}, w)$.

Now the *optimistic TU-game* (N, v) associated with the mcst situation with two sources $(N, \{0_1, 0_2\}, w)$ is defined as $v(S) = w_S(\Gamma)$ where Γ is a mcst for the optimistic game for S for every $S \subset N$ with $v(\emptyset) = 0$.

Example 3.4.5 Consider the mcst situation with two sources $(N, \{0_1, 0_2\}, w)$ with $N = \{1, 2, 3\}$ and w as depicted in Figure 3.5. Consider coalition $\{1, 2\}$. In Figure 3.7 we provide the optimistic mcst situation for $\{1, 2\}$ and a related mcst. Hence the worth of coalition $\{1, 2\}$ in the optimistic TU-game associated with $(N, \{0_1, 0_2\}, w)$ is 10. The complete optimistic TU-game associated with $(N, \{0_1, 0_2\}, w)$ is given below:

Figure 3.7: The optimistic mcst situation for $\{1, 2\}$ (left side) and a related mcst

S	{1}	{2}	{3}	{1,2}	{1,3}	{2,3}	N
$v(S)$	4	4	3	10	7	7	17

$\diamond$

In the following, we show that for every ordering of the agents, the cost allocation provided by the related voccp and the related marginal vector of the optimistic game are equal to each other. First, we consider in the following lemma two optimistic mcst games, one derived from an mcst situation with two sources and the other derived from the associated mcst situation. We show that for proper subsets of N they are equal to each other

Lemma 3.4.5 *Let $w \in \mathcal{W_T}^N$ and let $w_0 \in \mathcal{W}^N$ be the associated mcst situation. Also let (N, v) be the optimistic TU-game associated with w and (N, v_0) be the optimistic TU-game associated with w_0. Then, $v(S) = v_0(S)$ for every $S \subsetneq N$.*

Proof. Let $(S, 0, w_S)_\mathcal{T}$ be the optimistic mcst situation for S derived from w and let $(S, 0, w_{0,S})$ be the optimistic mcst situation for S derived from w_0. We want to show that $w_S(e) = w_{0,S}(e)$ for every $e \in E_{S\cup\{0\}}$. Obviously, $w_S(e) = w_{0,S}(e) = w(e)$ for every $e \in E_S$. So consider an edge $\{i, 0\}$ for some $i \in S$. We know that $w_S(\{i, 0\}) = \min_{j\in N\cup\{0_1,0_2\}\setminus S} w(\{i, j\})$ and $w_{S,0}(\{i, 0\}) = \min_{j\in N\cup\{0\}\setminus S} w_0(\{i, j\})$. We also know that $w_0(\{i, 0\}) = min\{\{i, 0_1\}, \{i, 0_2\}\}$. Hence, $w_S(\{i, 0\}) = w_{0,S}(\{i, 0\})$. So, we can conclude that $w_S = w_{0,S}$. But, since $w_S = w_{0,S}$, the cost of an mcst of $(S, 0, w_S)_\mathcal{T}$ is equal to the cost of an mcst of $(S, 0, w_{0,S})$ and hence $v(S) = v_0(S)$. $\square$

Proposition 3.4.2 *Let $w \in \mathcal{W_T}^N$. Then $v^\sigma_\mathcal{T}(w) = m^\sigma(v)$ for every $\sigma \in \Pi(N)$.*

Proof. Let $w \in \mathcal{W}_{\mathcal{T}}{}^N$, $w_0 \in \mathcal{W}^N$ be the associated mcst situation and (N, c) be the optimistic TU-game associated with w_0. Also let $\sigma \in \Pi(N)$ and denote by S_k the set of first k agents with respect to order σ, i.e., $S_k = \{\sigma(1), \sigma(2), ..., \sigma(k)\}$ for every $k \in \{1, 2, ..., n\}$. Lastly let $S_0 = \emptyset$.

Now let $k \in \{1, ..., n-1\}$ and $\sigma(k) = i$. Then

$$m^\sigma(v)_i = v(S_k) - v(S_{k-1}) = c(S_k) - c(S_{k-1}) = v^\sigma(w_0)_i = v^\sigma_{\mathcal{T}}(w)_i, \tag{3.13}$$

where the second equality follows from Lemma 3.4.5, the third equality follows from Proposition 3.2.6 and the last equality follows from Lemma 3.4.3.

Note that by (3.13),

$$v(S_k) = \sum_{i \in S_k} v^\sigma_{\mathcal{T}}(w)_i = \sum_{t \in \{1,2,...,k\}} w(e^\sigma_t)$$

for every $k \in \{1, 2, ..., n-1\}$.

Now let $k = n$ and consider $\sigma(k) = i$ the last agent with respect to σ. Then,

$$m^\sigma(v)_i = v(N) - v(S_{n-1}) = v(N) - \sum_{t \in \{1,2,...n-1\}} w(e^\sigma_t) = w(e^\sigma_n) + w(e^\sigma_{n+1}) = v^\sigma_{\mathcal{T}}(w)_i,$$

where the second equality follows from the fact that $v(S_{n-1})$ is equal to the total cost of first $n-1$ edges constructed by voccp related to σ and the third equality follows from the fact that $v(N)$ is equal to the cost of an mcst of the mcst situation with two sources w.

Hence, we can conclude that $v^\sigma_{\mathcal{T}}(w) = m^\sigma(v)$. $\square$

Since the Shapley value of a game is the average of its marginals, Proposition 3.4.2 implies that $ERO_{\mathcal{T}}$ is equal to the Shapley value of the optimistic game.

Corollary 3.4.1 *Let $w \in \mathcal{W}_{\mathcal{T}}{}^N$. Then $ERO_{\mathcal{T}}(w)$ is equal to the Shapley value of the optimistic game associated with w.*

3.5 Highway Games on Weakly Cyclic Graphs

In this section, we study the concavity and the balancedness of highway games on *weakly cyclic graphs*. Section 3.5.1 formally introduces highway problems and highway games. Section 3.5.2 presents the characterization of HG-concave graphs. Section 3.5.3 proves that highway games on weakly cyclic graphs are balanced.

3.5.1 Highway problems and highway games

A *highway problem* is defined as a tuple $\Gamma = (N, G, \{s_i\}_{i\in N}, \{t_i\}_{i\in N}, w)$. $N = \{1, ..., n\}$ is a nonempty, finite set of players and G=(V, E) is a connected graph. The graph G determines the possibilities regarding the construction of the highway network. That is, any constructed highway network has to be a subgraph of G. Note that G need not be the complete graph, since the construction of some edges may be infeasible due to geographic or socioeconomic reasons. For each player $i \in N$, s_i and t_i are vertices in G and they are called the *connection vertices* of i. The connection vertices of player i represent the locations (think of entry and exit) that i has to establish a connection between. Finally, $w : E \to \mathbb{R}_+$ is a *cost function* and associates to each edge, $e \in E$, the nonnegative cost $w(e)$ of constructing e. The total cost of constructing a set of edges $E' \subset E$ is abbreviated by $w(E') = \sum_{e\in E'} w(e)$. In the following we do not always distinguish strictly between a graph and its edge set. For example, given a graph $G = (V', E')$, we may write $w(G')$ rather than $w(E')$.

In a highway problem, a coalition S of cooperating players will construct a cheapest set of edges that connects the connection points of every member of S. Therefore, given a highway problem $\Gamma = (N, G, \{s_i\}_{i\in N}, \{t_i\}_{i\in N}, w)$, the corresponding *highway game* (N, c_Γ) is defined by

$$c_\Gamma(S) = \min_{E'\subset E} \{w(E')|s_i \text{ and } t_i \text{ are connected in } (V, E') \text{ for every } i \in S\} \quad (3.14)$$

for all $S \subset N$. Clearly, (N, c_Γ) is subadditive and monotonic.

Example 3.5.1 Consider the cycle $C = v_1v_2v_3v_4v_1$. The construction costs of edges are given by: $w(\{v_1, v_2\}) = w(\{v_2, v_3\}) = 2$ and $w(\{v_3, v_4\}) = w(\{v_4, v_1\}) = 3$. Consider $N = \{1, 2, 3\}$ with $s_1 = v_1$, $t_1 = v_3$, $s_2 = v_2$, $t_2 = v_3$, $s_3 = v_4$, $t_3 = v_1$. The corresponding highway problem Γ is depicted in Figure 3.8.

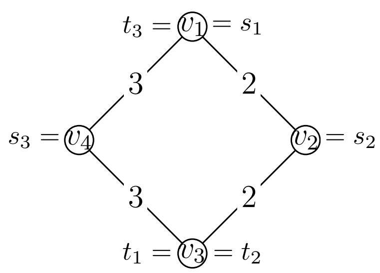

Figure 3.8: A highway problem with three players

Consider player 1. There are two paths in C between player 1's connection vertices v_1 and v_3: $v_1v_2v_3$ and $v_1v_4v_3$. Since player 1 will not construct any superfluous edges, $c_\Gamma(\{1\})$ is the minimum of $w(\{v_1, v_2\}) + w(\{v_2, v_3\}) = 4$ and $w(\{v_1, v_4\}) + w(\{v_4, v_3\}) = 6$, i.e., $c_\Gamma(\{1\}) = 4$.

Next, consider the coalition $\{1, 3\}$. Clearly, players 1 and 3 will construct the set of edges $\{\{v_1, v_4\}, \{v_3, v_4\}\}$. Hence, $c_\Gamma(\{1, 3\}) = 6$.

The complete corresponding highway game (N, c_Γ) is given in Table 3.8 below.

S	{1}	{2}	{3}	{1,2}	{1,3}	{2,3}	N
c_Γ(S)	4	2	3	4	6	5	7

Table 3.8: The highway game (N, c_Γ) in Example 3.5.1.

Observe that this highway game is not concave:

$$c_\Gamma(\{1, 2\}) + c_\Gamma(\{1, 3\}) < c_\Gamma(\{1\}) + c_\Gamma(\{1, 2, 3\}).$$

◇

The above example illustrates that if there are multiple paths between the connection vertices of players in the underlying graph, then players can select different paths in different coalitions and this may lead to the violation of concavity conditions of the associated highway game. When the underlying graph is a tree as it is the case for the highway problems considered by Mosquera and Zarzuelo (2006), the players use the unique path between their connection vertices independent of the coalition they belong to. From this, it can readily be derived that the highway games induced by trees are concave.

3.5.2 HG-Concavity

In this section, we characterize HG-concave graphs. Here, a graph G is called *HG-concave* if for every highway problem $\Gamma = (N, G, \{s_i\}_{i\in N}, \{t_i\}_{i\in N}, w)$, the corresponding highway game (N, c_Γ) is concave. Explicitly, we show that a graph is HG-concave if and only if it is weakly triangular.

For this aim, we first show that every highway game on a cycle of length 3 is concave.

Lemma 3.5.1 *Let $\Gamma = (N, C, \{s_i\}_{i\in N}, \{t_i\}_{i\in N}, w)$ be a highway problem where C is a cycle of length 3. Then, the corresponding highway game (N, c_Γ) is concave.*

Proof. Without loss of generality, assume that $s_i \neq t_i$ for all $i \in N$. It can easily be observed that

(i) for every coalition, it is optimal to construct either one edge in C or the two cheaper edges in C;

(ii) for a coalition $S \subset N$, if it is optimal to construct just one edge $\{u, v\}$ in C, then $\{s_i, t_i\} = \{u, v\}$ for every $i \in S$;

(iii) for a coalition $S \subset N$, if it is optimal to construct the two cheaper edges in C, then constructing the two cheaper edges is optimal for any superset of S, too.

Now, we will show that $c_\Gamma(S \cup T) + c_\Gamma(S \cap T) \leq c_\Gamma(S) + c_\Gamma(T)$ for every $S, T \subset N$, i.e., that the corresponding highway game (N, c_Γ) is concave. Take $S, T \subset N$ and assume that $S \cap T \neq \emptyset$. For, if $S \cap T = \emptyset$, then the inequality follows directly from the subadditivity of (N, c_Γ).

Firstly, (i) implies that, for any coalition $K \subset N$, $c_\Gamma(K)$ is either equal to the sum of the costs of the two cheaper edges in C or equal to the cost of one of the three edges in C.

If both $c_\Gamma(S)$ and $c_\Gamma(T)$ are equal to the sum of the costs of the two cheaper edges in C, then the inequality follows from the monotonicity of c_Γ and (iii). If only one of $c_\Gamma(S)$ and $c_\Gamma(T)$ is equal to the sum of the costs of the two cheaper edges in C and the other is equal to the cost of one edge, say e, in C, then (iii) implies that $c_\Gamma(S \cup T)$ is equal to the sum of the costs of the two cheaper edges in C and (ii) implies that $c_\Gamma(S \cap T)$ is also equal to the cost of e. Hence, $c_\Gamma(S \cup T) + c_\Gamma(S \cap T) = c_\Gamma(S) + c_\Gamma(T)$.

Lastly, assume that $c_\Gamma(S)$ and $c_\Gamma(T)$ are equal to the cost of an edge in C. Then, since $S \cap T \neq \emptyset$, (ii) implies that $c_\Gamma(S)$ and $c_\Gamma(T)$ have to be equal to the cost of the same edge. Then, both $c_\Gamma(S \cup T)$ and $c_\Gamma(S \cap T)$ are equal to the cost of the same edge, too. Hence, $c_\Gamma(S \cup T) + c_\Gamma(S \cap T) = c_\Gamma(S) + c_\Gamma(T)$. □

We now discuss some properties of weakly cyclic graphs. Let G=(V, E) be a weakly cyclic graph. Clearly, each edge in G is either a bridge edge or belongs to exactly one cycle in G. Let $\mathcal{C}(G)$ denote the set of cycles in G and $\mathcal{BE}(G)$ denote the set of bridge edges in G. Observe that every path in G which connects two vertices has to pass through (has a common edge with) the same set of cycles and the same set of bridge edges in G. More specifically, every path that connects the same two vertices, passes through the same cycles and the same bridge edges but the edges followed in cycles may differ. Moreover, every path that connects the same two vertices in G enter and leave a cycle that they pass through at the same vertices.

