Cost Allocation Problems in Network and Production Settings

by Endre Bjørndal

A dissertation submitted for the degree of dr. oecon.
April 2002

02h002912

ehs. 2

Acknowledgements

First and foremost I would like to thank my advisor, Kurt Jörnsten. He has contributed directly through his many creative ideas and comments, and indirectly through his enthusiasm and moral support.

I am also grateful to my coauthors Maurice Koster, Stef Tijs, and Herbert Hamers at Tilburg University, who contributed to chapters 2 and 4. Mette Bjørndal has introduced me to the functioning of electricity markets and networks, the subject of Chapter 7, and has also read and given useful comments on an unfinished version of this chapter.

This research is sponsored by the Foundation for Research in Economics and Business Administration and Telenor, and I very much appreciate this support. I am also thankful for the hospitality and friendly atmosphere provided by the Department of Finance and Management Science at NHH. The support of friends and colleagues there, as well as at Bergen University College, has been important to me. I would also like to thank CentER, Tilburg University, for accepting me as a visitor for three months in 1998.

Finally, I would like to thank my family and friends, who have contributed indirectly with their support and patience.

NHH, Bergen, April 2002
Endre Bjørndal

Contents

Chapter 1

Introduction

1.1 Motivation

This thesis deals with situations in which a set of individuals consider forming a joint venture, such as the construction of a communication network or the operation of a production facility. By cooperating, they may realize cost savings (or increased revenues), hence cooperation is beneficial for the *collective* of agents. Whether cooperation will take place or not depends on whether the participants manage to share the cost in a manner such that every *individual* finds it worthwhile to take part in the joint venture. It is the problem of allocating the joint cost among the agents that is the subject of the thesis.

A cost allocation *method* is a mechanism that solves the cost allocation problem, and could be anything from a mathematical formula to a procedure such as a competitive market or a court trial. In all the problems studied in this thesis we implicitly assume that the information needed to compute allocations is known by all participants, and that binding agreements can be enforced. These assumptions leads us to model the situations as *cooperative games*, and the focus is on finding an appropriate formula for making the allocation. If these assumptions were not valid, the focus should be on designing a procedure, and *non-cooperative game theory* has a role to play.

In spite of the relatively primitive assumptions used, the problems that we study are complex. Sometimes, as in Chapter 7, complexity stems from a relatively rich description of the technology embedded in the joint venture. Also, the ability of dissatisfied individuals to break out of the agreement and form *coalitions*, i.e., subgroups establishing their own joint venture, adds further complexity to the models, since the number of coalitions grows exponentially with the number of players. Complexity is clearly a constraint when suggesting solutions to allocation problems, since excessive computational requirements or non-transparency of cost allocation methods may prohibit their use in practice.

There exist an abundance of solution concepts for cooperative games. Although words such as *fair* or *just* are used to describe all of them, it is clear that the theory does not offer an unambiguous answer to the question of how costs should be allocated. The most widely used solution concept is the *core*. A solution belongs to the core if no individual, or group of individuals, can lower their cost by breaking out of the agreement and forming their own (reduced) joint venture. The core will be used extensively throughout this thesis. Since the core is a convincing solution concept in most applications, other solution concepts, such as the *Shapley value* and the *nucleolus*, are often evaluated in relation to it. Much of the literature[1] on cooperative games focuses on describing solution methods in terms of certain *axioms* that may be perceived as fair in certain settings. An example is *population-monotonicity*, which is used in Chapter 4, and in Chapter 3 we make use of the *consistency* axiom.

1.2 Outline of the thesis

All the chapters of the thesis, with the exception of Chapter 6, deal with *applications* of cooperative game theory. The games studied in chapters 2-4 have relatively simple structures, and we can therefore focus on providing nice descriptions of core allocations. In the remaining chapters, the games studied are more complex, and we therefore need to focus more on computational issues.

In Chapter 2 we study the *standard fixed tree game*, for which we generalize

[1]See Moulin (1988).

the intuitive *painting stories* of Maschler et al. (1995). In Section 2.4 we study the weighted *down-home* allocation rule, and we show that any core point can be obtained as a weighted down-home allocation by appropriately choosing the weights. Moreover, we give expressions that can be used to calculate these weights. This result is used to show that the core equals the set of weighted down-home allocations. By relating the weighted down-home allocation to the weighted Shapley value, we give a constructive proof of the result in Monderer (1992), where it was shown that the core equals the set of weighted Shapley values for the class of convex games. The weighted *neighbour-home* allocation, in Section 2.5, differs from the down-home allocation in the type of social obligations that it imposes on the players, and has the nucleolus as a special case. The results of Section 2.5 are analogous to those of Section 2.4, i.e., we show how any core point can be obtained as a weighted neighbour-home allocation by appropriately choosing the weights, and we give explicit formulas for the weights.

In Chapter 3 we relate two well-known problems to each other. The *flow sharing problem* allocates the maximal flow of a capacitated network among its terminal nodes, and the *bankruptcy problem* allocates the worth of an estate among a set of claimants. The literature on the former problem has focused on egalitarian solutions, e.g., the lexicographically optimal solution, while many types of solutions have been proposed for the latter problem. Section 3.2 introduces the bankruptcy problem, and Section 3.3 shows how the flow sharing problem may be seen as a generalization of the single-estate bankruptcy problem. In Section 3.4 we form reduced two-person flow sharing problems, which can be interpreted as single-estate bankruptcy problems. By requiring that the solution to the n-person problem should be consistent with the solutions to every two-person problem, we get unique solutions to the former problem. Two such solutions are studied, the *contested garment* (CG) consistent solution and the *constrained equal award* (CEA) consistent solution. The latter is similar to many of the solutions of the flow sharing problem that are known from the operations research literature. In Section 3.5 we formulate the flow sharing game, which is a generalization of the bankruptcy game of Aumann and Maschler (1985). The set of maximal flows of the flow sharing problem corresponds to the core of this game, and the CG-and CEA-consistent solutions of Section 3.4 coincide with the nucleolus and the constrained egalitarian solution, respectively, of the game. Section 3.6 describes two transfer schemes that converge to the two solutions of Section 3.4. Section 3.7 generalizes the CEA-consistent solution by assigning weights to the players, and we also relate this solution to the lexicograhically optimal solution of the flow sharing problem.

In Chapter 4 we study a cost allocation problem arising in connection with a bank ATM network, where the member banks seek to allocate the transactions costs. The cost allocation problem is equivalent to a *cost savings* allocation problem, where the banks owning ATMs provide cost savings to the (customers of) other banks. The problem is thus how to divide these cost savings between the bank(s) owning the ATMs, and the "owners" of the transactions that are processed by the ATMs. The game is defined by aggregating over the locations (of the ATMs), and Section 4.3 investigates the properties of single-location games, which are shown to be a special case of the information market games of Muto et al. (1989) and Potters and Tijs (1989). If only one bank has ATMs in a location, the core is relatively large, and several well-known solution concepts coincides with a central point in the core. According to this allocation, the cost savings are split equally between the bank owning the ATMs, and the banks whose transactions need to be processed by the ATMs. If several banks have ATMs in a location, the core consists of a single point. In Section 4.4, we combine allocations for the single-location games, and two allocation rules are studied. The *equal-split* rule splits cost savings equally whenever possible, and corresponds to the τ-value of the cost savings game. It yields core points, as does the *transaction-based* rule, where no splitting of cost savings is performed, but the latter rule has the additional advantage of being population-monotonic, i.e., no member bank looses as a result of the inclusion of new member banks in the network.

Chapter 5 is about *linear production games*, in which a set of agents own the input resources to a production process that can be described by a linear programming problem. It is known from Owen (1975) that a subset of the core can be obtained from the solution of the dual corresponding to the grand coalition, but there is no readily available description of the rest of the core. In Section 5.3 we provide lower (upper) bounds on the values of the game by aggregating over columns (rows) of the LP-problem. By choosing aggregation weights corresponding to optimal solutions of the primal (dual) LP-problem, we can create new games whose core form a superset (subset) of the original core. In Section 5.4 we show how one can obtain an estimate of the resulting error, in terms of and ϵ-core, by solving a mixed-integer programming problem, and suggest an iterative procedure for improving the bounds. In Section 5.5 we investigate, using a set of numerical examples, how the performance of the aggregation approach depends on the structure of the problem data.

Chapter 6 is a note on the computation of the pre-nucleolus. The LP-based procedure of Kopelowitz (1967) requires, in every iteration, the identification of active constraints, and we show, via an example, that using an interior-point method for solving the LP-problem reduces the number of iterations needed, compared to using an extreme-point method.

Chapter 7 deals with cost allocation in an electricity network. Producers and consumers inhabit the nodes of the network, and have to cover the fixed cost of the transmission facilities. In Section 7.2 we introduce a model of a network connecting a set of regional electricity markets. The network is considered fixed, i.e., no lines are added/removed, and the inhabitants of each node can choose whether to make use of the network or not. The consumers and producers inhabiting the nodes are described by demand and supply functions, and the network is described by the reactances and thermal capacities of the existing lines. Given that a certain subset of nodes choose to connect themselves through the network, equilibrium quantities and prices will be determined as a result of these parameters. In Section 7.3 we formulate the cost allocation problem and the corresponding cooperative game. The value of a coalition, forming an upper bound on the cost that can be assigned to its members, is defined as the increase in total surplus (excluding the surplus of the network owner) that occurs when the coalition starts using the network. The main result of Section 7.3 is that the core will be nonempty if the grand coalition is not constrained by the line capacities. In Section 7.4 we discuss briefly how the network owner may handle the problem of an empty core. Section 7.5 illustrates, using examples, how cost allocation methods based on observed quantities may fail to yield core allocations. However, by combining several methods, as shown in Section 7.6, core points may in many cases be obtained. An LP-based procedure for finding such combinations is discussed.

1.3 Cooperative games with transferable utility

We will denote the set of *players* by $N := \{1, \ldots, n\}$. A subset $S \subseteq N$ will be called a *coalition*, and the set of coalitions will be denoted by 2^N. The set N will sometimes be referred to as the *grand coalition*. To every coalition we assign a value, given by the *characteristic function* $c : 2^N \to \mathbf{R}^1$, and in most cases it will be convenient to refer to games by their characteristic function.[2] A game c is said to be *sub-additive* if

$$c(S \cup T) \leq c(S) + c(T) \qquad \text{for all } S, T \subseteq N \text{ such that } S \cap T = \emptyset. \tag{1.1}$$

For an additive game, the inequality in (1.1) holds as an equality. The game c is *concave* if

$$c(S \cup T) + c(S \cap T) \leq c(S) + c(T) \qquad \text{for all } S, T \subseteq N. \tag{1.2}$$

It is easily seen that concavity implies sub-additivity.[3]

A solution of a game is given by an allocation vector $x \in \mathbf{R}^n$, where x_i is the amount allocated to player $i \in N$. For a coalition $S \subseteq N$, we define the total amount allocated to the members of S as $x(S) := \sum_{i \in S} x_i$. We shall let $x_S := \{x_i\}_{i \in S}$ refer to the restriction of x to the members of S. In this chapter we define three solution concepts that are used in several of the chapters. These are the core, the Shapley value, and the nucleolus. Other solution concepts, such as the τ-value (Chapter 4) and the constrained egalitarian solution (Chapter 3) will be defined in the respective chapters where they are used.