Before presenting the main result of this section, we will show that for every highway problem on a weakly cyclic graph, the corresponding highway game is equal to the sum of specific sub-highway games on each cycle and on each bridge edge in the graph. These sub-highway games are formally defined as follows.

Consider a highway problem $\Gamma = (N, G, \{s_i\}_{i\in N}, \{t_i\}_{i\in N}, w)$ where G is a weakly cyclic graph. Let $C = v_1v_2...v_kv_1$ be a cycle in G. Now, the sub-highway problem with respect to C is defined by $\Gamma^C = (N, C, \{s_i^C\}_{i\in N}, \{t_i^C\}_{i\in N}, w_{|C})$, where $w_{|C}$ is the restriction of the cost function w to the edges in C. For each player $i \in N$, if the paths connecting s_i and t_i pass through C, then s_i^C and t_i^C are the vertices in C at which the paths connecting s_i and t_i enter and leave C. If the paths connecting s_i and t_i do not pass through C, then we set $s_i^C = t_i^C = v_1$.

Next, let $e = \{u, v\}$ be a bridge edge in G. Then, the sub-highway problem with respect to e is defined by $\Gamma^e = (N, (\{u, v\}, \{e\}), \{s_i^e\}_{i\in N}, \{t_i^e\}_{i\in N}, w_{|e})$. Set $s_i^e = u$ and $t_i^e = v$ if the paths connecting s_i and t_i pass through e. Otherwise, set $s_i^e = t_i^e = u$.

Lemma 3.5.2 *Let $\Gamma = (N, G, \{s_i\}_{i\in N}, \{t_i\}_{i\in N}, w)$ be a highway problem where G is a weakly cyclic graph. Then, $c_\Gamma(S) = \sum_{C\in\mathcal{C}(G)} c_{\Gamma^C}(S) + \sum_{e\in\mathcal{BE}(G)} c_{\Gamma^e}(S)$ for every $S \subset N$.*

We omit the proof of Lemma 3.5.2 since it is straightforward[2].

We are now ready to present the main result of this section.

Theorem 3.5.1 *A graph G is HG-concave if and only if it is weakly triangular.*

Proof. We first show the if-part. Let G=(V, E) be a weakly triangular graph and consider a highway problem $\Gamma = (N, G, \{s_i\}_{i\in N}, \{t_i\}_{i\in N}, w)$. We will show that the corresponding highway game (N, c_Γ) is concave.

We know by Lemma 3.5.2 that $c_\Gamma(S) = \sum_{C\in\mathcal{C}(G)} c_{\Gamma^C}(S) + \sum_{e\in\mathcal{BE}(G)} c_{\Gamma^e}(S)$ for every $S \subset N$. By Lemma 3.5.1, we have that c_{Γ^C} is concave for every triangle $C \in \mathcal{C}(G)$ and we also know that highway games induced by trees are concave. In particular, c_{Γ^e} is concave for every $e \in \mathcal{BE}(G)$. We may conclude that c_Γ is concave, since it is a non-negative linear combination of concave games.

For the only-if part of the proof, choose a graph G=(V, E) that is not weakly triangular. Now, we construct a player set N, connection vertices s_i, t_i for each player i in N and a cost function w such that the highway game corresponding to the highway problem $(N, G, \{s_i\}_{i\in N}, \{t_i\}_{i\in N}, w)$ is not concave.

Since G is not weakly triangular, it contains a cycle $C = v_1v_2...v_kv_1$ with length $k \geq 4$. Let the player set be $N = \{1, 2, 3\}$ and let $s_1 = v_1$, $t_1 = v_3$, $s_2 = v_2$, $t_2 = v_3$

[2]Lemma 3.5.2 can be extended to decompose a highway game induced by an arbitrary graph into the sum of sub-highway games associated to the bridge edges and the components obtained after deleting all bridge edges.

and $s_3 = v_k$, $t_3 = v_1$. Define the cost function w by:

$$w(e) = \begin{cases} 2, & \text{if } e \in \{\{v_1, v_2\}, \{v_2, v_3\}\}, \\ 0, & \text{if } e \in \{\{v_3, v_4\}, ..., \{v_{k-2}, v_{k-1}\}\}, \\ 3, & \text{if } e \in \{\{v_k, v_1\}, \{v_k, v_{k-1}\}\}, \\ 100, & \text{if } e \notin C. \end{cases}$$

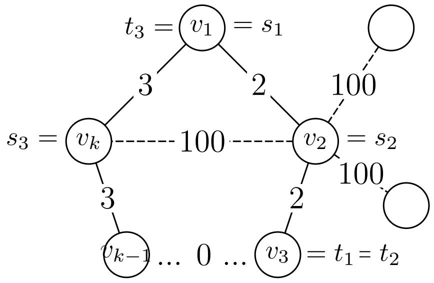

Figure 3.9: An auxiliary figure for the proof of Theorem 3.5.1

Figure 3.9 depicts a part of the highway problem $\Gamma = (N, G, \{s_i\}_{i \in N}, \{t_i\}_{i \in N}, w)$. Now, it can easily be shown that the highway game corresponding to the highway problem Γ is equal to the highway game presented in Example 3.5.1, which is not concave. □

3.5.3 Balancedness of Highway Games on Weakly Cyclic Graphs

In this section, we show that highway games induced by weakly cyclic graphs are balanced. First we provide an example (cf. Kuipers, 1997) which shows that highway games in which the underlying graphs allow for multiple paths between vertices need not be balanced in general.

Example 3.5.2 Let G=(V, E) be the complete graph on $V = \{v_1, v_2, ..., v_6\}$. The construction costs of the edges are given by:

$$w(e) = \begin{cases} 5, & \text{if } e \in \{\{v_1, v_4\}, \{v_2, v_5\}, \{v_3, v_6\}, \{v_4, v_5\}, \{v_4, v_6\}, \{v_5, v_6\}\}, \\ 3, & \text{otherwise.} \end{cases}$$

Consider $N = \{1, 2, 3\}$ with $s_1 = v_1$, $t_1 = v_4$, $s_2 = v_2$, $t_2 = v_5$, $s_3 = v_3$, $t_3 = v_6$. A part of the corresponding highway problem $\Gamma = (N, G, \{s_i\}_{i \in N}, \{t_i\}_{i \in N}, w)$, where only the edges of cost 3 are drawn, is depicted in Figure 3.10.

It can easily be observed that the cost of any two-player coalition is 9. For example, the optimal collection of edges for the coalition $\{1, 2\}$ consists of edges

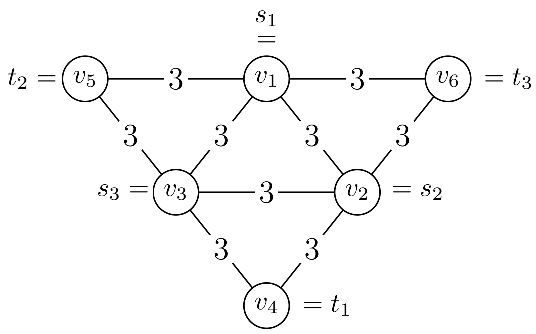

Figure 3.10: An auxiliary figure for Example 3.5.2

$\{\{v_5, v_1\}, \{v_1, v_2\}, \{v_2, v_4\}\}$ with a total cost of 9. There are several possibilities for the optimal collection of edges for the grand coalition. One of them can be obtained by adding the edge $\{v_3, v_6\}$ to the optimal collection of edges of the coalition $\{1, 2\}$. Hence, $c_\Gamma(N) = 14$. Suppose that (x_1, x_2, x_3) is a core element of (N, c_Γ). Then, it can be shown by using the core conditions for coalitions $\{1, 2\}$, $\{1, 3\}$ and $\{2, 3\}$ that

$$2(x_1 + x_2 + x_3) \leq c_\Gamma(\{1, 2\}) + c_\Gamma(\{1, 3\}) + c_\Gamma(\{2, 3\}) = 27 < 2c_\Gamma(N),$$

a contradiction with the efficiency of a core allocation. Hence, (N, c_Γ) is not balanced. $\diamond$

In the following, we first focus on highway problems on cycles and prove that the induced highway games are balanced.

Let $(N, C, \{s_i\}_{i\in N}, \{t_i\}_{i\in N}, w)$ be a highway problem on a cycle $C = (V, E)$. Observe that each player has two alternative paths between his connection vertices in C. For player i, we denote *an individually optimal path for* i by P_i and *the alternative path* by Q_i. Note that $P_i \cap Q_i = \emptyset$ and $P_i \cup Q_i = E$ for every player $i \in N$. Let $S \subset N$. A collection of paths, $\{R_i\}_{i\in S}$ with $R_i \in \{P_i, Q_i\}$, for all $i \in S$ is called a *path profile for* S. The set of path profiles for S is denoted by $\mathcal{R}^S$. For each path profile $R \in \mathcal{R}^S$, the set of edges $\bigcup_{i\in S} R_i$ is called *the set of edges corresponding to* R and is denoted by $E(R)$. $\bar{E}(R)$ is defined as the complement of $E(R)$, i.e., $\bar{E}(R) = E \backslash E(R)$. Let $T = \{T_i\}_{i\in N} \in \mathcal{R}^N$. The *restriction of* T *to* S, $T_{|S} = \{(T_{|S})_i\}_{i\in S}$, is the path profile for S defined by $(T_{|S})_i = T_i$ for every $i \in S$.

Every set of edges which connects the connection vertices of every member of a coalition of players has to contain either the individually optimal path or the alternative path of each member of the coalition. Moreover, since a coalition will not construct any superfluous edges, a cheapest set of edges that connects the connection vertices of every member of a coalition will be equal to a set of edges corresponding to a path profile. Hence, the highway game (N, c_Γ) corresponding to the highway problem $(N, C, \{s_i\}_{i\in N}, \{t_i\}_{i\in N}, w)$ on a cycle $C = (V, E)$ can be reformulated as

$$c_\Gamma(S) = \min_{R\in\mathcal{R}^S}\{w(E(R))\}, \tag{3.15}$$

or equivalently as

$$c_\Gamma(S) = w(E) - \max_{R\in\mathcal{R}^S}\{w(\bar{E}(R))\}, \tag{3.16}$$

for every $S \subset N$.

Lemma 3.5.3 *Let $\Gamma = (N, C, \{s_i\}_{i\in N}, \{t_i\}_{i\in N}, w)$ be a highway problem such that $C = (V, E)$ is a cycle and let $S \subset N$.*

(i) If $R, R' \in \mathcal{R}^S$ with $R \neq R'$, then $\bar{E}(R) \cap \bar{E}(R') = \emptyset$.

(ii) $\displaystyle\bigcup_{R\in\mathcal{R}^S} \bar{E}(R) = E$.

(iii) $w(E) = \sum_{R\in\mathcal{R}^S} w(\bar{E}(R))$.

(iv) $\displaystyle\bar{E}(R) = \bigcup_{T\in\mathcal{R}^N:T_{|S}=R} \bar{E}(T)$ *for every* $R \in \mathcal{R}^S$.

(v) $\displaystyle w(\bar{E}(R)) = \sum_{T\in\mathcal{R}^N:T_{|S}=R} w(\bar{E}(T))$ *for every* $R \in \mathcal{R}^S$.

Proof. (i) Let $R, R' \in \mathcal{R}^S$ with $R \neq R'$. Clearly, there exists $i \in S$ such that $R_i \neq R'_i$. Assume without loss of generality that $R_i = P_i$ and $R'_i = Q_i$. Then, $\bar{E}(R) \subset Q_i$ and $\bar{E}(R') \subset P_i$, and hence $\bar{E}(R) \cap \bar{E}(R') = \emptyset$.

(ii) Obviously, $\displaystyle\bigcup_{R\in\mathcal{R}^S} \bar{E}(R) \subset E$. Now, pick $e \in E$. Consider $R \in \mathcal{R}^S$ such that $R_i = P_i$ if $e \in Q_i$ and $R_i = Q_i$ otherwise. Clearly, $e \notin R_i$ for any $i \in S$ and hence $e \in \bar{E}(R)$. So, $\displaystyle E \subset \bigcup_{R\in\mathcal{R}^S} \bar{E}(R)$.

(iii) readily follows from (i) and (ii).

(iv) Let $R \in \mathcal{R}^S$. Let $T \in \mathcal{R}^N$ be such that $T_{|S} = R$. Then

$$\bar{E}(T) = \left(E \backslash \bigcup_{i\in N} T_i\right) \subset \left(E \backslash \bigcup_{i\in S} T_i\right) = \bar{E}(R),$$

and hence $\displaystyle\bar{E}(R) \supset \bigcup_{T\in\mathcal{R}^N:T_{|S}=R} \bar{E}(T)$.

Now pick $e \in \bar{E}(R)$. Consider $T \in \mathcal{R}^N$ with $T_{|S} = R$ and for every $i \in N \backslash S$, $T_i = P_i$ if $e \in Q_i$ and $T_i = Q_i$ otherwise. Obviously, $e \notin T_i$ for any $i \in N$ and hence $e \in \bar{E}(T)$. So, $\bar{E}(R) \subset \bigcup_{T \in \mathcal{R}^N : T_{|S} = R} \bar{E}(T)$.

(v) readily follows from (i) and (iv). □

Let us denote with R_S^* an optimal path profile for S, i.e., $c_\Gamma(S) = w(E) - w(\bar{E}(R_S^*))$ (cf. (3.16)) for every $S \subset N$ with $S \neq \emptyset$. Let $\mathcal{S} = (S_1, S_2, ..., S_m)$ be a sequence[3] of coalitions in N. For every path profile T for N, we denote by $\alpha(T, \mathcal{S})$ the number of coalitions S in sequence $\mathcal{S}$ such that the restriction of T to S is equal to R_S^*, i.e.,

$$\alpha(T, \mathcal{S}) = \left| \left\{ t \in \{1, 2, ..., m\} | T_{|S_t} = R_{S_t}^* \right\} \right|.$$

For every $x \in \mathbb{R}$, $(x)_+$ is defined as $(x)_+ = \max\{x, 0\}$.

Lemma 3.5.4 *Let* $\Gamma = (N, C, \{s_i\}_{i \in N}, \{t_i\}_{i \in N}, w)$ *be a highway problem such that* $C = (V, E)$ *is a cycle. Let* $\mathcal{S} = (S_1, S_2, ..., S_m)$ *be a sequence of coalitions in* N *and* $l \in \{1, 2, ..., m\}$[4].

$$\textit{If} \sum_{T \in \mathcal{R}^N} (\alpha(T, \mathcal{S}) - (m - l))_+ > l \textit{ then there exists } j \in N \textit{ such that}$$

$$|\{t \in \{1, 2, ..., m\} | j \in S_t\}| < l.$$

Proof. The proof will proceed by induction on l.

Let us first prove the assertion for an arbitrary $m \in \{1, 2, ...\}$ and $l = 1$. Assume that $\sum_{T \in \mathcal{R}^N} (\alpha(T, \mathcal{S}) - (m - 1))_+ > 1$. Then there exists $R, R' \in \mathcal{R}^N$ with $R \neq R'$ such that $\alpha(R, \mathcal{S}) = \alpha(R', \mathcal{S}) = m$, i.e., $R_{|S_t} = R'_{|S_t} = R_{S_t}^*$ for every $t \in \{1, 2, ..., m\}$. Since $R \neq R'$, there exists $j \in N$ such that $R_j \neq R'_j$. Hence, $j \notin S_t$ for every $t \in \{1, 2, ..., m\}$, i.e., $|\{t \in \{1, 2, ..., m\} | j \in S_t\}| = 0 < 1$.

Let us now prove the assertion for $l \geq 2$ and an arbitrary $m \geq l$. Assume that the assertion holds for every $l' < l$ and arbitrary $m' \geq l'$.

[3]What we need of the sequence $\mathcal{S}$ is the fact that the same coalition S can occur several times, the precise order of the coalitions is not relevant.

[4]In Theorem 3.5.2, we will use Lemma 3.5.4 in order to prove the balancedness of highway games induced by cycles. There l will be taken as the smallest strictly positive integer such that $l\lambda_S$ is an integer for every coalition S in a balanced set $\mathcal{B}$ with balancing weights $(\lambda_S)_{S \in \mathcal{B}}$ and a sequence $\mathcal{S}$ will be constructed by duplicating each $S \in \mathcal{B}$ $l\lambda_S$ times. Hence, m will be equal to $\sum_{S \in \mathcal{B}} l\lambda_S$.

Case 1: For all $i \in \bigcup_{t=1}^{m} S_t$ there exists $R_i \in \{P_i, Q_i\}$ such that $(R^*_{S_t})_i = R_i$ for every $t \in \{1,2,...,m\}$ such that $i \in S_t$. Fix such R_i for every $i \in \bigcup_{t=1}^{m} S_t$. For $i \in N \backslash \bigcup_{t=1}^{m} S_t$, take $R_i = P_i$.

Consider the path profile $R = \{R_i\}_{i \in N}$. Then $(\alpha(R,\mathcal{S}) - (m-l))_+ = (m-(m-l))_+ = l$. Assume that $\sum_{T \in \mathcal{R}^N} (\alpha(T,\mathcal{S}) - (m-l))_+ > l$. Then there exists $R' \neq R$ such that $\alpha(R',\mathcal{S}) > m-l$. Since $R' \neq R$, either there exists $j \in \bigcup_{t=1}^{m} S_t$ such that $R'_j \neq R_j$ or there exists $j \in N \backslash \bigcup_{t=1}^{m} S_t$ such that $R'_j \neq P_j$. If the latter is the case, then we are automatically finished since $|\{t \in \{1,2,...,m\} | j \in S_t\}| = 0 < l$. If the former is the case, then for every S in sequence $\mathcal{S}$ such that $R'_{|S} = R^*_S$, $j \notin S$. But, then

$$|\{t \in \{1,2,...,m\} | j \in S_t\}| \leq m - \alpha(R',\mathcal{S}) < m-(m-l) = l.$$

Note that for this case we did not have to use the induction hypothesis.

Case 2: There exists $i \in \bigcup_{t=1}^{m} S_t$ and S, S' in the sequence $\mathcal{S}$ such that $i \in S \cap S'$, $(R^*_S)_i = P_i$ and $(R^*_{S'})_i = Q_i$. Choose one such i. If $|\{t \in \{1,2,...,m\} | i \in S_t\}| < l$, then we are automatically finished. So assume that $|\{t \in \{1,2,...,m\} | i \in S_t\}| \geq l$. Reorder the coalitions in $\mathcal{S}$ so that

$$\mathcal{S} = (S_1, ..., S_p, S_{p+1}, ..., S_l, S_{l+1}, ..., S_m), \text{ where } l > p \geq 1 \text{ and}$$

- $i \in S_t$ for every $t \leq l$,
- $(R^*_{S_t})_i = P_i$ for every $t \in \{1,2,...,p\}$,
- $(R^*_{S_t})_i = Q_i$ for every $t \in \{p+1, p+2, ..., l\}$.

Define $\mathcal{S}' = (S_{p+1}, S_{p+2}, ..., S_l, S_{l+1}, ..., S_m)$ and $\mathcal{S}'' = (S_1, S_2, ..., S_p, S_{l+1}, ..., S_m)$. Assume that $\sum_{T \in \mathcal{R}^N} (\alpha(T,\mathcal{S}) - (m-l))_+ > l$.

Case 2.1: $\sum_{T \in \mathcal{R}^N} (\alpha(T,\mathcal{S}') - (m-l))_+ > l-p$. The induction hypothesis implies that there exists $j \in N$ such that $|\{t \in \{p+1,...,l,l+1,...,m\} | j \in S_t\}| < l-p$. But then

$$|\{t \in \{1,2,...,m\} | j \in S_t\}| \leq |\{t \in \{p+1,...,l,l+1,...,m\} | j \in S_t\}| + p < (l-p)+p = l.$$

Case 2.2: $\sum_{T \in \mathcal{R}^N} (\alpha(T,\mathcal{S}') - (m-l))_+ \leq l-p$. Then

$$\begin{aligned}
\sum_{T \in \mathcal{R}^N} (\alpha(T,\mathcal{S}') - (m-l))_+ &= \sum_{T \in \mathcal{R}^N : T_i = P_i} (\alpha(T,\mathcal{S}') - (m-l))_+ + \sum_{T \in \mathcal{R}^N : T_i = Q_i} (\alpha(T,\mathcal{S}') - (m-l))_+ \\
&= \sum_{T \in \mathcal{R}^N : T_i = Q_i} (\alpha(T,\mathcal{S}') - (m-l))_+ \\
&= \sum_{T \in \mathcal{R}^N : T_i = Q_i} (\alpha(T,\mathcal{S}) - (m-l))_+ \leq l-p. \qquad (3.17)
\end{aligned}$$

Also

$$\begin{aligned} l &< \sum_{T\in\mathcal{R}^N} (\alpha(T,\mathcal{S})-(m-l))_+ \\ &= \sum_{T\in\mathcal{R}^N:T_i=P_i} (\alpha(T,\mathcal{S})-(m-l))_+ + \sum_{T\in\mathcal{R}^N:T_i=Q_i} (\alpha(T,\mathcal{S})-(m-l))_+ \end{aligned}$$

and, hence,

$$\sum_{T\in\mathcal{R}^N:T_i=P_i} (\alpha(T,\mathcal{S})-(m-l))_+ > l - \sum_{T\in\mathcal{R}^N:T_i=Q_i} (\alpha(T,\mathcal{S})-(m-l))_+ \geq l-(l-p) = p, \quad (3.18)$$

where the second inequality follows by (3.17).

Now we show that $\sum_{T\in\mathcal{R}^N}\left(\alpha(T,\mathcal{S}'')-(m-l)\right)_+ > p$. This follows from the fact that

$$\begin{aligned} \sum_{T\in\mathcal{R}^N}\left(\alpha(T,\mathcal{S}'')-(m-l)\right)_+ &= \Bigg(\sum_{T\in\mathcal{R}^N:T_i=P_i}\left(\alpha(T,\mathcal{S}'')-(m-l)\right)_+ \\ &\quad + \sum_{T\in\mathcal{R}^N:T_i=Q_i}\left(\alpha(T,\mathcal{S}'')-(m-l)\right)_+\Bigg) \\ &= \sum_{T\in\mathcal{R}^N:T_i=P_i}\left(\alpha(T,\mathcal{S}'')-(m-l)\right)_+ \\ &= \sum_{T\in\mathcal{R}^N:T_i=P_i}(\alpha(T,\mathcal{S})-(m-l))_+ > p, \end{aligned}$$

where the inequality follows from (3.18).

Hence, by the induction hypothesis, there exists $j\in N$ such that

$$|\{t\in\{1,2,...,p,l+1,...,m\}|j\in S_t\}| < p.$$

But then

$$\begin{aligned} |\{t\in\{1,2,...,m\}|j\in S_t\}| &\leq |\{t\in\{1,2,...,p,l+1,...,m\}|j\in S_t\}| + (l-p) \\ &< p+(l-p) = l. \end{aligned}$$

□

Theorem 3.5.2 *Let $\Gamma=(N,C,\{s_i\}_{i\in N},\{t_i\}_{i\in N},w)$ be a highway problem such that $C=(V,E)$ is a cycle. Then the highway game (N,c_Γ) is balanced.*

Proof. Let $\mathcal{B}=\{B_1,B_2,...,B_k\}$ be a balanced set with balancing weights $(\lambda_S)_{S\in\mathcal{B}}$. Since

$$\sum_{S\in\mathcal{B}}\lambda_S c_\Gamma(S) = \sum_{S\in\mathcal{B}}\lambda_S\left(w(E)-w(\bar{E}(R_S^*))\right)$$

and

$$c_\Gamma(N) = w(E) - w(\bar{E}(R^*_N)),$$

it is sufficient to show that

$$w(\bar{E}(R^*_N)) \geq (1 - \sum_{S\in\mathcal{B}} \lambda_S)w(E) + \sum_{S\in\mathcal{B}} \lambda_S w(\bar{E}(R^*_S)). \tag{3.19}$$

Note that right hand side of (3.19) can be rewritten as follows:

$$\begin{aligned}
(1 - \sum_{S\in\mathcal{B}} \lambda_S)w(E) + \sum_{S\in\mathcal{B}}&\lambda_S w(\bar{E}(R^*_S)) \\
&= (1 - \sum_{S\in\mathcal{B}} \lambda_S) \sum_{T\in\mathcal{R}^N} w(\bar{E}(T)) + \sum_{S\in\mathcal{B}} \lambda_S \sum_{T\in\mathcal{R}^N : T_{|S}=R^*_S} w(\bar{E}(T)) \\
&= (1 - \sum_{S\in\mathcal{B}} \lambda_S) \sum_{T\in\mathcal{R}^N} w(\bar{E}(T)) + \sum_{T\in\mathcal{R}^N} w(\bar{E}(T)) \sum_{S\in\mathcal{B}:T_{|S}=R^*_S} \lambda_S, \\
&= \sum_{T\in\mathcal{R}^N} \left(\sum_{S\in\mathcal{B}:T_{|S}=R^*_S} \lambda_S - (\sum_{S\in\mathcal{B}} \lambda_S - 1) \right) w(\bar{E}(T)),
\end{aligned}$$

where the first equality is obtained by using (iii) and (v) in Lemma 3.5.3.