[2] Problems where a positive value (revenue) is to be allocated, such as in chapters 3 and 5, could be fitted into this framework by letting c take negative values. Instead we shall follow the conventional approach and let the coalition values be given by the (non-negative) characteristic function $v : 2^N \to \mathbf{R}^1$. This means that the definitions of sub-additivity and concavity, as well as the definition of the core and the nucleolus, must be changed accordingly.

[3] The corresponding properties of a revenue game v, *super-additivity* and *convexity*, are obtained by reversing the inequalities of (1.1) and (1.2).

The set of *pre-imputations* is defined as

$$I^*(c) := \{x \in \mathbf{R}^n : x(N) = c(N)\},$$

i.e., the set of solution vectors such that the entire cost corresponding to the grand coalition is allocated. By adding the requirement that the solutions should be acceptable to individuals (*individual rationality*), we get the imputation set, i.e.,

$$I(c) := \{x \in \mathbf{R}^n : x(N) = c(N),\ x_i \leq c(i)\ \forall i \in N\}.$$

The *core* is obtained by adding a stronger requirement, that of *group rationality*, i.e.,

$$C(c) := \{x \in I^*(c) : x(S) \leq c(S)\ \forall S \subset N\}. \tag{1.3}$$

Bondareva (1961) and Shapley (1967) show that the core of a game is nonempty if and only if the game is *balanced*, i.e., iff

$$\sum_{S \subset N} \lambda_S c(S) \geq c(N) \tag{1.4}$$

for any set of non-negative weights $\{\lambda_S\}_{S \subset N}$ such that $\sum_{S \ni i} \lambda_S = 1$ for all $i \in N$. The core is used in all the chapters of this thesis.[4] In Chapter 5 we also use the concept of *strong* ϵ-*core*.[5] With respect to the game c, this is the set

$$C_\epsilon(c) := \{x \in I^*(c) : x(S) \leq c(S) - \epsilon\ \forall S \in 2^N \setminus \{N, \emptyset\}\}. \tag{1.5}$$

If $\epsilon \leq 0$, we have $C(c) \subseteq C_\epsilon(c)$.

In order to define the *nucleolus* (Schmeidler (1969)) of c, let $e(c, S, x) := c(S) - x(S)$ denote the *excess* of coalition S, given the allocation $x \in \mathbf{R}^n$. Then, let

$$\theta(x) := \{e(c, S_1, x), e(c, S_2, x), \ldots, e(c, S_{2^n-2}, x)\}$$

denote the vector obtained by sorting the excess values for all the $2^n - 2$ nonempty coalitions, except N, in a non-decreasing order, i.e., such that

[4]For a revenue game v, the equivalents to (1.3), (1.4), and (1.5) are obtained by reversing inequalities.

[5]See Maschler et al. (1979).

$i < j \Rightarrow e(c, S_i, x) \leq e(c, S_j, x)$. The nucleolus is the unique element in $I(c)$ for which the vector of excesses is lexicographically maximal[6], i.e.,

$$NU(c) := \{x \in I(c) : \theta(x) \geq_{LEX} \theta(y) \ \forall y \in I(c)\}. \qquad (1.6)$$

If, in the above definition, we replace $I(c)$ by $I^*(c)$, we get the pre-nucleolus $NU^*(c)$. The nucleolus, as opposed to the Shapley value, is always a member of the core[7], if the latter set is nonempty. Hence, if a game c has a nonempty core, then $NU^*(c) = NU(c)$. The nucleolus is studied in chapters 2, 3 and 4, as well as 6 and 7.

The *Shapley value* (Shapley (1953)) can be defined[8] as

$$\Phi_i(c) := \sum_{S \subset N} \frac{(|S|-1)!(|N|-|S|)!}{|N|!}[c(S) - c(S \setminus i)] \qquad i \in N. \qquad (1.7)$$

The Shapley value does not in general belong to the core. An alternative definition, used in Chapter 4, expresses $\Phi(c)$ as the average of the *marginal vectors*, and Shapley (1971) shows that for convex games, the core is the convex hull of the marginal vectors, implying that the Shapley value is in the core. In Chapter 2, a weighted generalization, due to Kalai and Samet (1988), is analyzed in connection with standard fixed tree games.

Sometimes we shall draw pictures illustrating[9] solutions, e.g., the core, for a specific game. For $n = 3$, the pre-imputation set $I^*(c)$ is a two-dimensional hyperplane, as is illustrated in Figure 1.1. For every coalition $S \subset N$, the set

$$H_S(c) := \{x \in I^*(c) : x(S) = c(S)\}$$

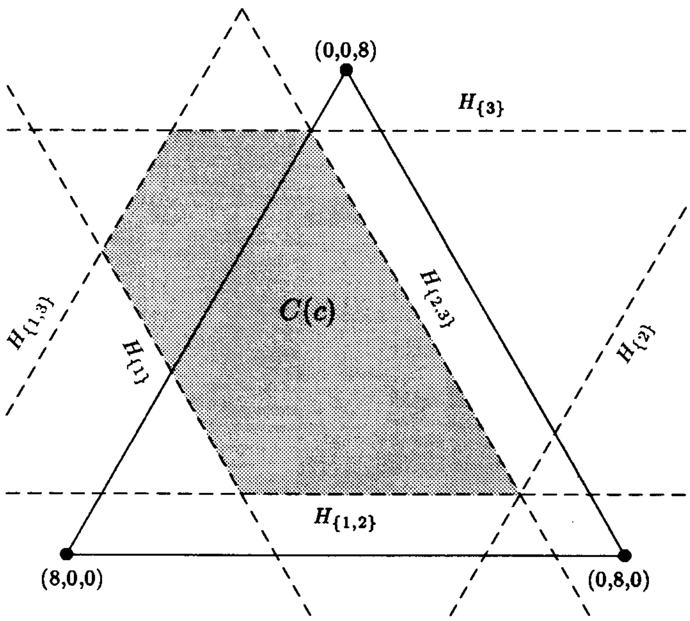
represents the set of points where coalition S pays exactly its stand-alone cost $c(S)$. In Figure 1.1, these hyperplanes are shown as dashed lines, and the core is the grey area enclosed by all the dashed lines. Note that the core contains points where some players are charged negative amounts, i.e., where they receive money for participating in the joint venture.

[6] The nucleolus of a revenue game v is obtained by *minimizing* the excess vector, where this vector is sorted in a non-*increasing* order.

[7] See e.g. Maschler et al. (1979).

[8] In contrast to the core and the nucleolus, the definition of the Shapley value does does not depend on whether we are dealing with a cost game or a revenue game.

[9] See Maschler et al. (1979) and Chardaire (2001) for geometric interpretations of some solution concepts.



S	$\{1\}$	$\{2\}$	$\{3\}$	$\{1,2\}$	$\{1,3\}$	$\{2,3\}$	N
$c(S)$	5	6	7	7	10	7	8

Figure 1.1: Graphical description of the core

Chapter 2

Weighted Allocation Rules for Standard Fixed Tree Games[†]

2.1 Introduction

We consider cost sharing problems arising from *standard fixed tree enterprises.* There is a fixed and finite set of agents who are connected through a fixed tree network to a special node called *root.* We seek to allocate the cost of this tree corresponding to the maintenance of the different connections. Many real-life situations can be modelled to fit in this general setting. For instance, consider the problem of allocating the maintenance cost of an irrigation network or a cablevision network, setting airport taxes for planes or setting dredging fees for ships. In a natural way each standard fixed tree problem gives rise to a *standard fixed tree game*, which relates each coalition of agents/players to the minimal expenses for maintaining the connections of all its members to the root. This makes it possible to investigate this type of problems with techniques from cooperative game theory. The same problem is studied in Megiddo (1978) , Koster et al. (1998) , whereas Granot et al. (1996), Maschler and Granot (1998) and Maschler et al. (1995) study a generalization, where more than one player is allowed to occupy each node. A special case, where the underlying structure of the game is a chain, is

[†]This chapter is based on Bjørndal, Koster and Tijs (1999).

also known as the airport problem and considered by Littlechild (1974), Littlechild and Owen (1977), Littlechild and Thompson (1977), Dubey (1982), Potters and Sudhölter (1999) , and Aadland and Kolpin (1998).

In Section 2.3, we are concerned with the core of the standard fixed tree game; essentially, this section is based on Koster et al. (1998) . Inspired by the *painting story* presented by Maschler et al. (1995) we introduce, in Section 2.4, the *weighted down-home allocation* in which each player is allocated a share, corresponding to his relative weight, of the cost of each edge along the path from the (local) root to his home. We show, by explicitly characterizing the corresponding weight system, that each core element can be obtained as a weighted down-home allocation. Especially, the core element as determined by the Shapley value corresponds to the weighted down-home allocation with equal weights to all players. Moreover, each weighted down-home allocation is equivalent to a weighted Shapley value, and therefore our results provide an alternative proof of the result in Monderer et al. (1992), where it is shown that the core of a concave game (it is well known that fixed tree games are concave) equals the set of weighted Shapley values. In Section 2.5 we introduce the *weighted neighbour-home allocation*, a generalization of the scheme in Maschler et al. (1995) for computing the nucleolus, and show that the set of weighted neighbour-home allocations equals the core. The weighted neighbour-home allocation is equal to the nucleolus in the special case where all players are given equal weight.

2.2 The fixed tree maintenance problem: the model and its game

In this chapter we consider a *fixed tree maintenance problem* $\mathcal{G} := (G, c, N)$. Here $G = (V, E)$ is a tree, i.e. a directed connected graph without cycles, with node set V and edge set E. The set V contains a node which has a special meaning. We denote this node by r and refer to it as the *root*. The function $c : E \to \mathbb{R}_+$, called *cost function*, associates with each edge e a cost $c(e)$. It can be interpreted as the cost to maintain e. At each node there is exactly one player, the finite set of all players is denoted by $N = \{1, \dots, n\}$ for some natural number n. The objective of the players is to maintain sufficiently many edges such that by the corresponding network each finds himself connected to the root. We address the problem of allocating the total costs of the network among the players.