Hence it is sufficient to prove that

$$w(\bar{E}(R^*_N)) \geq \sum_{T\in\mathcal{R}^N} \left(\sum_{S\in\mathcal{B}:T_{|S}=R^*_S} \lambda_S - (\sum_{S\in\mathcal{B}} \lambda_S - 1) \right) w(\bar{E}(T)). \tag{3.20}$$

Let $l \in \{1, 2, ...\}$ be such that $l\lambda_S \in \{1, 2, ...\}$ for all $S \in \mathcal{B}$ and, if $l' < l$, then there exists $S \in \mathcal{B}$ such that $l\lambda_S \notin \{1, 2, ...\}$. Let $\sum_{S\in\mathcal{B}} l\lambda_S = m$ and define a sequence of (duplicated) coalitions $\mathcal{S} = (S_1, S_2, ..., S_m)$ by $S_1 = S_2 = ... = S_{l\lambda_{B_1}} = B_1$, $S_{l\lambda_{B_1}+1}{=}S_{l\lambda_{B_1}+2} = ... = S_{l\lambda_{B_1}+l\lambda_{B_2}} = B_2$ and so on. Observe that for every path profile $R \in \mathcal{R}^N$,

$$\sum_{S\in\mathcal{B}:R_{|S}=R^*_S} l\lambda_S = \alpha(R, \mathcal{S}).$$

Then, it is sufficient to prove that

$$lw(\bar{E}(R^*_N)) \geq \sum_{T\in\mathcal{R}^N} \left(\sum_{S\in\mathcal{B}:T_{|S}=R^*_S} l\lambda_S - (\sum_{S\in\mathcal{B}} l\lambda_S - l) \right) w(\bar{E}(T))$$

or that

$$lw(\bar{E}(R^*_N)) \geq \sum_{T\in\mathcal{R}^N} (\alpha(T, \mathcal{S}) - (m - l))\, w(\bar{E}(T)). \tag{3.21}$$

Note that Lemma 3.5.4 implies that $\sum_{T\in\mathcal{R}^N} (\alpha(T, \mathcal{S}) - (m - l))_+ \leq l$. If not, there would exist a $j \in N$ such that $|\{t \in \{1, 2, ..., m\}|j \in S_t\}| < l$. However, since $\mathcal{B}$

is a balanced collection, it holds that

$$l = \sum_{S \in \mathcal{B}: j \in S} l\lambda_S = \left|\{t \in \{1, 2, ..., m\} | j \in S_t\}\right|.$$

Then

$$\begin{aligned} lw(\bar{E}(R_N^*)) &\geq \sum_{T \in \mathcal{R}^N} \left(\alpha(T, \mathcal{S}) - (m - l)\right)_+ w(\bar{E}(R_N^*)) \\ &\geq \sum_{T \in \mathcal{R}^N} \left(\alpha(T, \mathcal{S}) - (m - l)\right)_+ w(\bar{E}(T)) \\ &\geq \sum_{T \in \mathcal{R}^N} \left(\alpha(T, \mathcal{S}) - (m - l)\right) w(\bar{E}(T)), \end{aligned}$$

where the second inequality follows from the fact that $w(\bar{E}(R_N^*)) \geq w(\bar{E}(T))$ for every $T \in \mathcal{R}^N$. Hence, we can conclude that (N, c_Γ) is balanced. □

Finally, we show in Corollary 3.5.1 that the highway games induced by the weakly cyclic graphs are balanced.

Corollary 3.5.1 *Let $\Gamma = (N, G, \{s_i\}_{i \in N}, \{t_i\}_{i \in N}, w)$ be a highway problem such that G is a weakly cyclic graph. Then the highway game (N, c_Γ) is balanced.*

Proof. We know by Lemma 3.5.2 that a highway game on a weakly cyclic graph is equal to the sum of the sub-highway games with respect to each of its cycles and with respect to each bridge edge in the graph, i.e., $c_\Gamma(S) = \sum_{C \in \mathcal{C}(G)} c_{\Gamma^C}(S) + \sum_{e \in \mathcal{BE}(G)} c_{\Gamma^e}(S)$ for every $S \subset N$. Moreover, we know by Theorem 3.5.2 that highway games induced by cycles are balanced. Lastly, we know that highway games induced by trees are balanced. Pick $y^C \in Core(c_{\Gamma^C})$ for each $C \in \mathcal{C}(G)$ and $y^e \in Core(c_{\Gamma^e})$ for each $e \in \mathcal{BE}(G)$. Set $x = \sum_{C \in \mathcal{C}(G)} y^C + \sum_{e \in \mathcal{BE}(G)} y^e$. Clearly, $x \in Core(c_\Gamma)$. □

Chapter 4

Population Monotonic Path Schemes for Simple Games

In this chapter, which is based on Çiftçi et al. (2009b), we consider the formation of coalitions through binding bilateral agreements in voting/government formation situations.

In many real life contexts, ranging from the formation of pre/post-electoral coalitions of parties to the formation of mergers and partnerships between firms, coalitions form through a sequence of binding bilateral agreements. From among the numerous examples of such coalition formation processes, we may single out the recent mergers between the banks and between the consultancy firms that are observed in many countries and the Oslo agreements between Israel and its neighbors. An important characteristic of such coalition formation processes is the effect of the sequence of agreements on the future potential agreements. For a coalition formed through bilateral agreements may grow larger because the synergy/commitment obtained by a coalition may create new agreement opportunities which are profitable both for the members of the coalition and the agent which will join the coalition. Hence, the determination of the sequences of binding bilateral agreements which will result in the exploitation of the greatest possible amount of synergy is of both theoretical and practical importance.

The coalition formation processes which end up with the formation of the grand coalition deserve particular interest. Because, first of all, in many situations (e.g., situations of increasing returns to size), the grand coalition is the unique efficient coalition structure. Secondly, the formation of the grand coalition among agents which have common properties (e.g., the formation of the grand coalition among leftist parties) has been the focal point of many branches of social sciences.

In this chapter, we will focus on the formation of the grand coalition through binding bilateral agreements in voting/government formation situations. We aim to address two important questions in this context.

(i) Which voting situations allow for the formation of the grand coalition through binding bilateral agreements?

(ii) In these situations, which agreement sequences must be followed to form the grand coalition?

We will address these questions by modeling voting situations by simple transferable utility cooperative games. In voting situations, the voters' incentive to form coalitions arises from their will to increase their power to affect the outcome of the voting process. Modelling of these situations as simple transferable utility games allows us to predict the voters' power to affect the result of voting by using appropriate values for transferable utility games. Many values have been offered for simple games as appropriate measures of voting power and the two most widely used ones are the Shapley and Shubik (1954) and Banzhaf (1965) power indices. If we assume that each voter's voting power is predicted by such an appropriate index, then the sequences of binding bilateral agreements which result in the formation of the grand coalition boils down to the notion of *population monotonic path schemes*. Postponing a precise definition to the next section, a *path scheme* for a simple game is composed of a *path*, i.e., a sequence of coalitions that is formed through a sequence of binding bilateral agreements which result in the formation of the grand coalition and a *scheme*, i.e., a power index vector for each coalition in the path based on the associated subgame. A path scheme is called *population monotonic* if each player's index does not decrease as the path coalition grows. In this study, we focus on the Shapley-Shubik power index as an appropriate measure of voting power. Hence, the two questions that we address can be rephrased as

(i) Which simple games allow for population monotonic Shapley path schemes?

(ii) In these simple games, which Shapley path schemes are population monotonic?

It turns out that existence of veto players, i.e., a subgroup of voters whose unanimous agreement is necessary to pass a decision, is required for the existence of population monotonic Shapley path schemes and vice versa. Moreover, a Shapley path scheme is population monotonic if and only if the first winning coalition that is formed along the path contains every minimal winning coalition of the game. We also show that each Shapley path scheme of a game is population monotonic if and only if the set of veto players of the game is a winning coalition. We further show how to extend these results to the probabilistic values, generalizations of the Shapley value introduced by Weber (1988).

The notion of population monotonic (Shapley) path schemes is introduced by Cruijssen et al. (2005). This study analyzes *insinking* (the antonym of outsourcing) situations in logistics and the transportation sector. In these sectors, shippers often outsource their transportation activities to a logistics service provider of their choice. Cruijssen et al. (2005) proposes an insinking procedure in which the logistics service provider initiates the shift of logistics activities instead of waiting for the shippers to outsource their activities. To obtain the greatest possible amount of gains, the service provider has to find an effective way of proposing offers to shippers through which it can acquire the involvement of each shipper. At this point, Cruijssen et al. (2005) proposes a sequence of binding bilateral agreements arguing that compared to the simultaneous comprehensive agreements, by following an appropriate sequence of binding bilateral agreements, the service provider can attract new customers to the project by using the level of synergy and commitment already attained in the sequence.

Our study in particular provides an alternative prediction of what kind of coalitions form in voting situations which differs from the mainstream prediction of Riker (1962). Riker (1962) predicts that only minimal winning coalitions will form in equilibrium. This idea has been the conclusion of many studies in the general coalition formation literature based on the seminal noncooperative bargaining approach of Baron and Ferejohn (1989) and also the studies which analyze coalition formation in voting situations that are modeled by simple TU-games like Shenoy (1979). However, the empirical data on government/coalition formation shows that among all coalitions formed after the second world war in European democracies only a third of them is minimal winning (Laver and Schofield, 1990). Our current study shows that a wide spectrum of coalitions including the minimal winning ones can form as a result of binding bilateral agreements providing an alternative point of view for the analysis and the explanation of the data.

Population monotonic path schemes (PMPS) are in the same spirit as population monotonic allocation schemes (PMAS) for cooperative games, introduced by Sprumont (1990) and further analyzed in e.g., Norde and Reijnierse (2002) and Slikker et al. (2003). An allocation scheme for a cooperative game specifies how to distribute the worth of every coalition among its members and it is called population monotonic if the share of any player does not decrease as the coalition he belongs to grows larger. Clearly, also a PMAS's main concern is to ensure that no player is worse off with additional cooperation between players. However, a PMAS compares the allocations assigned to a coalition of players with every sub-coalition's allocation while a PMPS restricts the comparison to the allocations of path coalitions that are formed previously. In fact, the existence of a PMPS is a weaker condition for a TU-game than the existence of a PMAS since every path scheme induced by a PMAS is population monotonic. Another difference between the two notions is that each allocation provided by a PMAS has to belong to the core of the associated subgame. However, this may not be the case for a PMPS as we exemplify in our study.

The outline of the chapter is as follows. Section 4.1 recalls some well-known concepts on simple TU-games. Section 4.2 introduces Shapley path schemes and presents the main results regarding the characterization of population monotonic Shapley path schemes of simple games. Section 4.3 discusses extensions of the results to other probabilistic values.

4.1 Preliminaries

A TU-game $v \in \mathcal{G}^N$ is called *simple* if v is monotonic, $v(S) \in \{0,1\}$ for every $S \in 2^N$ and $v(N) = 1$. We denote the set of simple TU-games with player set N by $\mathcal{S}^N$. Given $v \in \mathcal{S}^N$, a coalition $S \in 2^N$ is called a *winning coalition* if $v(S) = 1$ and is called a *losing coalition* if $v(S) = 0$. A winning coalition S is called *minimal winning* if there does not exist a coalition $T \subsetneq S$ which is winning. Every simple game v is characterized by its set of minimal winning coalitions, $MWC(v)$. A player $i \in N$ is a *veto player* in $v \in \mathcal{S}^N$ if $S \subset N$, $v(S) = 1$ implies that $i \in S$. The set of veto players of v is denoted by $veto(v)$. It is readily verified that a simple game v is balanced if and only if $veto(v) \neq \emptyset$.

Voting or decision making situations in committees like parliaments can easily be modeled into the framework of simple games by representing the coalitions which possesses the necessary power to pass a decision as the winning coalitions of the game. This model enables the employment of values for simple games to measure the parties' power to affect the outcome of the voting situations at hand. Many values have been offered for simple games and studied in the literature as appropriate measures of decisional power, i.e., as *power indices*. We will shortly review the Shapley and Shubik (1954) power index that arises from the Shapley value.

Shapley and Shubik (1954) proposed to use the Shapley value as a power index for voting situations in committees. For a simple game $v \in \mathcal{S}^N$ the Shapley-Shubik index Φ assigns to player $i \in N$

$$\Phi_i(v) = \sum_{\{S \subset N\setminus\{i\} | v(S)=0, v(S\cup\{i\})=1\}} \frac{|S|!(|N|-|S|-1)!}{|N|!}. \tag{4.1}$$

The value assigned to each voter can be interpreted by using the sequential probabilistic interpretation of the Shapley value which stems from a procedure to form the grand coalition (which is described also by Shapley, 1953) that yields the Shapley value of the game as an expected payoff of each player. In this procedure, the grand coalition N is formed by introducing the players one by one and each player is assigned the marginal contribution to the worth of the coalition formed when she joins the set of her predecessors. Hence, the value assigned by Shapley-Shubik index is the probability of turning the coalition of predecessors from losing to winning when the order of arrival of players is random and all orders are equally likely. For further discussion of the importance of the Shapley value as an estimator of political power and several examples of its applications, the reader is referred to Straffin (1994) and Winter (2002). Lastly, we know (for example by Derks and Haller (1999) that the Shapley value satisfies the null player out property.

4.2 Population Monotonic Shapley Path Schemes

In this section we introduce and analyze the Shapley path schemes of simple games.

Let $v \in \mathcal{G}^N$. A *path* consists of a sequence $\mathbb{S} = \{S_1, S_2, ..., S_{|N|}\}$ of coalitions such that $|S_k| = k$ for all $k \in \{1, ..., |N|\}$ and $S_m \subset S_{m+1}$ for all $m \in \{1, ..., |N| - 1\}$. A *path scheme* specifies how to distribute the worth of every coalition on the path among its members. Formally, a *path scheme* $(\mathbb{S}, (x^S)_{S\in\mathbb{S}})$ for v consists of a path $\mathbb{S}$ and a vector $(x^S)_{S\in\mathbb{S}}$ such that

$$\sum_{i\in S} x_i^S = v(S)$$

for every coalition $S \in \mathbb{S}$.

A path scheme $(\mathbb{S}, (x^S)_{S\in\mathbb{S}})$ for $v \in \mathcal{G}^N$ is called *population monotonic* if it satisfies the following conditions:

- $x_i^S \geq v(\{i\})$ for all $S \in \mathbb{S}$ and $i \in S$. (individual rationality)
- $x_i^S \geq x_i^T$ for every $S, T \in \mathbb{S}$ such that $T \subset S$ and $i \in T$. (monotonicity)

A path scheme in which the Shapley value is used as an allocation vector is called a *Shapley path scheme*. Clearly, the Shapley allocation scheme of a TU-game is population monotonic if and only if all Shapley path schemes of the game are population monotonic. We will illustrate the notion of Shapley path schemes and their properties in the following example.

Example 4.2.1 Let $N = \{1, 2, 3\}$ and consider the simple game $v \in \mathcal{S}^N$ with $MWC(v) = \{\{1, 2\}, \{2, 3\}\}$. The Shapley allocation scheme of v is provided in Table 4.1 below.

Coalition	Player 1	Player 2	Player 3
{1}	0	-	-
{2}	-	0	-
{3}	-	-	0
{1,2}	$\frac{1}{2}$	$\frac{1}{2}$	-
{1,3}	0	-	0
{2,3}	-	$\frac{1}{2}$	$\frac{1}{2}$
N	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$

Table 4.1: The Shapley allocation scheme of v in Example 4.2.1

It can easily be observed that the Shapley allocation scheme of v is not population monotonic but that there are exactly two population monotonic Shapley path schemes on the paths $\{\{1\},\{1,3\},N\}$ and $\{\{3\},\{1,3\},N\}$, respectively.

Observe also that the game v has a unique core allocation, $(0,1,0)$ different from the Shapley value of v. So, in particular, the allocation prescribed by a (Shapley) PMPS may not belong to the core of the associated subgame.

◇

We will begin with presenting a preliminary result which is useful in understanding the structure of population monotonic Shapley path schemes of simple games.

Lemma 4.2.1 *Given a simple game $v \in \mathcal{S}^N$, let $\mathbb{S} = \{S_1, S_2, ..., S_{|N|}\}$ be a path of coalitions such that $S_m = \{i_1, ..., i_m\}$ for every $m \in \{1, ..., |N|\}$. Assume that the first winning coalition along the path $\mathbb{S}$ is S_k, i.e., $v(S_1) = ... = v(S_{k-1}) = 0$ and $v(S_k) = 1$. If the Shapley path scheme $(\mathbb{S}, (\Phi(v_{|S}))_{S\in\mathbb{S}})$ is population monotonic, then the following must hold:*

(R1) $\Phi_{i_m}(v_{|S_p}) = 0$, for all $m \in \{k+1, ..., |N|\}$ and for all $p \in \{m, ..., |N|\}$.
(R2) $\Phi_i(v_{|S_k}) = \Phi_i(v_{|S_p})$, for all $p \in \{k+1, ..., |N|\}$ and for all $i \in S_k$.
(R3) $MWC(v_{|S_k}) = MWC(v)$.

Proof. (R1) and (R2). On the one hand $\sum_{i\in S_p} \Phi_{i_m}(v_{|S_p}) = 1$ for all $p \in \{k, ..., |N|\}$ by the efficiency of the Shapley value. On the other hand, by the population monotonicity of $(\mathbb{S}, (\Phi(v_{|S}))_{S\in\mathbb{S}})$, $\Phi_i(v_{|S_p}) \geq \Phi_i(v_{|S_k})$ for every $p \in \{k+1, ..., |N|\}$ and $\Phi_{i_m}(v_{|S_p}) \geq 0$, for all $m \in \{k+1, ..., |N|\}$ and for all $p \in \{m, ..., |N|\}$. Hence $\Phi_{i_m}(v_{|S_p}) = 0$ for all $m \in \{k+1, ..., |N|\}$ and for all $p \in \{m, ..., |N|\}$ and $\Phi_i(v_{|S_k}) = \Phi_i(v_{|S_p})$ for all $p \in \{k+1, ..., |N|\}$ and for all $i \in S_k$.

(R3). Suppose on the contrary that $MWC(v_{|S_k}) \neq MWC(v)$. Then there exists a $T \in MWC(v)$ such that $T\backslash S_k \neq \emptyset$. But then $\Phi_j(v) > 0$ for every $j \in T\backslash S_k$, a contradiction with (R1). □

We now provide a characterization of the family of simple games which allow for population monotonic Shapley path schemes.

Theorem 4.2.1 *Let $v \in \mathcal{S}^N$. Then v has a population monotonic Shapley path scheme if and only if v is balanced.*

Proof. Let $v \in \mathcal{S}^N$ have a population monotonic Shapley path scheme. Also let $(\mathbb{S}, (\Phi(v_{|S}))_{S\in\mathbb{S}})$ be a population monotonic Shapley path scheme for v such that $\mathbb{S} = \{S_1, S_2, ..., S_{|N|}\}$ and $S_m = \{i_1, ..., i_m\}$ for every $m \in \{1, ..., |N|\}$. Assume that the first winning coalition along the path $\mathbb{S}$ is S_k. Obviously $i_k \in veto(v_{|S_k})$ and hence $veto(v_{|S_k}) \neq \emptyset$. Moreover, we know by (R3) in Lemma 4.2.1 that $MWC(v_{|S_k}) = MWC(v)$. Hence, $veto(v_{|S_k}) = veto(v)$ and v is balanced.

Now, assume that v is balanced. Then, $veto(v) \neq \emptyset$. Let $i \in veto(v)$ and consider a path $\mathbb{S} = \{S_1, S_2, ..., S_{|N|}\}$ with $S_{|N|-1} = N\backslash\{i\}$. We know that $S_{|N|-1} = N\backslash\{i\}$ is a losing coalition. Then $v_{|N\backslash\{i\}}$ is a null game and hence $\Phi_j(v_{|S_t}) = 0$ for all $t \in \{1, ..., |N| - 1\}$ and $j \in S_t$. Also, $\Phi_j(v) \geq 0$ for all $j \in N$, since v is monotonic. So, the Shapley path scheme $(\mathbb{S}, (\Phi(v_{|S}))_{S\in\mathbb{S}})$ is population monotonic. □

Theorem 4.2.1 reveals that, in the class of simple games, the existence of veto players (or, equivalently, a nonempty core) is a must for the existence of population monotonic Shapley path schemes and vice versa. We can interpret this result as follows. When a winning coalition is formed through a sequence of binding bilateral agreements, we know that the restriction of the TU-game to this coalition has veto players, that is, in this winning coalition, there is a subgroup of agents whose unanimous agreement/involvement is necessary to pass a decision. We also know that the formation of the grand coalition starting from this winning coalition via binding bilateral agreements requires the remaining players to be null players. But, this in turn implies that the veto players of the winning coalition are in fact the veto players of the whole game, i.e., the game is balanced.

Next we turn to our second question of which Shapley path schemes are population monotonic. We will show in the following theorem that the requirement that the first winning coalition along a path has to include all minimum winning coalitions of the game is both necessary and sufficient for the population monotonicity of the corresponding Shapley path scheme.

Theorem 4.2.2 *Let $v \in \mathcal{S}^N$ be balanced. A Shapley path scheme $(\mathbb{S}, (\Phi(v_{|S}))_{S\in\mathbb{S}})$ is population monotonic if and only if the first winning coalition along $\mathbb{S}$ contains every minimal winning coalition of v.*

Proof. Let $(\mathbb{S}, (\Phi(v_{|S}))_{S\in\mathbb{S}})$ be a population monotonic Shapley path scheme for v and assume that the first winning coalition along the path $\mathbb{S}$ is S_k. We already know by (R3) in Lemma 4.2.1 that $MWC(v_{|S_k}) = MWC(v)$. Then, clearly, S_k contains every minimal winning coalition of v.

Let $\mathbb{S}$ be a path of coalitions with $S_m = \{i_1, ..., i_m\}$ for every $m \in \{1, ..., |N|\}$. Assume that the first winning coalition along the path $\mathbb{S}$ is S_k ($k \in \{1, ..., |N|\}$) and S_k contains every minimal winning coalition of v. Now, $\Phi_j(v_{|S_t}) = 0$ for all $t \in \{1, ..., k-1\}$ and $j \in S_t$ since S_{k-1} is a losing coalition. Also, $\Phi_i(v_{|S_k}) \geq 0$ for all $i \in S_k$ since v is monotonic. We know that each player i_m ($m \in \{k+1, ..., |N|\}$) is a null player in $v_{|S_p}$ ($p \in \{m, ..., |N|\}$) since S_k contains every minimal winning coalition of v. Then, firstly, $\Phi_{i_m}(v_{|S_p}) = 0$ for all $m \in \{k+1, ..., |N|\}$ and for all $p \in \{m, ..., |N|\}$ and secondly, one can easily show that $\Phi_i(v_{|S_k}) = \Phi_i(v_{|S_{k+1}}) = ... = \Phi_i(v)$ for all $i \in S_k$ by applying the null player out property recursively. So, we conclude that the Shapley path scheme $(\mathbb{S}, (\Phi(v_{|S}))_{S\in\mathbb{S}})$ is population monotonic. $\square$

In the light of Theorem 4.2.2, we can answer one other important question in this context: For which simple games all Shapley path schemes are population monotonic, i.e., which simple games have a population monotonic Shapley allocation scheme?

Theorem 4.2.3 *Let $v \in \mathcal{S}^N$ be a simple game. Then the following statements are equivalent:*

(i) All Shapley path schemes of v are population monotonic.

(ii) The set of veto players of v is a winning coalition.

(iii) The game v is convex.

(iv) The Shapley allocation scheme of v is population monotonic.

Proof. $(i) \rightarrow (ii)$ Assume that all Shapley path schemes of v are population monotonic. Suppose that $veto(v)$ is losing. Then there exists a minimum winning coalition $S = \{i_1, ..., i_m\}$ with $m \in \{1, ..., |N| - 1\}$. We know that $\Phi_i(v_{|S}) = \frac{1}{m}$ for every $i \in S$ since S is a minimal winning coalition. Pick a path of coalitions $\mathbb{S} = \{S_1, S_2, ..., S_{|N|}\}$ with $S_m = S$. The Shapley path scheme $(\mathbb{S}, (\Phi(v_{|S}))_{S\in\mathbb{S}})$ is population monotonic by assumption. Consequently, $\Phi_i(v) = \frac{1}{m}$ for every $i \in S$. Observe that there exists $i^* \in S$ such that $i^* \notin veto(v)$ since S is a minimal winning coalition and $veto(v)$ is losing. Then, there exists another minimal winning coalition $T \subsetneq N$ such that $i^* \notin T$. Pick a path of coalitions $\mathbb{S}' = \{S'_1, S'_2, ..., S'_{|N|}\}$ with $S'_{|T|} = T$. Now, the Shapley path scheme $(\mathbb{S}', (\Phi(v_{|S}))_{S\in\mathbb{S}'})$ is also population monotonic by assumption. Then, $(R1)$ implies that $\Phi_{i^*}(v) = 0$ since $i^* \notin T$, a contradiction with $\Phi_{i^*}(v) = \frac{1}{m}$ as derived earlier.

$(ii) \rightarrow (iii)$ Let $v \in \mathcal{S}^N$ be such that $veto(v)$ is a winning coalition. Then, all players in $N \backslash veto(v)$ are null players in v. Hence, v is the unanimity game on $veto(v)$ and is convex.

$(iii) \rightarrow (iv)$ See Sprumont (1990), Corollary 2.

$(iv) \rightarrow (i)$ Obvious. □

Theorem 4.2.3 reveals that, in the class of simple games, the existence of a winning veto player set is both necessary and sufficient for the existence of a population monotonic Shapley allocation scheme. This result can be interpreted by making use of our results on population monotonic Shapley path schemes as follows. We know that the existence of a population monotonic Shapley allocation scheme implies the population monotonicity of each Shapley path scheme of the game and vice versa. Then, by Theorem 4.2.2, the existence of a population monotonic Shapley allocation scheme requires the first winning coalition along each path to include all minimum winning coalitions of the game. But this is possible only when the game has a unique minimum winning coalition, i.e., when the set of veto players is winning.

4.3 Extensions to Probabilistic Values

Probabilistic values, introduced and characterized by Weber (1988), are generalizations of the Shapley value for finite TU-games. These values keep one essential feature of the Shapley value, they assign each player an average of his marginal contributions. They, however, fail to satisfy either the efficiency or anonymity property. In fact, the Shapley value is the unique probabilistic value satisfying both anonymity and efficiency. Probabilistic values can be classified into two groups: Quasi-values which are efficient probabilistic values and Semi-values, the probabilistic values which satisfy anonymity (see Weber, 1988). We refer to Monderer and Samet (2002) for a detailed discussion of probabilistic values.

Probabilistic values are formally defined as follows. Given N and $i \in N$, let P_N^i denote the set of probability distributions on $2^{N \backslash \{i\}}$, the family of coalitions not containing i. A value F (defined on $\mathcal{G}^N$) is called a *probabilistic value* (Weber, 1988) if for every $v \in \mathcal{G}^N$ and $i \in N$

$$F_i(v) = \sum_{T \subset N \backslash \{i\}} p^i(T) \left(v(T \cup \{i\}) - v(T) \right), \tag{4.2}$$

for some $p^i \in P^i_N$ and for all $i \in N$. Here $p^i \in P^i_N$ can be interpreted as the player's subjective evaluation of the probability of joining different coalitions. For example, the probabilistic value which is defined by $p^i(T) = \frac{1}{|N|}\binom{|N|-1}{|T|}^{-1}$ for all $i \in N$ is the Shapley value.

In the following two subsections we will discuss the extensions of the results obtained for the Shapley value on quasi-values and on semi-values, respectively.

4.3.1 Population monotonic path schemes of quasi-values

Let $\mathcal{P}(\Pi(N))$ denote the set of probability distributions on the set of permutations of the player set N. Given $i \in N$ and $S \in 2^{N\setminus\{i\}}$, we will denote by $\Pi^{S,i}(N)$ the set

$$\{\tau \in \Pi(N) | \tau(j) < \tau(i) \text{ if and only if } j \in S\}.$$

If we think of a permutation $\tau \in \Pi(N)$ as the order in which players enter the game, then $\Pi^{S,i}(N)$ stands for the set of orders in which exactly all members of S enter the game before player i enters.