We assume, for simplicity, that the root is not occupied and that only one edge is incident with the root[1]. Then $\mathcal{G}$ is referred to as simply a *maintenance problem*. In the sequel we identify nodes with players ($V = N \cup \{r\}$). For any subgraph G' of G, we will let $E(G')$ and $V(G')$ denote the corresponding edge set and node set, respectively. Sometimes we will also denote the player set corresponding to G' by $N(G') \subseteq V(G')$. For each node $i \in N$ there is a unique path P_i from the root to node i. If $V(P_i)$ consist of the nodes $j_0 = r, j_1, \dots, j_q = i$, then j_{q-1} is called the *predecessor* $\pi(i)$ of node i. We put $N(P_i) := V(P_i) \setminus \{r\}$. We denote by e_i the edge $(\pi(i), i)$, and we will sometimes write $c_i := c(e_i)$. The *precedence relation* $(V, \preceq)$ on the set of nodes and/or players is defined by $i \preceq j$ if and only if $i \in V(P_j)$. Analogously we define the precedence relation $(E, \preceq)$ on the edges. In this way, the edges are considered to be directed away from the root. A *trunk* of $G = (V, E)$ is a set of nodes $T \subseteq N$, which is closed under the precedence relation defined above, i.e. if $i \in T$ and $j \preceq i$, then $j \in T$. Also, let the *followers* of a node i be denoted by $F(i) := \{j \in N : i \preceq j\}$. A node i is called a *leaf* if $F(i) = \{i\}$. If $e = (i, j)$, then by B_e we denote the *branch* at i in the direction of j, i.e. the subgraph defined by $V(B_e) := \{i\} \cup F(j)$ and $E(B_e) := \{(k, \ell) \in E | k, \ell \in V(B_e)\}$. The set $N(B_e) := F(j)$ consists of the *users* of the edge e.

In this chapter we study the maintenance problem in the setting of cooperative game theory, by associating each maintenance problem with an appropriate cooperative *cost game*. By a *cost game* we mean an ordered pair (N, g) consisting of a finite set N of *players* and the *characteristic function* $g : 2^N \to \mathbf{R}$ relating each coalition of players $S \subseteq N$ to a real number $g(S)$ that is interpreted as the total cost of serving the collective S. Moreover, it is assumed that $g(\emptyset) = 0$, i.e. serving nobody can be done at no cost. The

[1]If more than one edge is incident with the root, then the cost game can be decomposed by considering each subtree separately. As shown in Granot and Huberman (1981), the core and nucleolus for the original cost game can be obtained as the cartesian product of the cores and nucleoli of for the subgames. This property also holds for the (weighted) Shapley value, because of the additivity of this solution concept.

set of all cost games is denoted Γ, and the restriction to cost games with player set N is denoted Γ^N. Each maintenance problem $\mathcal{G} = (G, c, N)$ is naturally related to the cost game $(N, c_{\mathcal{G}}) \in \Gamma$, where the cost $c_{\mathcal{G}}(S)$ of each coalition S is defined as the minimal cost needed to maintain all connections of the members of S to the root via a connected subgraph of (V, E), i.e.

$$c_{\mathcal{G}}(S) = \sum_{i \in T_S} c(e) \text{ for all } \quad \emptyset \neq S \subseteq N, \tag{2.1}$$

where $T_S = \{i \in N : \exists\, j \in S \text{ with } i \preceq j\}$, and $c_{\mathcal{G}}(\emptyset) = 0$. T_S is the smallest trunk containing S. Since $i \in T_S$ is equivalent to $S \cap N(B_{e_i}) \neq \emptyset$, we can rewrite (2.1) as

$$c_{\mathcal{G}}(S) = \sum_{\substack{e \in E \\ S \cap N(B_e) \neq \emptyset}} c(e_i) \text{ for all } \quad \emptyset \neq S \subseteq N, \tag{2.2}$$

i.e., the cost of an edge e should count towards the stand-alone cost of the coalition S if any of the users of e, the set $N(B_e)$, are members of S. Let the ***unanimity game*** (N, u_S) be given by

$$u_S(T) := \begin{cases} 1 & \text{if } S \subseteq T \\ 0 & \text{otherwise,} \end{cases} \tag{2.3}$$

for all $T \subseteq N$. The ***dual*** unanimity game (N, u_S^*) is then given by

$$u_S^*(T) = u_S(N) - u_S(N \setminus T) = \begin{cases} 1 & \text{if } S \cap T \neq \emptyset, \\ 0 & \text{otherwise,} \end{cases} \tag{2.4}$$

for all $T \subseteq N$. From (2.2) we see that the game $(N, c_{\mathcal{G}})$ can be expressed using the basis $\{(N, u_S^*)\}_{S \subseteq N}$ of dual unanimity games, i.e.,

$$c_{\mathcal{G}} = \sum_{e \in E} c(e) u^*_{N(B_e)}. \tag{2.5}$$

2.3 The core of a maintenance game

In this secction we will present some alternative characterizations of the core of maintenance games. Given some cost allocation vector $x \in \mathbf{R}^N$, we define the ***overflow*** over the edge e as the amount that the users of e pay in excess of the cost of the edges belonging to the branch B_e, i.e.

$$O_e(x) := \sum_{i \in N(B_e)} x_i - \sum_{f \in E(B_e)} c(f) = \sum_{i \in N(B_e)} (x_i - c_i).$$

If $e = (i, j)$, we will sometimes write $O_j(x)$ instead of $O_e(x)$, and it is easily seen that

$$O_j(x) = \sum_{\ell \in F(j)} (x_\ell - c_\ell) = (x_j - c_j) + \sum_{\ell \in \pi^{-1}(j)} O_\ell(x). \tag{2.6}$$

Characterizations of the core of the game $(N, c_{\mathcal{G}})$ are found in Koster et al. (1998) and Granot and Maschler (1998). The following proposition summarizes these results and adds a characterization of the core in terms of the overflows.

Proposition 2.3.1 *Let $x \in \mathbb{R}^N$. Then the following statements are equivalent:*

(i) $x \in C(c_{\mathcal{G}})$.

(ii) $x(N) = c_{\mathcal{G}}(N)$, $x \geq 0$, *and* $x(T) \leq c_{\mathcal{G}}(T)$ *for every trunk* T.

(iii) $x(N) = c_{\mathcal{G}}(N)$, $x \geq 0$, *and* $O_e(x) \geq 0$ *for all* $e \in E$.

(iv) *There exist* $y^e \in \Delta(N(B_e))$ *for all* $e \in E$, *such that*

$$x_i = \sum_{e \in E(P_i)} y_i^e c(e) \qquad \textit{for all } i \in N.$$

Proof. These results essentially appear as Propositions 3.1 [$(i) \Leftrightarrow (ii)$], 3.2 [$(ii) \Leftrightarrow (iii)$], and 3.3 [$(i) \Leftrightarrow (iv)$] in Koster et al. (1998). □

Definition 2.3.2 A *pseudo subtree* of a tree $G = (V, E)$ is a connected subgraph $G' = (V', E')$ such that there exists an $r' \in V(G')$ such that

(*i*) r' is the minimal element in $V(G')$ with respect to $\preceq$,

(*ii*) there is exactly one node in $V(G')$ that has r' as predecessor.

A pseudo subtree $G' = (V', E')$ of G rooted at r' yields a *restricted maintenance problem* $\mathcal{G}' = (G', c', N')$ where c' is the restriction of c to E', and $N' = V' \setminus \{r'\}$. Let $\mathcal{T} = (G^1, \ldots, G^p)$ be an ordered collection of pseudo subtrees of G. Then $\mathcal{T}$ is said to be a *partition of G into pseudo subtrees* if and only if (*i*) for all $k = 1, \dots, p$, there exists $r_k \in V(G^k)$ such that G^k is the pseudo subtree of G rooted at r_k,

(*ii*) $(N(G^1), \dots, N(G^p))$ is a partition of N.

Given an allocation vector x, let $E(x) := \{e \in E : O_e(x) > 0\}$. The graph $(V, E(x))$ contains p connected subgraphs, where $1 \leq p \leq n$. For each of these subgraphs, $1 \leq k \leq p$, we construct a pseudo subtree G^k with player set $N(G^k)$. Let $r_k \in V \backslash N(G^k)$ be such that $r_k \in V(P_i)$ for every $i \in N(G^k)$, and $r_k = \pi(i)$ for exactly one $i \in N(G^k)$. Let $V(G^k) := N(G^k) \cup \{r_k\}$ and $E(G^k) := \{e = (i,j) | i,j \in V(G^k)\}$. Then $G^k := (V(G^k), E(G^k))$ is a pseudo subtree rooted at r_k, and $\mathcal{T}(x) := (G^1, \dots, G^p)$ is a partition of G into pseudo subtrees. We will refer to $\mathcal{T}(x)$ as the partition of G *induced by* x.

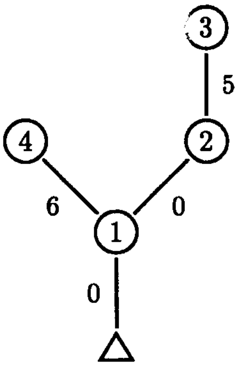


Figure 2.1: The maintenance problem of Example 2.3.3

Example 2.3.3 [Figure 2.1] Consider the maintenance problem $\mathcal{G} = (G, c, N)$ described by Figure 2.1, where the edge weights are given by $c(e) := 10$ for all $e \in E$. The allocation $x = (4, 5, 15, 16)$ is a core element, and the corresponding overflows are indicated next to the edges in the figure. By removing all the edges with zero overflows, we obtain the partition of G into the pseudo subtrees G^1 and G^2, where $N(G^1) = \{1, 4\}$, $N(G^2) = \{2, 3\}$, $r_1 = r$, and $r_2 = 1$. $\triangleleft$

Let $k(i)$ denote the index of the subtree to which $i \in N$ belongs, i.e., $i \in N(G^{k(i)})$. Let $\tilde{F}(i) := F(i) \cap V(G^{k(i)})$ be the *local followers* of i in his subtree

$G^{k(i)}$. Similarly, for $e \in E(G^k)$, for some subtree $1 \leq k \leq p$, let $\tilde{B}_e$ be that part of the branch B_e that belongs to G^k, i.e., $V(\tilde{B}_e) := V(B_e) \cap V(G^k)$, $N(\tilde{B}_e) := N(B_e) \cap N(G^k)$, and $E(\tilde{B}_e) := E(B_e) \cap E(G^k)$. The overflow of e, when restricted to the subtree G^k, is denoted $\tilde{O}_e(x) := \sum_{i \in N(\tilde{B}_e)}(x_i - c_i)$. In an analogous manner, for $1 \leq k \leq p$ and $i \in V(G^k)$, define $\tilde{P}_i$.

Proposition 2.3.4 *Let $\mathcal{T} = (G^1, \ldots, G^p)$ be a partition of G into pseudo subtrees, and let $\mathcal{G}^1, \ldots, \mathcal{G}^p$ be the corresponding induced maintenance problems.*

(i) Then

$$\prod_{k=1}^{p} C(c_{\mathcal{G}^k}) \subseteq C(c_{\mathcal{G}}),$$

where $(N(G^k), c_{\mathcal{G}^k})$ is the cost game corresponding to the restricted maintenance problem $\mathcal{G}^k$.

(ii) Let $x \in C(c_{\mathcal{G}})$, and suppose that $\mathcal{T} = (G^1, \ldots, G^p) := \mathcal{T}(x)$. Then

$$x \in \prod_{k=1}^{p} C(c_{\mathcal{G}^k}).$$

Proof. These results appear as Proposition 3.4 (i) and (ii), respectively, in Koster et al. (1998). We will give an alternative prove of (*ii*), thereby using the core characterization in Proposition 2.3.1(iii).

Let $1 \leq k \leq p$. Because $x \in C(c_{\mathcal{G}})$ satisfies $x(N) = c_{\mathcal{G}}(N)$, and since $\mathcal{T}$ has been constructed by removing only edges with zero overflows, it is clear that $x(N(G^k)) = c_{\mathcal{G}^k}(N(G^k))$. Also, $x^{N(G^k)} \geq 0$ follows from $x \in C(c_{\mathcal{G}})$ and Proposition 2.3.1.

We will complete the proof by showing that

$$\tilde{O}_i(x) = O_i(x) \geq 0$$

for all $i \in N(G^k)$, where the inequality follows directly from $x \in C(c_{\mathcal{G}})$ and Proposition 2.3.1(iii). In order to show the equality, note that, by (2.6) and the construction of $\mathcal{T}$, $\tilde{O}_i(x) = x_i - c_i = O_i(x)$ for any $i \in N(G^k)$ such that i is a leaf in G^k, since i must either be a leaf in G, or we must have $O_j(x) = 0$

for every $j \in \pi^{-1}(i)$. Then, for every $i \in N(G^k)$ such that i is not a leaf in G^k, $\tilde{O}_i(x) = (x_i - c_i) + \sum_{j \in \pi^{-1}(i) \cap \tilde{F}(i)} \tilde{O}_j(x) = (x_i - c_i) + \sum_{j \in \pi^{-1}(i)} O_j(x) = O_i(x)$. □

2.4 The core and the set of weighted down-home allocations

A well-known single-valued solution concept for cooperative cost games is the Shapley value (Shapley (1953)). For airport games and maintenance games there exists a nice expression for the Shapley value (cf. Littlechild and Thompson (1977), Dubey (1982), and Koster et al. (1998)). Roughly, the Shapley value of a maintenance game can be obtained through a dynamical process of uniformly distributing the costs of the edges from the root in the direction of leafs. In this section we will show this procedure and that by a simple adaptation of this algorithmic approach we obtain the class of weighted Shapley values (Kalai and Samet (1988)). First we will develop the dynamical approach that specifies a *weighted down-home allocation*. Then afterwards we conclude that it represents no more than a weighted Shapley value. Monderer et al. (1992) show, for the class of concave games, that the set of weighted Shapley values equals the core. Here we show, in a *constructive* way, that each core element of the maintenance game corresponds to a weighted Shapley value.

Definition 2.4.1 Let $\mathcal{G} = (G, c, N)$ be a maintenance problem, and let $\mathcal{B}(\mathcal{G})$ denote the set of *weight systems* for $\mathcal{G}$. Then $\beta := (\mathcal{T}, w) \in \mathcal{B}(\mathcal{G})$ iff

(i) $\mathcal{T} = (G^1, \dots, G^p)$ is a partition of G into pseudo subtrees,

(ii) $w \geq 0$,

(iii) $w(\tilde{F}(i)) > 0$ for all $i \in N$ such that $c_i > 0$.

Consider a maintenance problem $\mathcal{G} = (G, c, N)$ and some weight system $\beta \in \mathcal{B}(\mathcal{G})$. For each pseudo subtree G^k, interpret the nodes in $N(G^k)$ as the homes of the different players and the edges in $E(G^k)$ as the roads to the community center (r_k). The cost of a road is expressed as the number of days it takes (for one person) to paint the stripes on the road. The work is done by the players themselves according to the following rules[2]:

(i) Every worker keeps painting as long as the road from the community center to his home has not been completed.

(ii) Every worker does his job on an unfinished segment between the community center and his home.

(iii) Every worker starts painting at the same moment.

(iv) Every worker $i \in N$ paints with velocity w_i.

(v) Each worker paints as close to the community center as the rules (i)-(iv) permit him to.

We call the resulting allocation the *weighted down-home*[3] *allocation*, and denote it $\delta^\beta(\mathcal{G})$. It is given, for any player $i \in N$, by

$$\delta_i^\beta(\mathcal{G}) = \sum_{e \in E(\tilde{P}_i)} \frac{w_i}{w(N(\tilde{B}_e))} c(e). \tag{2.7}$$

Note that worker i only contributes towards the cost of the edges of his own subtree $k(i)$, i.e., the edges belonging to the path $\tilde{P}_i$.

Example 2.4.2 [Figure 2.2] Let $c(e) := 10$ for every $e \in E$. Also, let $\mathcal{T} := (G^1, G^2)$, where $N(G^1) := \{1, 2, 3\}$ and $N(G^2) := \{4\}$, and $w := (1, 1, 3, 1)^T$. For $\beta := (\mathcal{T}, w)$ we have $\delta^\beta(\mathcal{G}) = \left(2, 4\frac{1}{2}, 23\frac{1}{2}, 10\right)$. Player 1 only contributes to the cost of edge $(r, 1)$, so his total contribution is $10 \cdot \frac{1}{5} = 2$. Player 2 contributes to the cost of edge $(r, 1)$ and $(1, 2)$, with relative weights of $\frac{1}{5}$ and $\frac{1}{4}$, respectively, so his total contribution is $10 \cdot \frac{9}{20} = 4\frac{1}{2}$. Player 3 contributes at edge $(r, 1)$, $(1, 2)$, and $(2, 3)$, with relative weights of $\frac{3}{5}$, $\frac{3}{4}$, and 1, respectively, hence his total contribution is $10 \cdot \frac{47}{20} = 23\frac{1}{2}$. Player 4 is

[2]These rules are inspired by the *painting story* presented in Maschler et al. (1995).

[3]Koster et al. (1998) treat the weighted *home-down* allocation, which results by replacing "the community center" in (v) by "his home". The resulting allocation is related to a weighted version of the *constrained egalitarian solution* of Dutta and Ray (1989), see, e.g., Koster (1999) or Hokari (1998).

the only player in his pseudo subtree, and contributes the entire cost of the edge that he uses, i.e. 10. ◁

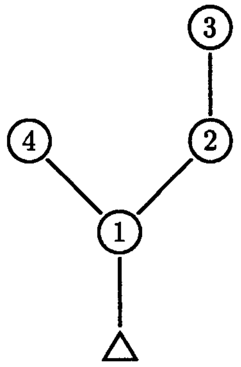


Figure 2.2: The maintenance problem of Example 2.4.2

From Proposition 2.3.1(iv) it follows that each down-home allocation specifies a core-element. But, as we are about to show, the converse also holds. For each core element x there is a weight system β such that the corresponding down-home allocation $\delta^{\beta}(\mathcal{G})$ equals x. We will show how such a weight system β is easily calculated for a given $x \in C(c_{\mathcal{G}})$.

Suppose that x is a down-home allocation. First of all the partition of the player set is derived from the partition of $\mathcal{G}$ into pseudo subtrees induced by x; this can be done by considering the overflows in the tree. Next the weights for the players are calculated for each separate subproblem. Since the weights can be calculated for each subtree separately, we can assume that the partition into pseudo subtrees of $\mathcal{G}$ with respect to x has only one element, or, equivalently, all the overflows are positive except at the edge that leaves the root. Then our objective is, if at all possible, to find a suitable vector of weights w such that for $\beta = (\{G\}, w) \in \mathcal{B}(\mathcal{G})$ we have $\delta^{\beta}(\mathcal{G}) = x$.

Without loss of generality we will assume that player 1 is the player directly connected to the root. The cost of the corresponding edge e_1 is covered by the collective of players N. First of all, with the interpretation of the weights as painting speeds, the edge e_1 is painted in

$$\frac{c(e_1)}{w(N)} = c(e_1)$$

units of time, if we assume that w is normalized such that $w(N) = 1$. Moreover, each of the painting players is finished with e_1 at the same time. In particular, if player 1 is painting at all (in case $x_1 > 0$) then he is also painting for $c(e_1)$ units of time. On the other hand he must complete x_1 by himself, at speed w_1, so we have the condition

$$\frac{x_1}{w_1} = c(e_1),$$

and thus

$$w_1 = \frac{x_1}{c(e_1)}.$$

Note that $c(e_1) > 0$, since $O_i(x) > 0$ for all $i \in \pi^{-1}(1)$. After having calculated this first weight, we proceed by consecutively assigning weights to each of the players in the sets $\pi^{-1}(1), \pi^{-1}(\pi^{-1}(1)), \ldots$, until even the leaf players have a weight. Basically we repeat the above type of reasoning.

Consider a player $i \notin \pi^{-1}(1)$. Then, according to x, his followers $F(i)$ contribute $O_i(x) > 0$ to the maintenance cost of the path from the root to his predecessor, player $\pi(i)$. Recall again the painting story. The speed at which the collective of players $F(i)$ operates on the path from r to $\pi(i)$ is given by the aggregate of the weights $w(F(i))$. Then the time that the group of players $F(i)$ needs to complete $O_i(x)$ is given by

$$\frac{O_i(x)}{w(F(i))}.$$

Similarly, it holds that the followers of $\pi(i)$, which include the followers of i, contribute $O_{\pi(i)}(x)$ to the path from the root to $\pi(\pi(i))$ ***plus*** the full cost of maintaining the edge $(\pi(i), \pi(\pi(i)))$. The collective of players $F(\pi(i))$ paints at speed $w(F(\pi(i)))$, which means that the time that they need to complete their part of the path from the root to $\pi(i)$ equals

$$\frac{O_{\pi(i)}(x) + c(e_{\pi(i)})}{w(F(\pi(i)))}.$$

This expression indicates the time that each of the individuals in $F(\pi(i))$ is working on the path from r to $\pi(i)$, and especially each of the players in $F(i)$. But then we must have the equality

$$\frac{O_i(x)}{w(F(i))} = \frac{O_{\pi(i)}(x) + c(e_{\pi(i)})}{w(F(\pi(i)))}.$$

This determines an iterative procedure for calculating all the weights $w(F(i))$ for each $i \in F(1)\backslash\{1\}$, since

$$w(F(i)) = w(F(\pi(i)))\frac{O_i(x)}{O_{\pi(i)}(x) + c(e_{\pi(i)})}$$

for all $i \in N$, and then we can use

$$w_i = w(F(i)) - \sum_{j\in\pi^{-1}(i)} w(F(j))$$

to calculate the individual weight for every $i \in N$.