The following characterization of efficient probabilistic values is provided by Weber (1988).

Theorem 4.3.1 *(Weber, 1988) Let F be a probabilistic value as given in (4.2) defined by $p = \{p^i\}_{i\in N}$ with $p^i \in P^i_N$ for every $i \in N$. Then F is efficient if and only if there exists $b \in \mathcal{P}(\Pi(N))$ such that*

$$p^i(S) = \sum_{\tau \in \Pi^{S,i}(N)} b(\tau) \tag{4.3}$$

for every $i \in N$ and $S \in 2^{N\setminus\{i\}}$.

Observe that probabilistic values are originally defined for a fixed player set. However, our analysis requires the values to be defined on every subset of the player set under consideration. Because, for every simple game, we want to be able to compare the payoffs assigned by a value to the players at every subgame of the game. We now extend probabilistic values in such a way that the players' subjective evaluation of the probability of joining different coalitions will be consistent in the sense defined below. For this aim we will define the *restrictions* of a probabilistic value to subgames.

Let $F : \mathcal{G}^N \to \mathbb{R}^N$ be a probabilistic value defined by $\{p^i_N\}_{i\in N}$ where $p^i_N \in P^i_N$ for every $i \in N$. For each $S \subset N$, the *restriction* of F to $\mathcal{G}^S$ is denoted by F_S and for each player $i \in S$, his restricted evaluations $p^i_S \in P^i_S$ are constructed by using the following consistency condition.

$$p^i_S(T) = \sum_{T' \subset N\setminus S} p^i_N(T \cup T'), \tag{4.4}$$

for all $T \subset S\backslash\{i\}$.

The consistency condition can be interpreted by using the sequential probabilistic interpretation of the Shapley value (and of the probabilistic values of course). If we let the players enter a room one by one and assign each player the marginal contribution created by her, then p_N^i represents player i's assessment of the probability of different coalitions to be the set of predecessors of her. Then if one player j leaves N, it is natural to expect that every player $i \in N\backslash\{j\}$ will update her assessment of joining a coalition $T \subset N\backslash\{i,j\}$ by merging her previous assessments of T and of $T \cup \{j\}$[1].

We illustrate the notion of the restriction of a probabilistic value in the following example.

Example 4.3.1 Let F be a probabilistic value on $N = \{1,2,3\}$. Assume that F is defined by the following subjective evaluations of players.

$$\begin{array}{l} p_N^1(\{2,3\}) = \frac{5}{16},\ p_N^1(\{2\}) = \frac{1}{16},\ p_N^1(\{3\}) = \frac{4}{16} \text{ and } p_N^1(\emptyset) = \frac{6}{16}. \\ p_N^2(\{1,3\}) = \frac{8}{16},\ p_N^2(\{1\}) = \frac{2}{16},\ p_N^2(\{3\}) = \frac{4}{16} \text{ and } p_N^2(\emptyset) = \frac{2}{16}. \\ p_N^3(\{1,2\}) = \frac{3}{16},\ p_N^3(\{1\}) = \frac{4}{16},\ p_N^3(\{2\}) = \frac{1}{16} \text{ and } p_N^3(\emptyset) = \frac{8}{16}. \end{array}$$

F satisfies (4.3) by taking the following probability distribution on the set of permutations on the player set:

$$b(123) = \tfrac{2}{16}, b(132) = \tfrac{4}{16}, b(213) = \tfrac{1}{16}, b(231) = \tfrac{1}{16}, b(312) = \tfrac{4}{16}, \text{ and } b(321) = \tfrac{4}{16}.$$

Hence F is efficient.
Now consider $S = \{1,2\}$. According to (4.4), the restriction F_S is defined by:

$$\begin{array}{l} p_S^1(\{2\}) = \frac{3}{8} = p_N^1(\{2\}) + p_N^1(\{2,3\}) \text{ and } p_S^1(\emptyset) = \frac{5}{8} = p_N^1(\emptyset) + p_N^1(\{3\}). \\ p_S^2(\{1\}) = \frac{5}{8} = p_N^2(\{1\}) + p_N^2(\{1,3\}) \text{ and } p_S^2(\emptyset) = \frac{3}{8} = p_N^2(\emptyset) + p_N^2(\{3\}). \end{array}$$

Notice that F_S can be described via (4.3) by taking:

$$b(12) = \tfrac{5}{8} \text{ and } b(21) = \tfrac{3}{8}.$$

So F_S is an efficient probabilistic value on $\mathcal{G}^S$. ◇

In the previous example, we have shown that the specific restriction under consideration is again an efficient probabilistic value. Indeed, every restriction of an efficient probabilistic value is an efficient probabilistic value for the corresponding subgame as shown in the following proposition.

[1] As discussed in the following subsection every semi-value readily satisfies our consistency condition (4.4).

Proposition 4.3.1 *Let F be an efficient probabilistic value defined by $\{p_N^i\}_{i\in N}$ where $p_N^i \in P_N^i$ for every $i \in N$. Then, F_S is an efficient probabilistic value for every $S \subset N$, $S \neq \emptyset$.*

Proof. By Theorem 4.3.1 there exists $b \in \mathcal{P}(\Pi(N))$ such that $p_N^i(T) = \sum_{\tau\in\Pi^{T,i}(N)} b(\tau)$ for every $i \in N$ and $T \in 2^{N\setminus\{i\}}$. Take $S \subset N, S \neq \emptyset$. Given $\tau \in \Pi(N)$, $\tau_{|S}$ denotes the restriction of τ to S, i.e., $\tau_{|S} = \pi$ for some $\pi \in \Pi(S)$ with $\pi(i) < \pi(j)$ if and only if $\tau(i) < \tau(j)$, for all $i, j \in S$. We can induce a probability distribution c on $\Pi(S)$ from b as follows.

$$c(\pi) = \sum_{\tau\in\Pi(N):\tau_{|S}=\pi} b(\tau), \text{ for all } \pi \in \Pi(S). \tag{4.5}$$

Let F_S be defined by $\{p_S^i\}_{i\in S}$ as determined by (4.4). Pick $i \in S$ and $T \subset S\setminus\{i\}$. Obviously,

$$\bigcup_{T'\subset N\setminus S} \Pi^{(T\cup T'),i}(N) = \bigcup_{\pi\in\Pi^{T,i}(S)} \{\tau \in \Pi(N) | \tau_{|S} = \pi\} \tag{4.6}$$

Notice that

$$\Pi^{(T\cup T'),i}(N) \cap \Pi^{(T\cup T''),i}(N) = \emptyset \text{ for every } T', T'' \subset N\setminus S \text{ with } T' \neq T''$$

and

$$\{\tau \in \Pi(N) | \tau_{|S} = \pi\} \cap \{\tau \in \Pi(N) | \tau_{|S} = \pi'\} = \emptyset \text{ for every } \pi, \pi' \in \Pi^{T,i}(S) \text{ with } \pi \neq \pi'.$$

Then,

$$\begin{aligned} p_S^i(T) &= \sum_{T'\subset N\setminus S} p_N^i(T \cup T') = \sum_{T'\subset N\setminus S} \sum_{\tau\in\Pi^{(T\cup T'),i}(N)} b(\tau) \\ &= \sum_{\pi\in\Pi^{T,i}(S)} \sum_{\tau\in\Pi(N):\tau_{|S}=\pi} b(\tau) = \sum_{\pi\in\Pi^{T,i}(S)} c(\pi) \end{aligned}$$

where the first equality follows from (4.4) and the last but one equality follows from (4.6) and the remarks below it. Then, Theorem 4.3.1 implies that F_S is an efficient probabilistic value on $\mathcal{G}^S$. □

Having defined the restrictions of probabilistic values, we can now illustrate the path schemes associated with these values in the following example.

Example 4.3.2 Consider the probabilistic value F defined in Example 4.3.1 and let $v \in \mathcal{S}^N$ with $N = \{1, 2, 3\}$ be defined by $MWC(v) = \{\{1, 2\}, \{2, 3\}\}$. From Table 4.2 it can easily be observed that this balanced game has two population monotonic F-path schemes related to the paths {{1},{1,3},N} and {{3},{1,3},N}. ◇

Coalition	Player 1	Player 2	Player 3
{1}	0	-	-
{2}	-	0	-
{3}	-	-	0
{1,2}	$\frac{3}{8}$	$\frac{5}{8}$	-
{1,3}	0	-	0
{2,3}	-	$\frac{6}{8}$	$\frac{2}{8}$
N	$\frac{1}{16}$	$\frac{14}{16}$	$\frac{1}{16}$

Table 4.2: The restrictions of F for v and its subgames in Example 4.3.2

The following theorem states that the results for population monotonic Shapley path schemes in fact can be extended to all efficient probabilistic values which are defined by strictly positive subjective evaluations of joining different coalitions for each player.

Theorem 4.3.2 *Let* $F : \mathcal{G}^N \to \mathbb{R}^N$ *be an efficient probabilistic value defined by* $\{p^i_N\}_{i \in N}$ *with* $p^i_N > 0$ *for all* $i \in N$. *Then*

(i) A simple game $v \in \mathcal{S}^N$ *has a population monotonic* F*-path scheme if and only if* v *is balanced.*

(ii) Let v *be balanced. Then an* F*-path scheme* $(\mathbb{S}, (F_S(v_{|S}))_{S \in \mathbb{S}})$ *is population monotonic if and only if the first winning coalition along* $\mathbb{S}$ *contains every minimal winning coalition of* v.

(iii) Let $v \in \mathcal{S}^N$ *be a simple game. Then the following statements are equivalent:*

(a) All F*-path schemes of* v *are population monotonic.*

(b) The set of veto players of v *is a winning coalition.*

(c) The game v *is convex.*

(d) The F*-allocation scheme of* v, $(F_S(v_{|S}))_{S \in 2^N \setminus \{\emptyset\}})$ *is population monotonic.*

The proof of Theorem 4.3.2 is similar to the proofs of Theorems 4.2.1, 4.2.2 and 4.2.3, respectively and is therefore omitted.

It is important at this point to observe that if for an efficient probabilistic value F, $p^i_N(S) = 0$ for some $i \in N$ and $S \in 2^{N \setminus \{i\}}$, then an unbalanced simple game may have population monotonic F-path schemes. This is illustrated in Example 4.3.3.

Example 4.3.3 Let $N = \{1,2,3\}$. Let F be the efficient probabilistic value determined by

$$p_N^1(S) = \frac{1}{4} \text{ for all } S \subset N\backslash\{1\}; p_N^2(S) = \frac{1}{4} \text{ for all } S \subset N\backslash\{2\} \text{ and}$$
$$p_N^3(\{1,2\}) = p_N^3(\emptyset) = \frac{1}{2}, p_N^3(\{1\}) = p_N^3(\{2\}) = 0.$$

Consider $v \in \mathcal{S}^N$ defined by $MWC(v) = \{\{1,2\},\{1,3\},\{2,3\}\}$. Clearly $veto(v) = \emptyset$. But, v has population monotonic F-path schemes related to the paths $\{\{1\},\{1,2\},N\}$ and $\{\{2\},\{1,2\},N\}$ as can be seen in Table 4.3. $\diamond$

Coalition	Player 1	Player 2	Player 3
{1}	0	-	-
{2}	-	0	-
{3}	-	-	0
{1,2}	$\frac{1}{2}$	$\frac{1}{2}$	-
{1,3}	$\frac{1}{2}$	-	$\frac{1}{2}$
{2,3}	-	$\frac{1}{2}$	$\frac{1}{2}$
N	$\frac{1}{2}$	$\frac{1}{2}$	0

Table 4.3: The restrictions of F for v and its subgames in Example 4.3.3

4.3.2 Population monotonic path schemes of semi-values

In this section, we will focus on one particular, well-known semi-value, the Banzhaf value (Banzhaf, 1965). Given $v \in \mathcal{G}^N$, the Banzhaf value β assigns to player $i \in N$

$$\beta_i(v) = \sum_{S \subset N\backslash\{i\}} \frac{1}{2^{|N|-1}} \left(v(S \cup \{i\}) - v(S)\right).$$

It can easily be observed that the Banzhaf value is defined for every finite player set, and in particular also for all subgames of a specific game. In fact, every semi-value is defined for every finite player set, and hence for all subgames of a specific game. Moreover, the restriction of a semi-value obtained by using the consistency condition (4.4) boils down to the definition of the same semi-value for the corresponding subgame. This can be readily verified from the characterization of semi-values on TU-games with finite support provided by Dubey et al. (1981). Dubey et al. (1981) show that, for every semi value, the players' underlying subjective evaluations $\{p_N^i\}_{i \in N}$, depend only on the cardinalities of S and N and hence every semi-value is defined for all subgames of a particular game. It can also easily be checked that the restriction of a semi-value by using the consistency condition (4.4) like we did for quasi-values. But, one can check that every semi-value satisfies the consistency condition (4.4) by making use of the characterization of semi-values on TU-games with finite support provided by Dubey et al. (1981).

For population monotonic Banzhaf path schemes the situation essentially differs from the population monotonic Shapley path schemes. This is illustrated in Examples 4.3.4 and 4.3.5.

Example 4.3.4 Let $N = \{1, 2, 3\}$ and consider the simple game $v \in \mathcal{S}^N$ defined by $MWC(v) = \{\{1, 2\}, \{1, 3\}, \{2, 3\}\}$. The Banzhaf value of v and its subgames are provided in the Table 4.4.