Example 2.4.3 [Figure 2.3] As in earlier examples the maintenance costs of the different edges are all 10. Check that $x = (5, 13, 12)^T$ is a core element for $c_{\mathcal{G}}$. The numbers at the edges in Figure 2.3 denote the overflows corresponding to x. Firstly, observe that the partition $\mathcal{T}$ of $\mathcal{G}$ into pseudo subtrees induced by x is trivial. Assume that x is a down-home allocation: there is a vector of weights w with $w_i > 0$ for all $i \in \{1, 2, 3\}$ such that $\delta^{\beta}(\mathcal{G}) = x$ for $\beta = (\mathcal{T}, w) \in \mathcal{B}(\mathcal{G})$. Recall the painting story for the weighted down-home allocation. The players 1,2, and 3 respectively paint at velocities w_1, w_2, and w_3 at e_1 as long as $c(e_1) = 10$ is not completed. Furthermore, the contribution of player 1 and the overflows $O_2(x)$ and $O_3(x)$ determine the parts of $c(e_1)$ that are individually covered by the players 1,2 and 3, respectively. Given the velocities we can compute the time that the players need to finish these parts in three ways, as

$$\frac{x_1}{w_1}, \frac{O_2(x)}{w_2}, \quad \text{or} \quad \frac{O_3(x)}{w_3}.$$

These numbers are equal by the fact that all the players will continue painting on e_1 until it is finished, which implies that the finishing time of the collective of players equals the individual finishing times.

Since we are completely informed about the individual contribution of player 1 and the overflows corresponding to each branch emanating from the node of player 1, we must therefore have

$$\frac{5}{w_1} = \frac{3}{w_2} = \frac{2}{w_3},$$

and thus $w = (w_1, \frac{3}{5}w_1, \frac{2}{5}w_1)^T$. If w is required to be a vector in the unit simplex, we get $w_1 = \frac{1}{2}$ and $w = (\frac{1}{2}, \frac{3}{10}, \frac{2}{10})^T$. The reader may verify that indeed $\delta^{\beta}(\mathcal{G}) = x$ for $\beta = (\{G\}, w)$. $\triangleleft$

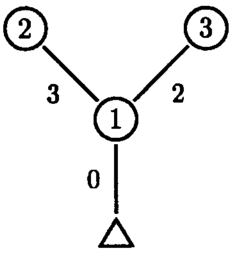


Figure 2.3: The tree of Example 2.4.3

Example 2.4.4 [Figure 2.4] All edges have equal maintenance cost 10. Consider the core element $x = (4, 12, 12, 12)^T$ of the corresponding 4-player maintenance game. The overflows corresponding to x are the numbers to the edges.

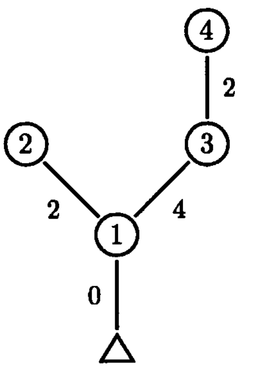


Figure 2.4: The maintenance problem of Example 2.4.4

The partition into pseudo subtrees by x is trivial. Assume that x is a down-home allocation, i.e. there is a vector $w \in \mathbf{R}^4$ with all positive coordinates such that for $\beta = (\{G\}, w)$ we have $\delta^\beta(\mathcal{G}) = x$. We will see that similar reasoning as in the Example 2.4.3 leads to conditions that determine w. Basically, the only difference with the situation in Example 2.4.3 is that it is not directly clear what are the individual contributions of players 3 and 4 at e_1. We are only able to monitor their aggregate efforts by means of $O_3(x)$. The same considerations as in the above example lead to the conclusion that players 1,2, and the collective of players 3 and 4 respectively finishes in $\frac{x_1}{w_1}$, $\frac{O_2(x)}{w_2}$ and $\frac{O_3(x)}{w_3+w_4}$ time units respectively. Since these numbers are all equal we have

$$\frac{4}{w_1} = \frac{2}{w_2} = \frac{4}{w_3 + w_4}.$$

Therefore, at this stage we are able to express w_2 and $w_3 + w_4$ in terms of w_1, i.e. $w_2 = \frac{1}{2}w_1, w_3 + w_4 = w_1$. If we require $w(N) = 1$, we get $w_1 = \frac{2}{5}$. This means that we only have to consider w_3 and w_4 since $w_2 = \frac{1}{2}w_1 = \frac{1}{5}$. Consider the path from the root to node 3. The players 3 and 4 reach node 3 at the same time. The time they need to complete the path from the root to node 3 equals the time for finishing e_1 plus the time necessary for completing e_3, i.e.

$$\frac{O_3(x)}{w_3 + w_4} + \frac{c(e_3)}{w_3 + w_4} = \frac{O_3(x) + c(e_3)}{w_3 + w_4}.$$

At this precise moment player 4 has completed exactly $O_4(x)$. Using the velocity of player 4, w_4, the time that player 4 must spend equals $\frac{O_4(x)}{w_4}$, and thus

$$\frac{O_4(x)}{w_4} = \frac{O_3(x) + c(e_3)}{w_3 + w_4}.$$

This means that

$$\frac{2}{w_4} = \frac{14}{w_3 + w_4} = 35,$$

from which we see that $w_4 = \frac{2}{35}$ and $w_3 = w_1 - w_4 = \frac{2}{5} - \frac{2}{35} = \frac{12}{35}$. Thus $w = \left(\frac{2}{5}, \frac{1}{5}, \frac{12}{35}, \frac{2}{35}\right)^T$. ◁

Now we will formalize the above ideas. For any core allocation x, we define a weight system $\beta \in \mathcal{B}(\mathcal{G})$ such that $x = \delta^\beta(\mathcal{G})$. First, find the partition $\mathcal{T} = (G^1, \dots, G^p)$ of G into pseudo subtrees induced by x. Then a weight vector w can be found by first, for all $i \in N$, calculating the sums

$$w(\tilde{F}(i)) = \begin{cases} 1 & \text{if } \pi(i) = r_{k(i)}, \\ \dfrac{\tilde{O}_i(x)}{\tilde{O}_{\pi(i)}(x) + c_{\pi(i)}} w(\tilde{F}(\pi(i))) & \text{else,} \end{cases} \tag{2.8}$$

in a recursive manner, and then the individual weight for a player $i \in N$ is given by

$$w_i = w(\tilde{F}(i)) - \sum_{j \in \pi^{-1}(i)} w(\tilde{F}(j)) \tag{2.9}$$

Proposition 2.4.5 *Let $x \in C(c_{\mathcal{G}})$. There exists $\beta := (\mathcal{T}, w) \in \mathcal{B}(\mathcal{G})$ such that $x = \delta^{\beta}(\mathcal{G})$, where $\mathcal{T} = \mathcal{T}(x)$, and w satisfies* (2.8) *and* (2.9).

Proof. First we show the existence part. Observe that $\mathcal{T}(x)$ exists and that if, for some $i \in N$, we have $|N(G^{k(i)})| = 1$, then (2.8) implies $w_i = 1$. To prove existence for w, it is therefore sufficient to show that

$$\tilde{O}_i(x) + c_i > 0 \text{ for all } i \in N \text{ such that } |N(G^{k(i)})| > 1. \tag{2.10}$$

Since $c_i \geq 0$ for all $i \in N$, and since $\tilde{O}_i(x) > 0$ for all $i \in N$ such that $\pi(i) \neq r_{k(i)}$, the only possible problem arises if $c_i = 0$ for a player i such that $\pi(i) = r_{k(i)}$. Suppose that this is the case. Then, since, by the construction of $\mathcal{T}$, $x(N(G^{k(i)})) = c_{\mathcal{G}^{k(i)}}(N(G^{k(i)}))$, we must have $\tilde{O}_j(x) = 0$ for all $j \in \pi^{-1}(i) \cap \tilde{F}(i)$, contradicting the fact that $\mathcal{T}$ is induced by x.

$\beta \in \mathcal{B}(\mathcal{G})$: Clearly, $\mathcal{T} = (G^1, \ldots, G^p)$ is a partition of G into pseudo subtrees. From (2.8), (2.10), and because $\tilde{O}_i(x) > 0$ if $\pi(i) \neq r_{k(i)}$, it follows that

$$w(\tilde{F}(i)) > 0 \qquad\qquad \text{for all } i \in N. \tag{2.11}$$

Also, for any $i \in N$, we have from (2.8) and (2.9) that

$$\begin{aligned} w_i &= w(\tilde{F}(i)) \left\{ 1 - \sum_{j \in \pi^{-1}(i) \cap \tilde{F}(i)} \frac{\tilde{O}_j(x)}{\tilde{O}_i(x) + c_i} \right\} \\ &= w(\tilde{F}(i)) \frac{\tilde{O}_i(x) + c_i - \sum_{j \in \tilde{F}(i) \setminus \{i\}} (x_j - c_j)}{\tilde{O}_i(x) + c_i} \\ &= w(\tilde{F}(i)) \frac{\sum_{j \in \tilde{F}(i)} (x_j - c_j) + c_i - \sum_{j \in \tilde{F}(i) \setminus \{i\}} (x_j - c_j)}{\tilde{O}_i(x) + c_i} \\ &= w(\tilde{F}(i)) \frac{x_i}{\tilde{O}_i(x) + c_i} \geq 0, \end{aligned} \tag{2.12}$$

where the last inequality follows from (2.11) and (2.10), and because Proposition 2.3.1 and $x \in C(c_{\mathcal{G}})$ imply $x \geq 0$.

Finally we show that $x = \delta^{\beta}(\mathcal{G})$. For any $i \in N$ it follows from (2.12) that

$$x_i = w_i \frac{\tilde{O}_i(x) + c_i}{w(\tilde{F}(i))}. \tag{2.13}$$

For any $j \in V(\tilde{P}_i)$, let

$$t_j := \begin{cases} 0 & \text{if } j = r_{k(i)}, \\ \dfrac{\tilde{O}_j(x) + c_j}{w(\tilde{F}(j))} & \text{otherwise.} \end{cases} \tag{2.14}$$

From (2.13) and (2.14) follows

$$x_i = w_i t_i = w_i \sum_{j \in N(\tilde{P}_i)} (t_j - t_{\pi(j)}). \tag{2.15}$$

Now note that, for any $j \in N(\tilde{P}_i)$,

$$t_j - t_{\pi(j)} = \frac{c_j}{w(\tilde{F}(j))}. \tag{2.16}$$

If $\pi(j) = r_{k(i)}$, then $\tilde{O}_j(x) = 0$, so (2.16) follows from (2.14). Else

$$\begin{aligned} t_j - t_{\pi(j)} &= \frac{\tilde{O}_j(x) + c_j}{w(\tilde{F}(j))} - \frac{\tilde{O}_{\pi(j)}(x) + c_{\pi(j)}}{w(\tilde{F}(\pi(j)))} \\ &= \frac{\tilde{O}_j(x) + c_j - \tilde{O}_j(x)}{w(\tilde{F}(j))} = \frac{c_j}{w(\tilde{F}(j))}, \end{aligned}$$

where the second equality follows from (2.8). To complete the proof, we rewrite (2.15) as

$$x_i = w_i t_i = w_i \sum_{j \in N(\tilde{P}_i)} \frac{c_j}{w(\tilde{F}(j))} = \sum_{e \in E(\tilde{P}_i)} \frac{c(e)}{w(\tilde{B}_e)},$$

which is exactly (2.7). □

Theorem 2.4.6 *The set of all down-home allocations equals the core of a maintenance game, i.e.* $\{\delta^\beta(\mathcal{G}) \mid \beta \in \mathcal{B}(\mathcal{G})\} = C(c_\mathcal{G})$.