Coalition	Player 1	Player 2	Player 3
{1}	0	-	-
{2}	-	0	-
{3}	-	-	0
{1,2}	$\frac{1}{2}$	$\frac{1}{2}$	-
{1,3}	$\frac{1}{2}$	-	$\frac{1}{2}$
{2,3}	-	$\frac{1}{2}$	$\frac{1}{2}$
N	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$

Table 4.4: The Banzhaf allocation scheme of v and in Example 4.3.4

Notice that v is not balanced since $veto(v) = \emptyset$ but that every Banzhaf path scheme of v is population monotonic. $\diamond$

Example 4.3.5 Let $N = \{1, 2, 3, 4\}$ and consider the simple game $v \in \mathcal{S}^N$ defined by $MWC(v) = \{\{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}\}$. Clearly, v is balanced. The Banzhaf value of v and its subgames are provided in Table 4.5. The Banzhaf values of the subgames corresponding to losing coalitions are omitted. Then, every

Coalition	Player 1	Player 2	Player 3	Player 4
{1,2,3}	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	-
{1,2,4}	$\frac{1}{4}$	$\frac{1}{4}$	-	$\frac{1}{4}$
{1,3,4}	$\frac{1}{4}$	$\frac{1}{4}$	-	$\frac{1}{4}$
N	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$

Table 4.5: The Banzhaf allocation scheme of v in Example 4.3.5

Banzhaf path scheme is population monotonic although the set of veto players of v

is a losing coalition. Secondly, there are path schemes of v, like the one related to path $\{\{1\},\{1,2\},\{1,2,3\},\{1,2,3,4\}\}$, which are population monotonic but the first winning coalition along these paths does not contain the union of minimal winning coalitions v. $\diamond$

The results for population monotonic Shapley path schemes can be extended only partly to the Banzhaf value. This is reflected in Theorem 4.3.3.

Theorem 4.3.3 *Let $v \in \mathcal{S}^N$ be balanced.*

(1) If the first winning coalition along a path $\mathbb{S}$ contains every minimal winning coalition of v, then the Banzhaf-path scheme $(\mathbb{S}, (\beta(v_{|S}))_{S\in\mathbb{S}})$ is population monotonic .

(2) If the set of veto players of v is a winning coalition, then all Banzhaf-path schemes of v are population monotonic.

The proof of Theorem 4.3.3 is similar to the corresponding parts of the proofs of Theorems 4.2.2 and 4.2.3, respectively and is therefore omitted.

By making use of the characterization of semi-values provided by Dubey et al. (1981), one can show that Theorem 4.3.3 can be extended to every semi-value.

Bibliography

Ahmadi, J. H., Ahmadi, R. H., Dasu, S., and Tang, C. S. (1992). Batching and scheduling jobs on batch and discrete processors. *Operations Research*, 40:750–763.

Allahverdi, A., Cheng, T. C. E., and Kovalyov, M. Y. (2008). A survey of scheduling problems with setup times or costs. *European Journal of Operational Research*, 187:985–1032.

Allahverdi, A., Gupta, J. N. D., and Aldowaisan, T. (1999). A review of scheduling research involving setup considerations. *OMEGA, The International Journal of Management Science*, 27:219–239.

Banzhaf, J. F. (1965). Weighted voting does not work: A mathematical analysis. *Rutgers Law Review*, 19:317–343.

Baron, D. P. and Ferejohn, J. A. (1989). Bargaining in legislatures. *American Political Science Review*, 83:1181–1206.

Bergantiños, G. and Vidal-Puga, J. J. (2005a). Additivity in cost spanning tree problems. RGEA WP 10-05, University of Vigo, Spain.

Bergantiños, G. and Vidal-Puga, J. J. (2005b). Several approaches to the same rule in cost spanning tree problems. Mimeo.

Bergantiños, G. and Vidal-Puga, J. J. (2007a). A fair rule in minimum cost spanning tree problems. *Journal of Economic Theory*, 137:326–352.

Bergantiños, G. and Vidal-Puga, J. J. (2007b). The optimistic TU game in minimum cost spanning tree problems. *International Journal of Game Theory*, 36:223–239.

Bird, C. G. (1976). On cost allocation for a spanning tree: a game theoretic approach. *Networks*, 6:335–350.

Bondereva, O. N. (1963). Some applications of linear programming methods to the theory of cooperative games. *Problemy Kybernetiki*, 10:119–139.

Borm, P., Fiestras-Janeiro, G., Hamers, H., Sánchez, E., and Voorneveld, M. (2002). On the convexity of games corresponding to sequencing situations with due dates. *European Journal of Operational Research*, 136:616–634.

Borm, P., Hamers, H., and Hendrickx, R. (2001). Operations research games: A survey. *TOP*, 9:139–199.

Branzei, R., Moretti, S., Norde, H., and Tijs, S. (2004). The P-value for cost sharing in minimum cost spanning tree situations. *Theory and Decision*, 56:47–61.

Çiftçi, B. B., Borm, P., and Hamers, H. (2007). Highway games on weakly cyclic graphs. CentER-DP 104, Tilburg University, Center for Economic Research, The Netherlands.

Çiftçi, B. B., Borm, P., and Hamers, H. (2009a). Family sequencing problems and cooperation. Preprint.

Çiftçi, B. B., Borm, P., and Hamers, H. (2009b). Population monotonic path schemes for simple games. *Theory and Decision*. (in press).

Çiftçi, B. B., Borm, P., Hamers, H., and Slikker, M. (2008). Batch sequencing and cooperation. CentER-DP 100, Tilburg University, Center for Economic Research, The Netherlands.

Çiftçi, B. B. and Tijs, S. (2009). A vertex oriented approach to the equal remaining obligations rule for minimum cost spanning tree situations. *TOP*. (in press).

Claus, A. and Kleitman, D. (1973). Cost allocation for a spanning tree. *Networks*, 3:289 − 304.

Cruijssen, F., Borm, P., Fleuren, H., and Hamers, H. (2005). Insinking : A methodology to exploit synergy in transportation. CentER-DP 121, Tilburg University, Center for Economic Research, The Netherlands.

Curiel, I., Pederzoli, G., and Tijs, S. (1989). Sequencing games. *European Journal of Operational Research*, 40:344–351.

Curiel, I., Potters, J., Prasad, R., Tijs, S., and Veltman, B. (1993). Cooperation in one machine scheduling. *Mathematical Methods of Operations Research*, 38:113–129.

Derks, J. J. M. and Haller, H. H. (1999). Null player out? linear values for games with variable supports. *International Game Theory Review*, 1:301–314.

Dobson, G., Karmarkar, S., and Rummel, J. L. (1987). Batching to minimize flow times on on one machine. *Management Science*, 33:784–799.

Dubey, P., Neyman, A., and Weber, R. J. (1981). Value theory without efficiency. *Mathematical Methods of Operations Research*, 6:122–128.

Dutta, B. and Kar, A. (2004). Cost monotonicity, consistency and minimum cost spanning tree games. *Games and Economic Behavior*, 48:223–248.

Estevez-Fernandez, A., Borm, P., and Hamers, H. (2007). Project games. *International Journal of Game Theory*, 36:149–176.

Feltkamp, V., Muto, S., and Tijs, S. (1994). On the irreducible core and the equal remaining obligations rule of minimum cost spanning extension problems. CentER-DP 106, Tilburg University, Center for Economic Research, The Netherlands.

Gerichhausen, M. and Hamers, H. (2008). Partitioning sequencing situations and games. *European Journal of Operational Research*, (in press).

Gillies, D. (1953). Some theorems on n-person games. Ph.D. thesis, Princeton University, University Press Princeton, New Jersey.

Graham, R. and Hell, P. (1985). On the history of minimum spanning tree problem. *Annals of the History of Computing*, 7:43–57.

Granot, D., Hamers, H., and Tijs, S. (1999). On some balanced, totally balanced and submodular delivery games. *Mathematical Programming*, 86:355–366.

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

Granot, D. and Huberman, G. (1982). The relationship between convex games and minimal cost spanning tree games: A case for permutationally convex games. *SIAM Journal of Algorithms and Discrete Methods*, 3:288–292.

Granot, D. and Huberman, G. (1984). On the core and nucleolus of the minimum cost spanning tree games. *Mathematical Programming*, 29:323–347.

Hamers, H., Borm, P., and Tijs, S. (1995). On games corresponding to sequencing situations with ready times. *Mathematical Programming*, 70:1–13.

Hamers, H., Klijn, F., and Suijs, J. (1999). On the balancedness of multimachine sequencing games. *European Journal of Operational Research*, 119:678–691.

Hamers, H., Klijn, F., and van Velzen, B. (2005). On the convexity of precedence sequencing games. *Annals of Operations Research*, 137:161–175.

Hamers, H., Suijs, J., Tijs, S., and Borm, P. (1996). The split core for sequencing games. *Games and Economic Behavior*, 15:165–176.

Herer, Y. and Penn, M. (1995). Characterization of naturally submodular graphs: A polynomial solvable class of the tsp. *Proceedings of the American Mathematical Society*, 123:613–619.

Kalai, E. and Zemel, E. (1982). Totally balanced games and games of flow. *Mathematics of Operations Research*, 7:476–478.

Kar, A. (2002). Axiomatization of the shapley value on minimum cost spanning tree games. *Games and Economic Behavior*, 38:265–277.

Kruskal, J. B. J. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7:48–50.

Kuipers, J. (1997). Minimum cost forest games. *International Journal of Game Theory*, 26:367 – 377.

Laver, K. and Schofield, N. J. (1990). *Multiparty governments: the politics of coalitions in Europe*. Oxford University Press, Oxford.

Le Breton, M., Owen, G., and Weber, S. (1992). Strongly balanced cooperative games. *International Journal of Game Theory*, 20:419–427.

Lee, C. Y. and Chen, Z. L. (2001). Machine scheduling with transportation considerations. *Journal of Scheduling*, 4:3–24.

Lee, C. Y., Uzsoy, R., and Martin-Vega, L. A. (1992). Efficient algorithms for scheduling semiconductor burn-in operations. *Operations Research*, 40:764–775.

Liaee, M. M. and Emmons, H. (1997). Scheduling families of jobs with setup times. *International Journal of Production Economics*, 51:165–176.

Megiddo, N. (1978). Cost allocation for steiner trees. *Networks*, 8:1–6.

Monderer, D. and Samet, D. (2002). Variations of the shapley value. In Aumann, R. and Hart, S., editors, *Handbook of Game Theory with Economic Applications*, pages 2055–2076. Elsevier.

Moretti, S., Branzei, R., Norde, H., and Tijs, S. (2005). Cost monotonic construct and charge rules for connection situations. CentER-DP 104, Tilburg University, Center for Economic Research, The Netherlands.

Mosquera, M. A. and Zarzuelo, J. M. (2006). Sharing costs in highways: a game theoretical approach. Preprint.

Norde, H., Moretti, S., and Tijs, S. (2004). Minimum cost spanning tree games and population monotonic allocation schemes. *European Journal of Operational Research*, 154:84–97.

Norde, H. and Reijnierse, H. (2002). A dual description of the class of games with a population monotonic allocation scheme. *Games and Economic Behavior*, 41:322–343.

Owen, G. (1975). On the core of linear production games. *Mathematical Programming*, 9:358–370.

Pinedo, M. (2005). *Planning and Scheduling in Manufacturing and Services*. Springer, New York.

Potters, J., Curiel, I., and Tijs, S. (1992). Traveling salesman games. *Mathematical Programming*, 53:199–211.

Potts, N. C. and Kovalyov, Y. M. (2000). Scheduling with batching: A review. *European Journal of Operational Research*, 120:228–249.

Potts, N. C. and Van Wassenhove, L. N. (1992). Integrating scheduling with batching and lot-sizing: a review of algorithms and complexity. *Journal of the Operational Research Society*, 43:395–406.

Prim, R. C. (1957). Shortest connection networks and some generalizations. *Bell Systems Technical Journal*, 36:1389–1401.

Riker, W. (1962). *The theory of political coalitions*. Yale University Press, New Haven.

Rosenthal, E. C. (1987). The minimum cost spanning forest game. *Economics Letters*, 23:355–357.

Santos, C. and Magazine, M. (1985). Batching in single operation manufacturing systems. *Operations Research Letters*, 4:99–103.

Schmeidler, D. (1969). The nucleolus of a characteristic function game. *SIAM Journal on Applied Mathematics*, 17:1163–1170.

Shapley, L. S. (1953). A value for n-person games. *Annals of Mathematics Studies*, 28:307–317.

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

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

Shapley, L. S. and Shubik, M. (1972). The assignment game: The core. *International Journal of Game Theory*, 1:111–130.

Shenoy, P. P. (1979). On coalition formation: a game-theoretical approach. *International Journal of Game Theory*, 8:133–164.

Slikker, M. (2005). Balancedness of sequencing games with multiple parallel machines. *Annals of Operations Research*, 137:177–189.

Slikker, M. (2006). Balancedness of multiple machine sequencing games revisited. *European Journal of Operational Research*, 174:1944–1949.

Slikker, M., Norde, H., and Tijs, S. (2003). Information sharing games. *International Game Theory Review*, 5:1–12.

Sprumont, Y. (1990). Population monotonic allocation schemes for cooperative games with transferable utility. *Games and Economic Behavior*, 2:378–394.

Straffin, P. D. (1994). Power and stability in politics. In Aumann, R. and Hart, S., editors, *Handbook of Game Theory with Economic Applications*, pages 1127–1151. Elsevier.

Suijs, J., Hamers, H., and Tijs, S. (1997). On consistency of reward allocation rules in sequencing situations. In Haneveld, K., Vrieze, O., and Kallenberg, L., editors, *Ten Years LNMB*, pages 223–232. CWI Tract.