Proof. That the weighted down-home allocations form a superset of $C(c_\mathcal{G})$ follows from Proposition 2.4.5. To show the inclusion, suppose $\beta \in \mathcal{B}(\mathcal{G})$. The proof is complete by first noting that $\delta^\beta(\mathcal{G}^k) \in C(c_{\mathcal{G}^k})$ for every $k = 1, \dots, p$ by (iv) in Proposition 2.3.1, and from Proposition 2.3.4(i). □

We will now show that the weighted down-home allocations are related to the *weighted Shapley value*, as defined byKalai and Samet (1987). In Monderer et al. (1992), it is shown in a non-constructive way that the set of weighted Shapley values equals the core for convex games. In order to define this generalization of the ordinary Shapley value, we need to make the following definition.

Definition 2.4.7 Call an S-weight system for the set of players N an ordered pair $\mu := (\mathcal{S}, \lambda)$, where $\mathcal{S} = (S_1, \dots, S_q)$ is an ordered partition of the player set N, and $\lambda^{S_\ell} \in \mathbf{R}^{S_\ell}_{++} \cap \Delta(S_\ell)$ for all $\ell = 1, \dots, q$. Let $\mathcal{M}(N)$ be the set of all S-weight systems for N.

Let $\mu \in \mathcal{M}(N)$. Define for each $S \subseteq N$, $m(S) := \min\{j : S_j \cap S \neq \emptyset\}$, and let $\bar{S} := S \cap S_{m(S)}$. The most important players, i.e. those in $\bar{S}$, will carry the entire cost. Then the *weighted Shapley value*[4] corresponding to μ is determined as the linear operator $\Phi^\mu : \Gamma^N \to \mathbf{R}^N$ such that for all $S \subseteq N$,

$$\Phi^\mu_i(u_S) = \begin{cases} \dfrac{\lambda_i}{\lambda(\bar{S})} & \text{if } i \in \bar{S}, \\ 0 & \text{otherwise.} \end{cases} \tag{2.17}$$

In the unanimity game u_S, the importance of the players depend on how they are "ranked", i.e. where they are located in the ordered collection $\mathcal{S}$. In the case of our cost game c_G, because of (2.5), we only need to consider the (dual) unanimity games corresponding to users of edges, i.e. the games $u^*_{N(B_e)}$ for all $e \in E$. If, for some $e \in E$ and $i \in N$, we have $i \in N(B_e)$, we say that i is *a user of e*. For some $e \in E$, let

$$S(e) := N(B_e) \cap S_{\min\{j | N(B_e) \cap S_j \neq \emptyset\}},$$

and if $i \in S(e)$, we say that i is a *senior user of e*. If i is a user, but not a senior one, of e, there must exist some $j \neq i$ such that $j \in S(e)$, and we say that *i is dominated by j at e.* The weighted Shapley value for the dual unanimity game corresponding to edge e is given by

$$\Phi^\mu_i(u^*_{N(B_e)}) = \begin{cases} \dfrac{\lambda_i}{\lambda(S(e))} & \text{if } i \in S(e), \\ 0 & \text{otherwise.} \end{cases} \tag{2.18}$$

[4]The ordinary Shapley value is the special case of (2.17) where $\mathcal{S} = (N)$, and all players have equal weights.

Then, since the weighted Shapley value is additive, and because of (2.5), we have

$$\Phi_i^{\mu}(c_{\mathcal{G}}) = \sum_{e \in E} \Phi_i^{\mu}(u^*_{N(B_e)}).$$

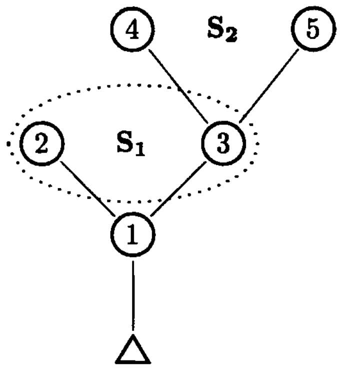


Figure 2.5: The maintenance problem of Example 2.4.8

Example 2.4.8 [Figure 2.5] Let $c(e) := 10$ for all $e \in E$. Also, let $\mathcal{S} := (\{2,3\},\{1,4,5\})$ and $\lambda := \left(\frac{1}{4},\frac{1}{2},\frac{1}{2},\frac{1}{4},\frac{1}{2}\right)^T$, hence $\mu = (\mathcal{S},\lambda) \in \mathcal{M}(N)$. The corresponding weighted Shapley value is $\Phi^{\mu}(c_{\mathcal{G}}) = (0,15,15,10,10)^T$. Player 1 pays nothing, since he is not among the senior users of any edge. Players 2 and 3 dominate all other players at edge $(r,1)$, and since they both have the same weight, they both pay 5 here. Only player 2 uses $(1,2)$, so he pays for this edge alone. Since he does not use any other edge except $(r,1)$, his total contribution is $5+10=15$. Player 3 dominates all other players at $(1,3)$, and since he is not using any other edge except $(r,1)$, his total contribution is $5+10=15$. Players 4 and 5 are dominated by other players at all edges that they use, except at the edges e_4 and e_5, respectively, where they make up the entire set of senior users, and therefore they contribute 10 each. $\triangleleft$

Theorem 2.4.9

(*i*) *For any $\beta \in \mathcal{B}(\mathcal{G})$, there exists $\mu \in \mathcal{M}(N)$ such that $\Phi^{\mu}(c_{\mathcal{G}}) = \delta^{\beta}(\mathcal{G})$.*

(*ii*) *For any $\mu \in \mathcal{M}(N)$, there exists $\beta \in \mathcal{B}(\mathcal{G})$ such that $\Phi^{\mu}(c_{\mathcal{G}}) = \delta^{\beta}(\mathcal{G})$.*

Proof. (*i*) Let $\beta = (\mathcal{T}, w) \in \mathcal{B}(\mathcal{G})$ for some maintenance problem $\mathcal{G}$. Note that the elements of $\mathcal{T} = (G^1, \ldots, G^p)$ can be ordered arbitrarily without affecting $\delta^\beta(\mathcal{G})$, and we choose an ordering such that $k(i) < k(j) \Rightarrow j \notin N(P_i)$ for any pair $i, j \in N$. Let, for every $k = 1, \ldots, p$, $S_k := \{i \in N(G^k) \mid w_i > 0\}$ and $S_{p+k} := \{i \in N(G^k) \mid w_i = 0\}$. The ordered collection $(S_1, \ldots, S_p, S_{p+1}, \ldots, S_{2p})$ contains q nonempty elements, where $p \leq q \leq 2p$, and let $\mathcal{S} := (S_1, \ldots, S_q)$ be the ordered collection obtained by deleting the empty elements. Also, for every $i \in N$, let

$$\lambda_i := \begin{cases} \dfrac{w_i}{w(S_{\ell(i)})} & \text{if } w_i > 0, \\ \dfrac{1}{|S_{\ell(i)}|} & \text{otherwise,} \end{cases} \tag{2.19}$$

where $\ell(i) = \ell$ if and only if $i \in S_\ell$. It is easily seen that $\mu := (\mathcal{S}, \lambda) \in \mathcal{M}(N)$. For any $i \in N$, we have

$$\begin{aligned} \Phi_i^\mu(c_{\mathcal{G}}) &= \sum_{\substack{e \in E \\ S(e) \ni i}} \frac{\lambda_i}{\lambda(S(e))} c(e) = \sum_{\substack{e \in E(\tilde{P}_i) \\ S(e) \ni i}} \frac{\lambda_i}{\lambda(S(e))} c(e) \\ &= \sum_{\substack{e \in E(\tilde{P}_i) \\ S(e) \ni i}} \frac{w_i}{w(S(e))} c(e) = \sum_{e \in E(\tilde{P}_i)} \frac{w_i}{w(N(\tilde{B}_e))} c(e) = \delta_i^\beta(\mathcal{G}). \end{aligned}$$

The first equality follows from (2.5), the additivity of the weighted Shapley value, and (2.18). The second equality follows from the fact that we can have $i \in S(e)$ only if $e \in E(\tilde{P}_i)$. Suppose, on the contrary, that $i \in S(e)$ for some $e \in E \setminus E(\tilde{P}_i)$. Since we can have $i \in S(e)$ only if i is a user of e, we must have $e \in E(P_i)$, i.e., e is on the path between i and the *global* root r, but does not belong to the local subtree to which i belongs. Then, by the construction of $\mathcal{S}$, we must have $N(B_e) \cap S_j \neq \emptyset$ for some $j < \ell(i)$, implying $i \notin S(e)$, a contradiction. In order to prove the third equality, it is sufficient to show that if $i \in S(e)$ for some $i \in N$ and $e \in E(\tilde{P}_i)$ such that $c(e) > 0$, then $\lambda_i = \frac{w_i}{w(S_{\ell(i)})}$, and hence $\lambda(S(e)) = \frac{w(S(e))}{w(S_{\ell(i)})}$. Suppose that this is not true. Then $w_i = 0$ by (2.19), and $\beta \in \mathcal{B}(\mathcal{G})$ implies that there exists some $j \in N(\tilde{B}_e)$ such that $w_j > 0$. Then, by the construction of $\mathcal{S}$, $i \notin S(e)$, a contradiction. The fourth equality follows because, for any $e \in E$ and $i \in N$, $i \in N(\tilde{B}_e) \setminus S(e)$ implies $w_i = 0$ (by the construction of $\mathcal{S}$), and hence $w(N(\tilde{B}_e)) = w(S(e))$. The last equality follows from (2.7).

(*ii*) Let $\mu = (\mathcal{S}, \lambda) \in \mathcal{M}(N)$ for some maintenance problem $\mathcal{G}$. We construct $\mathcal{T}$, i.e., a partition of G into pseudo subtrees, by applying Algorithm 2.4.10.

Algorithm 2.4.10

Initialization
Let $S'_m := S_m$ for every $m = 1, \dots, q$, $w := \lambda$, and $\ell := 1$.

Main step
Repeat
    For $i \in S'_\ell$ **do**
        For $j \in N(P_i)$ **do**
            If $\ell(j) > \ell(i)$ **then**
                $S'_{\ell(i)} := S'_{\ell(i)} \cup \{j\}$
                $S'_{\ell(j)} := S'_{\ell(j)} \setminus \{j\}$
                $w_j := 0$
    $\ell := \ell + 1$
until $\ell > q$

The algorithm will give as output the ordered set of coalitions $S'_1, \dots, S'_q$. Suppose that this ordered set has q' nonempty members. Delete the empty members, and for every $1 \leq \ell \leq q'$, let $G^\ell_1, \dots, G^\ell_{i_\ell}$ be the collection of pseudo subtrees corresponding to maximal connected, with respect to G, components of S'_ℓ. Clearly, the ordered set

$$G^1_1, \dots, G^1_{i_1}, G^2_1, \dots, G^2_{i_2}, \dots, G^{q'}_1, \dots, G^{q'}_{i_{q'}}$$

is a partition of G into pseudo subtrees. Let p be the number of members of this partition, re-index, and set $\mathcal{T} := (G^1, \dots, G^p)$. Since $\lambda \geq 0$, we have $w \geq 0$. Then it follows that $\beta := (\mathcal{T}, w) \in \mathcal{B}(\mathcal{G})$ since, for any $i \in N$, $w_i = 0$ implies, by Algorithm 2.4.10, that there exists some $j \in \tilde{F}(i) \setminus \{i\}$ such that

$w_j > 0$. Now, for every $i \in N$,

$$\begin{aligned} \delta_i^{\beta}(\mathcal{G}) &= \sum_{e \in E(\tilde{P}_i)} c(e) \frac{w_i}{w(N(\tilde{B}_e))} = \sum_{\substack{e \in E(\tilde{P}_i) \\ S(e) \ni i}} c(e) \frac{\lambda_i}{\lambda(S(e))} \\ &= \sum_{\substack{e \in E(P_i) \\ S(e) \ni i}} c(e) \frac{\lambda_i}{\lambda(S(e))} = \sum_{\substack{e \in E \\ S(e) \ni i}} c(e) \frac{\lambda_i}{\lambda(S(e))} = \phi_i^{\mu}(c_{\mathcal{G}}). \end{aligned} \tag{2.20}$$

The first equality follows from (2.7), and the second equality follows from the fact that $w_i = 0$ if $i \notin S(e)$ for some $e \in E(\tilde{P}_i)$, and since $w(N(\tilde{B}_e)) = \lambda(S(e))$ for every $e \in E$. To see that the latter equality is correct, consider some $e \in E$. After applying Algorithm 2.4.10, the nodes in $N(P_j) \cap N(B_e)$ will be included in $S'_{\ell(j)}$ for every $j \in S(e)$. Hence the node set $\cup_{j \in S(e)}(N(P_j) \cap N(B_e))$ will be connected, with respect to G, and we must therefore have $S(e) \subseteq N(\tilde{B}_e)$. Also, $j \in N(\tilde{B}_e) \setminus S(e)$ implies $w_j = 0$, and $j \in S(e)$ implies $w_j = \lambda_j$, hence we obtain the desired result.

The third equality in (2.20) follows because $e \in E(P_i) \setminus E(\tilde{P}_i)$ implies, from Algorithm 2.4.10, that $N(B_e) \cap S_j \neq \emptyset$ for some $j < \ell(i)$, i.e., i is dominated by the members of S_j, hence $i \notin S(e)$.

The fourth equality in (2.20) follows because $e \in E \setminus E(P_i)$ implies that i is not a user of e, hence $i \notin S(e)$, and the last equality follows from (2.5), the additivity of the weighted Shapley value, and (2.18). □

Example 2.4.11 [Figure 2.2] Consider the maintenance problem in Example 2.4.2, and the weight system $\beta = (\mathcal{T}, w) \in \mathcal{B}(\mathcal{G})$, where $\mathcal{T} = (G^1, G^2)$ and $w = (1, 1, 3, 1)^T$. Here, the corresponding $\mu = (\mathcal{S}, \lambda) \in \mathcal{M}(N)$ is uniquely given by $\mathcal{S} := (\{1,2,3\}, \{4\})$ and $\lambda = \left(\frac{1}{5}, \frac{1}{5}, \frac{3}{5}, 1\right)^T$. ◁

Example 2.4.12 [Figure 2.5] Consider the maintenance game in Example 2.4.8, and the weight system $\mu = (\mathcal{S}, \lambda) \in \mathcal{M}(N)$, where $\mathcal{S} = (\{2,3\}, \{1,4,5\})$ and $\lambda = \left(\frac{1}{4}, \frac{1}{2}, \frac{1}{2}, \frac{1}{4}, \frac{1}{2}\right)^T$. By applying Algorithm 2.4.10, we obtain the partition $\mathcal{S}' = (\{1,2,3\}, \{4,5\})$ of the player set, and the weight vector $w = \left(0, \frac{1}{2}, \frac{1}{2}, \frac{1}{4}, \frac{1}{2}\right)^T$. Note that player 1 has been absorbed by the partition member containing 2 and 3, since 1 is dominated by these two players, and that, accordingly, his weight is now zero. By taking maximal connected subsets of each partition member, we obtain a partition of G into pseudo subtrees, equal to $\mathcal{T} = (G^1, G^2, G^3)$, where $N(G^1) = \{1, 2, 3\}$, $N(G^2) = \{4\}$, and $N(G^3) = \{5\}$. $\triangleleft$

We now know, from Theorem 2.4.9, that our weighted down-home allocations are equivalent to weighted Shapley values, and, from Theorem 2.4.6, that the set of weighted down-home allocations equals the core, hence the following result.

Corollary 2.4.13 *The core of the maintenance game* $(N, c_{\mathcal{G}})$ *equals the set of weighted Shapley values, i.e.* $\{\Phi^{\mu}(c_{\mathcal{G}}) \mid \mu \in \mathcal{M}(N)\} = C(c_{\mathcal{G}})$.

Monderer et al. (1992) show a more general result, that the set of all weighted Shapley values equals the core of *any* concave cost game. However, in proving this they needed a fixed point theorem.

2.5 The core and the set of weighted neighbour-home allocations

In the case of the weighted down-home allocation, the players have an obligation to help their neighbours (predecessors), since they are required to start working from the community center towards their own home. A less extreme social obligation results by applying rules (i)-(iv) in Section 2.4, as well as (v) and (vi) below. The resulting allocation will be called the *neighbour-home* allocation.

(v) If, for any worker $i \in N$, the road between $r_{k(i)}$ and $\pi(i)$ has not been finished yet, then i is working outside his own edge e_i.

(vi) Each worker paints as close to his home as the rules (i)-(v) permit him to.

The algorithm in Maschler et al. (1995) returns a special case of the weighted neighbour-home allocation, the nucleolus, where $\mathcal{T} = \{G\}$ and $w_i = \frac{1}{|N|}$ for all $i \in N$. We will show, analogous to the treatment in Section 2.4 for the weighted down-home allocation, that the set of weighted neighbour-home allocations equals the core, when the weight systems vary over the set $\mathcal{B}(\mathcal{G})$. In order to do this, we need to present the scheme implied by rules (i)-(v) in a more formal manner, and this is done in Algorithm 2.5.1. Let $\beta = (\mathcal{T}, w) \in \mathcal{B}(\mathcal{G})$, where $\mathcal{T} = (G^1, \dots, G^p)$. The neighbour-home allocation, denoted $\eta^\beta(\mathcal{G})$, is obtained by, for each of the restricted maintenance problems $\mathcal{G}^k$, $1 \leq k \leq p$, applying Algorithm 2.5.1 to the restricted maintenance problem $\mathcal{G}^k$.

Let $x(e,q) \in [0, c(e)]$ be the part of the cost of edge $e \in E(G^k)$ which is paid before stage q. Let $E_q \subseteq E(G^k)$ be the subset of edges whose cost is covered at stage q, and let $E(q) := \cup_{j<q} E_j$. Let $e(i,q)$ be the edge to which player i is contributing in stage q, and let $S(e,q) := \{i \in N(G^k) \mid e(i,q) = e\}$ be the set of players contributing to edge e in stage q. Let $Q(i)$ denote the first stage in which i stops contributing.

Algorithm 2.5.1

STEP 0

$q := 1$

$x(e,1) := 0$ for all $e \in E(G^k)$

$E(1) := \emptyset$

$$e(i,1) := \begin{cases} e_{\pi(i)} & \text{if } \pi(i) \neq r_k \\ e_i & \text{otherwise} \end{cases}$$

STEP 1

For any $e \in E(G^k) \setminus E(q)$ such that $S(e,q) \neq \emptyset$, it would take

$$t(e,q) := \frac{c(e) - x(e,q)}{w(S(e,q))}$$

units of time to finish paying for edge e. Thus, the first edge will be finished after $t(q) := \min\{t(e,q) \mid e \in E(G^k) \setminus E(q) \text{ and } S(e,q) \neq \emptyset\}$ units of time. Then $w(S(e,q))t(q)$ is the fraction of an edge $e \in E(G^k) \setminus E(q)$ which is constructed at stage q, and therefore $x(e,q+1) := x(e,q) + w(S(e,q))t(q)$. Let $E_q := \{e \in E(G^k) \setminus E(q) \mid t(e,q) = t(q)\}$ be the subset of edges finished at stage q, and let $E(q+1) := E(q) \cup E_q$ be the subset of edges finished before stage $q+1$. Consider every $i \in S(e,q)$ for every $e \in E_q$. If there exists an unfinished edge between

$e = e(i,q)$ and the root, i.e. $f \preceq e$ such that $f \in E(G^k) \setminus E(q+1)$, then choose such an f as close to e as possible, and set $e(i,q+1) := f$. If such an edge does not exist, and if i's own edge is not finished, i.e. $e_i \in E(G^k) \setminus E(q+1)$, set $e(i,q+1) := e_i$. Otherwise, set $Q(i) := q$ and $\eta_i^\beta(\mathcal{G}) := \sum_{q=1}^{Q(i)} t(q)w_i$.

STEP 2

If $E(q+1) = E(G^k)$, terminate. Otherwise, set $q := q+1$, and repeat step 1.

We will first demonstrate the algorithm by an example.