Tijs, S. (1981). Bounds for the core and the τ-value. In Moeschlin, O. and Pallaschke, D., editors, *Game Theory and Mathematical Economics*, pages 123–132. North Holland.

Tijs, S., Branzei, R., Moretti, S., and Norde, H. (2006). Obligation rules for minimum cost spanning tree situations and their monotonicity properties. *European Journal of Operational Research*, 175:121–134.

van den Nouweland, A., Krabbenborg, M., and Potters, J. (1992). Flow-shops with a dominant machine. *European Journal of Operational Research*, 62:38–46.

van Velzen, B. (2006). Sequencing games with controllable processing times. *European Journal of Operational Research*, 172:64–85.

van Velzen, B. and Hamers, H. (2003). On the balancedness of relaxed sequencing games. *Mathematical Methods of Operations Research*, 57:287–297.

von Neumann, J. and Morgenstern, O. (1944). *Theory of Games and Economic Behavior*. Princeton University Press, Princeton.

Weber, R. (1988). Probabilistic values for games. In Roth, A., editor, *The Shapley value: Essays in honor of Lloyd S. Shapley*, pages 101–120. Cambridge University Press.

Webster, S. and Baker, R. K. (1995). Scheduling groups of jobs on a single machine. *Operations Research*, 43:692–703.

Winter, E. (2002). The Shapley value. In Aumann, R. and Hart, S., editors, *Handbook of Game Theory with Economic Applications*, pages 2025–2054. Elsevier.

Samenvatting

Hoofdstuk 2 bestudeert allocatieproblemen die samenhangen met een special type machinevolgorde problemen. In machinevolgorde problemen is er een groep klanten, die elk precies een taak willen laten uitvoeren op een (of meerdere) machine(s). De kosten van elke klant hangen af van het tijdstip waarop hun taak door de machine is uitgevoerd. Als we aannemen dat er een beginvolgorde bestaat op de verschillende taken dan zijn er twee vragen die beantwoord dienen te worden. “Welke volgorde zorgt ervoor dat de totale, gezamenlijke kosten geminimaliseerd wordt” en “Hoe dienen de hiermee corresponderende maximale kostenbesparingen op een eerlijke manier onder de klanten verdeeld te worden?”. Curiel et al. (1989) introduceerden een speltheoretische aanpak voor het verdelingsvraagstuk binnen de context van specifieke 1-machinevolgorde problemen met lineaire individuele kostenfuncties door corresponderende machinevolgorde spelen te analyseren. Aangetoond werd dat dit soort spelen convex en derhalve ook gebalanceerd zijn: er bestaat een zogenaamde stabiele verdeling van de maximaal haalbare kostenbesparingen onder alle klanten zodanig dat geen enkele deelgroep, in totaal, hogere kostenbesparingen kan behalen door toegelaten onderlinge volgordewisselingen dan wat deze coalitie in totaal volgens het verdelingsvoorstel krijgt toegewezen. Een speciale regel die voor elk 1-machinevolgorde probleem binnen de bestudeerde klasse een dergelijke stabiele uitkomst voorschrijft blijkt de zogenaamde EGS-regel te zijn, die bovendien axiomatisch gekarakteriseerd is. In de literatuur is het elementaire model van Curiel et al (1989) verfijnd door extra aannames op de taken te beschouwen zoals tijdstippen van beschikbaarheid, uiterlijke uitvoeringstijdstippen, andersoortige kostenfuncties etc. In alle studies tot nu toe wordt aangenomen dat een machine slechts een taak tegelijkertijd kan uitvoeren. In de praktijk echter bestaan er ook batch machines die in staat zijn tegelijkertijd meerdere taken uit te voeren.

In paragraaf 2.1 breiden we de speltheoretische aanpak voor machinevolgorde problemen uit naar batch machines. Eerst beschouwen we 1-batchmachinevolgorde problemen waarbij de capaciteit van de batchmachine vastligt en het maximaal aantal taken weergeeft dat deze machine tegelijkertijd kan uitvoeren. Aangetoond wordt dat de corresponderende 1-spelen convex zijn. Bovendien wordt een expliciete uitdrukking voor de Shapley waarde voor dit soort spelen afgeleid. Ook wordt de EGS-regel doorvertaald naar dit soort situaties en axiomatisch gekarakteriseerd. Vervolgens beschouwen we zogenaamde relaxaties waarbij de aannames over toelaatbare volgordewisselingen voor deelgroepen wordt gevarieerd. Voor deze relaxaties wordt aangetoond dat stabiele verdelingsvoorstellen bestaan. Tot slot kijken we naar m-batchmachinevolgorde problemen waarin elke taak dezelfde volgorde door de m batchmachines dient te doorlopen. In het bijzonder identificeren we hier twee gevallen zodanig dat het corresponderende m-spel een bijzonder type 1-spel is.

Een andere gemeenschappelijke aanname in de analyse van machinevolgorde spelen is dat geen omsteltijden vereist zijn. Deze aanname vereist dat de omsteltijden verwaarloosbaar zijn of in de verwerkingstijden kunnen worden omvat. Nochtans, in de praktijk, als een machine verschillende types taken moet uitvoeren, dan is een significante omsteltijd bijna altijd vereist (zie bijvoorbeeld Pinedo, 2005).

In paragraaf 2.2, analyseren we allocatieproblemen die gerelateerd zijn aan zogenaamde familie machinevolgorde problemen. Bij familie machinevolgorde problemen, zijn de taken verdeeld in families. Er is geen omsteltijd vereist als een andere taak van dezelfde familie er aan vooraf gaat. Een omsteltijd is echter wel vereist als een taak een lid van een andere familie volgt. Wij associëren cooperatieve spelen met familie machinevolgorde problemen door de waarde van een coalitie te definiëren als de maximumbesparing door middel van een toelaatbare volgordewisselingen. We bewijzen dat elk familie machinevolgorde spel een niet-lege core heeft. Dit resultaat wordt verkregen door aan te tonen dat een specifieke marginale vector van het spel een *core-element* is.

In hoofdstuk 3 bestuderen we allocatieproblemen die voortkomen uit verbindingsproblemen zoals in een economische omgeving waar agenten willen samenwerken en gezamenlijk willen investeren in de aanleg/onderhoud van een gemeenschappelijk netwerk. We bekijken twee speciale klassen van deze verbindingsproblemen: minimum opspannende boom problemen, waarin agenten verbinding willen maken met een bron, en *highway* problemen waarin agenten een verbinding leggen tussen een vertrek- en aankomstpunt.

Neem een aantal inwoners van een dorp dat een netwerk van leidingen wil aanleggen van hun huis naar een watervoorziening. Elke inwoner kan een directe verbinding maken met de watervoorziening, maar een dergelijke beslissing zal waarschijnlijk inefficiënt zijn. In de meeste gevallen zal het goedkoper zijn als enkele inwoners rechtstreeks verbonden zijn met de watervoorziening, terwijl anderen zich aansluiten op de watervoorziening via andere inwoners. Inderdaad, in deze situatie zal een netwerk ontstaan dat de totale verbindingskosten minimaliseert. Een dergelijk netwerk wordt gevonden door een minimum opspannende boom (mob). Deze situaties worden dan ook mob situaties genoemd. Indien de agenten het eens zijn over de mob die aangelegd moet worden, ontstaat het tweede probleem waarbij de gezamenlijke minimale kosten van de mob op een eerlijke manier verdeeld moeten worden onder de agenten. Deze allocatieproblemen zijn voor het eerst geintroduceerd in de economische literatuur door Claus en Kleitman (1973). Bird (1976) introduceerden een speltheoretische aanpak voor deze problemen door een cooperatief TU-spel te koppelen aan mob problemen. Vervolgens zijn voor mob problemen een groot aantal verdeelregels geintroduceerd in de literatuur die geschikt zijn om toe te passen op mob situaties. Bijvoorbeeld, de *equal remaining obligations* regel (Feltkamp et al., 1994) voor mob situaties voldoet aan een groot aantal aantrekkelijke eigenschappen (bijvoorbeeld monotonie in kosten, populatie monotonie, gelijke behandeling) en kan op verschillende manieren worden verkregen. Bovendien tonen Bergantiños en Vidal-Puga (2007a) aan dat andere regels uit de literatuur sommige eigenschappen niet bezitten die de *equal remaining obligations* regel wel heeft.

De oorspronkelijke definitie van de *equal remaining obligations* regel bestaat uit een sequentiële procedure: Kruskal's algoritme (Kruskal, 1956) wordt gebruikt om een mob te construeren en bij elke stap van het algoritme worden de kosten van de geconstrueerde verbinding verdeeld tussen de agenten die deze verbinding gebruiken. In paragraaf 3.2 introduceren we een andere benadering en onderbouwing voor de *equal remaining obligations* regel. Om dit te bereiken definiëren we eerst de knoop georiënteerde aanleg- en betaalprocedure die zowel tot een mob leidt als een verdeelregel van de kosten leidt waarin elke agent de verbinding betaalt die hij kiest om aan te leggen. Vervolgens tonen we aan dat de *equal remaining obligations* regel het gemiddelde is van de verdeelregels die verkregen zijn uit de knoop georiënteerde aanleg- en betaalprocedure voor elke volgorde van agenten. In paragraaf 3.3 en 3.4 tonen we aan dat de resultaten uit paragraaf 3.2 uit te breiden zijn naar de minimum opspannende bos situaties (cf. Rosenthal, 1987) en mob situaties met twee bronnen.

Het grootste deel van de huidige literatuur op het gebied van de allocatie van de kosten in verbindingsproblemen richt zich op de mob problemen of varianten daarvan. Een gemeenschappelijk kenmerk van deze problemen is dat elke persoon in het probleem een verbinding met een niet-lege deelverzameling van de beschikbare bronnen in het netwerk moet maken. Nochtans, in sommige verbindingssituaties is er geen bepaald punt waarmee elke persoon in het probleem moet worden verbonden. Bijvoorbeeld, de gebruikers van een *highway* netwerk hebben slechts een verbinding nodig tussen hun vertrek- en aankomstpunt in het netwerk.

Mosquera en Zarzuelo (2006) bestuderen het allocatieprobleem dat samenhangt met de bouwkosten van een *highway* netwerk. Voor dit doel, definëren zij formeel *highway* problemen en analyseren de bijbehorende cooperatieve kostenspelen, *highway* spelen genoemd. In een *highway* probleem, worden de mogelijkheden betreffende de bouw van het *highway* netwerk bepaald door een verbonden graaf. De verzameling knopen van de graaf vertegenwoordigt de potentiele vertrek- en aankomstpunten en de zijden in de graaf vertegenwoordigen de mogelijke wegverbindingen die kunnen worden geconstrueerd. Gegeven een *highway* probleem wordt een corresponderend *highway* spel gedefiniërd als een cooperatief kostenspel dat aan elke coalitie van spelers de totale kosten van de goedkoopste keuze van zijden in de graaf associërt die het vertrek- en aankomstpunt van elk lid van de coalitie met elkaar verbindt. Mosquera en Zarzuelo (2006) beperken zich tot *highway* problemen waarin de onderliggende graaf een *boom* is. In deze context is er slechts *één* pad tussen een vertrek- en een aankomstpunt.

In paragraaf 3.5 bestuderen we *highway* problemen waarin de onderliggende graven zijn *weakly cyclic*, d.w.z., verbonden graven waarvoor elke zijde in de graaf in hoogstens *één* cykel bevat is. We bestuderen eerst de klasse van graven waarvoor de corresponderende spelen altijd concaaf zijn. Voor dit doel, wordt een graaf G als *highway-spel-concaaf* gedefinëerd als voor elk *highway* probleem waarin G de onderliggende graaf is, het corresponderende *highway* spel concaaf is. Wij bewijzen dat een graaf *highway-spel-concaaf* is dan en slechts dan als deze *weakly triangular* is. Dan richten we ons op de *core* van *highway* spelen die door weakly cyclic graven worden geïnduceerd. Kuipers (1997) laat zien dat de *highway* spelen die door cyclische graven worden geïnduceerd hoeven niet gebalanceerd te zijn in het algemeen. Nochtans bewijzen we dat de *highway* spelen op weakly cyclic graven gebalanceerd zijn.

Wij beschouwen in Hoofdstuk 4 de vorming van coalities door sequentiële bilaterale onderhandelingen in stemmensituaties. Wij modelleren deze situaties door *simpele spelen* en nemen aan dat de macht van elke kiezer wordt beschrijven via een *machtsindex*. Een *pad schema* voor een simpele spel bestaat uit een *pad*, d.w.z., een volgorde van coalities die wordt gevormd tijdens het onderhandelingsproces en een *schema*, d.w.z., een allocatievector voor elke coalitie in het pad. Een pad schema wordt genoemd *populatie monotoon* als de allocatie van een speler niet afneemt als de pad-coalitie groeit. In de eerste plaats richten we ons op *Shapley pad schemas* van simpele spelen waarin voor elke pad-coalitie de Shapley waarde van het geassociërde *deelspel* de allocatie bepaalt. We bewijzen dat het bestaan van *veto spelers* nodige en voldoende voorwaarde is voor het bestaan van populatie monotoon Shapley pad schemas. Voorts is een Shapley pad schema populatie monotoon is dan en slechts dan als de eerste winnende coalitie die via het pad wordt gevormd, elke minimale winnende coalitie van het spel bevat. Wij bewijzen ook dat elke Shapley pad schema populatie monotoon is dan en slechts dan als de coalitie van veto spelers een winnende coalitie is. Wij bestuderen verder hoe deze resultaten uit te breiden naar *probabilistische waarden*, die de Shapley waarde generaliseren.

Author index

Subject index