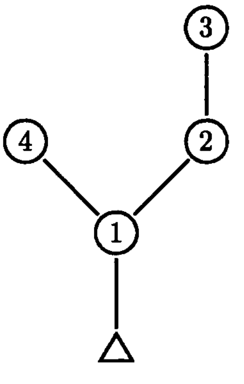


Figure 2.6: The maintenance problem of Example 2.5.2

Example 2.5.2 [Figure 2.6] This example is identical to Example 2.4.2, where $\beta = ((G^1, G^2), (1,1,3,1))$, $N(G^1) = \{1,2,3\}$, and $N(G^2) = \{4\}$, and we have $\eta^\beta(\mathcal{G}) = (4,4,22,10)^T$. Player 4 is alone in his pseudo subtree G^2, so he will contribute the entire cost of the edge $(1,4)$, i.e. 10. For pseudo subtree G^1 we apply Algorithm 2.5.1. Initially, $e(1,1) = e(2,1) = (r,1)$ and $e(3,1) = (1,2)$. The first edge is finished after $t(1) = \min\{\frac{10}{2}, \frac{10}{3}\} = \frac{10}{3} = t((1,2),1)$ units of time, and the set of edges finished in the first stage is $E_1 = \{(1,2)\}$. Now $e(3,2) = (r,1)$ and $S((r,1),2) = \{1,2,3\}$, i.e. all three players will be contributing to edge $(r,1)$ in the second stage. Then $t(2) = t((r,1),2) = \frac{10-\frac{10}{3}\cdot(1+1)}{5} = \frac{2}{3}$, and $E_2 = \{(r,1)\}$. Players 1 and 2 stop contributing after the second stage, i.e. $Q(1) = Q(2) = 2$, and they each contribute, in total, $1 \cdot (\frac{10}{3} + \frac{2}{3}) = 4$. Player 3 now starts contributing to his own edge, i.e. $e(3,3) = (2,3)$. He will finish this edge in $t(3) = \frac{10}{3}$ units of time, and then stop contributing ($Q(3) = 3$). His total contribution is $3 \cdot \left(\frac{10}{3} + \frac{2}{3} + \frac{10}{3}\right) = 22$. Since all the edges have been finished after stage 3, the algorithm terminates. $\triangleleft$

Now we turn to the following question: Given a core element y, can we find a weight system $\beta = (\mathcal{T}, w) \in \mathcal{B}(\mathcal{G})$ such that $y = \eta^{\beta}(\mathcal{G})$? It turns out that the answer is yes, and proposition 2.3.4(ii) suggests that we choose $\mathcal{T} := \mathcal{T}(x)$. We will now illustrate, using two examples, how the weight vector w can be found.

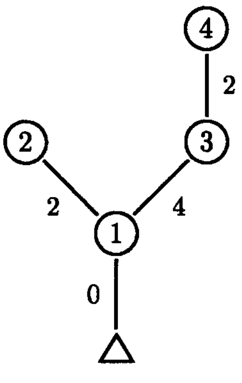


Figure 2.7: The tree of Example 2.5.3

Example 2.5.3 [Figure 2.7] This example is identical to Example 2.4.4, and the allocation $y = (4, 12, 12, 12)^T$ is a core element, since all the overflows, as shown in the figure, are non-negative. The partition $\mathcal{T}(y)$ is trivial, since the only edge with zero overflow is $(r, 1)$. First, note that edge e_1 will be finished after $\frac{4}{w_1}$ units of time. Moreover, players 2 and 3 will be contributing at this edge until it is finished, and will return home (to finish their own edges) exactly when this is the case. In order to calculate how long 2 and 3 will be contributing at edge e_1, we need to find their *far-away contributions*, i.e. how much they contribute at edges other than their own. We will do this by first finding their *home contributions*, i.e. how much they contribute at their own edges. Player 2's home contribution is obviously given by the cost of his own edge, i.e. 10, since he has no followers other than himself. Thus his far-away contribution, i.e. the amount that he will contribute at edge e_1, is 12 - 10 = 2. For player 3 the picture is more complicated, since he has a follower, player 4. Rule (vi) implies that if 4 contributes anything above the cost of his own edge, this contribution will first be used to finish edge e_3. This is indeed the case, since player 4 contributes 12 - 10 = 2 in excess of the cost of his own edge. The home contribution of player 3 will thus be only 10 - 2 = 8. Since he contributes 12 in total, his far-away contribution will be 12 - 8 = 4. To sum up, we know now that players 1, 2, and 3 contributes 4, 2, and 4, respectively, at edge e_1. This implies

$$\frac{4}{w_1} = \frac{2}{w_2} = \frac{4}{w_3}. \tag{2.21}$$

Player 4 will be contributing at e_3 until this edge is finished. His contribution at this edge is 2, as we stated above. Player 3 will stop contributing at all exactly when his own edge is finished, at which point he will have contributed 12. This implies

$$\frac{12}{w_3} = \frac{2}{w_4}. \tag{2.22}$$

A weight vector that satisfies both (2.21) and (2.22) is $w := (4, 2, 4, \frac{4}{6})^T$. ◁

To formalize the notions of home and far-away contributions, let $y = \eta^{\beta}(\mathcal{G})$ for some weight system $\beta \in \mathcal{B}(\mathcal{G})$, and let $i \in N(G^k)$, $1 \leq k \leq p$. Note that the contribution of the players in $\tilde{F}(i) \setminus \{i\}$ at or below e_i will be given by what they contribute in excess of the cost of their own edges, i.e. by $\sum_{j \in \tilde{F}(i) \setminus \{i\}} (y_j - c_j)$. Because of rule (vi), this excess contribution will first be used at edge e_i. Player i will cover the remaining part $c_i - \sum_{j \in \tilde{F}(i) \setminus \{i\}} (y_j - c_j)$ of the cost of his own edge, if this expression is positive. Hence, player i's home contribution is given by

$$h_i(y) := \left(c_i - \sum_{j \in \tilde{F}(i) \setminus \{i\}} (y_j - c_j) \right)_+ = \left(c_i - \sum_{j \in \pi^{-1}(i) \cap \tilde{F}(i)} \tilde{O}_j(y) \right)_+ ,$$

and his far-away contribution is $f_i(y) := y_i - h_i(y)$. Next we will consider an example where some players contribute nothing, which makes finding the weight vector slightly more complicated.

Example 2.5.4 [Figure 2.8] A core element is given by $y = (0, 12, 16, 0, 16, 16)^T$. We set the weights of players that does not make any contribution, to zero, i.e. $w_1 := w_4 := 0$. Player 2 has no followers other than himself, he will

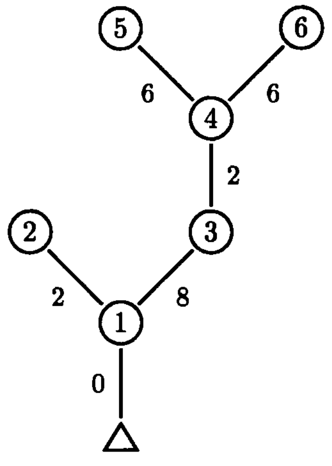


Figure 2.8: The maintenance problem of Example 2.5.4

have to contribute to the entire cost of his own edge e_2, i.e. $h_2(y) = 10$. The remaining 12 - 10 = 2 (= $f_2(y)$) that he contributes, will be towards the cost of edge e_1. Player 3 will also contribute at this edge, but how much? The answer can be found by noting that the followers of 3 (except himself), i.e. players 4, 5 and 6, contribute $0 + 16 + 16 = 32$, while the total cost of their own edges is only 30. Hence we have $h_3(y) = 10 - (32 - 30) = 8$ and $f_3(y) = 16 - 8 = 8$. Players 2 and 3 will return home at exactly the same time, i.e. when edge e_1 is finished. This will happen after

$$\frac{f_2(y)}{w_2} = \frac{2}{w_2} = \frac{f_3(y)}{w_3} = \frac{8}{w_3} \tag{2.23}$$

units of time. Note that the weights of players 2 and 3 are not related to the weight of the player in front of them, as was the case in the Example 2.5.3.

We have $h_5(y) = h_6(y) = 10$, since neither player 5 nor player 6 have followers other than themselves, and therefore $f_5(y) = f_6(y) = 16 - 10 = 6$. Because of rule (v), they cannot return home until the players in front of them have all finished. The last such player to finish will be the closest one that makes a positive contribution, i.e. player 3, who finishes after $\frac{16}{w_3}$ units of time. Our weight vector must therefore satisfy

$$\frac{6}{w_5} = \frac{6}{w_6} = \frac{16}{w_3}. \tag{2.24}$$

A weight vector that assigns weight zero to players that does not contribute anything, as well as satisfies (2.23) and (2.24), is given by $w = (0, 4, 16, 0, 6, 6)^T$. ◁

For any $i \in N(G^k)$, let $\pi^+(i)$ be the first predecessor of i in G^k such that $y_i > 0$. If no such predecessor exists, let $\pi^+(i) := r_k$. Also, let $N^+(G^k) := \{i \in N(G^k) \mid y_i > 0\}$. Note that if $i \in N(G^k)$ is such that $\pi(i) \neq r_k$ and $\tilde{O}_i(y) > 0$, then he will contribute a nonzero amount to the cost of the edges in $E(\tilde{P}_{\pi^+(i)})$, and will return home exactly when all the edges in $E(\tilde{P}_i) \setminus \{e_i\}$ have been finished. Since $\frac{f_i(y)}{w_i}$ is the total time that player i spends contributing to edges other than his own, we have

$$\pi^+(i) = \pi^+(j) \neq r_k \Rightarrow \frac{f_i(y)}{w_i} = \frac{f_j(y)}{w_j} \qquad \text{for all } i, j \in N^+(G^k). \tag{2.25}$$

Also, if a player contributes to the cost of the edges of his predecessors, he will return home exactly when the last one of his predecessors stops contributing, i.e.,

$$\frac{f_i(y)}{w_i} = \frac{y_{\pi^+(i)}}{w_{\pi^+(i)}} \qquad \text{for all } i \in N^+(G^k) \text{ such that } \pi^+(i) \neq r_k. \tag{2.26}$$

Let, for $k = 1, \dots, p$, $B^k(x) := \{i \in N(G^k) : \pi^+(i) = r_k\}$, i.e., the set of players who have no contributing players between themselves and the root.

Proposition 2.5.5 *Let $x \in C(c_{\mathcal{G}})$. Then there exists $\beta := (\mathcal{T}, w) \in \mathcal{B}(\mathcal{G})$ such that $\mathcal{T} = \mathcal{T}(x)$, and w satisfies, for every $k = 1, \dots, p$,*

$$\frac{f_i(x)}{w_i} = \frac{f_j(x)}{w_j} \qquad \forall i \neq j \in N^+(G^k) \cap B^k(x), \tag{2.27}$$

$$\frac{f_i(x)}{w_i} = \frac{x_{\pi^+(i)}}{w_{\pi^+(i)}} \qquad \forall i \in N^+(G^k) \setminus B^k(x), \tag{2.28}$$

$$w_i = 0 \qquad \forall i \in N(G^k) \setminus N^+(G^k). \tag{2.29}$$

Moreover, $x = \eta^\beta(\mathcal{G})$.

Proof.

Claim 1: Existence.

Clearly, $\mathcal{T}(x)$ exists. Let $1 \leq k \leq p$. In order to show that (2.27)-(2.29) have a solution, note that, for every $i \in N(G^k)$,

$$\begin{aligned} f_i(x) = x_i - h_i(x) &= x_i - \left(c_i - \sum_{j \in \tilde{F}(i) \setminus \{i\}} (x_j - c_j) \right)_+ \\ &= x_i - \left(c_i + x_i - c_i - \sum_{j \in \tilde{F}(i)} (x_j - c_j) \right)_+ \\ &= x_i - \left(x_i - \tilde{O}_i(x) \right)_+ \end{aligned}$$

The construction of $\mathcal{T}$ implies, for every $i \in N(G^k)$, that $\tilde{O}_i(x) > 0$ if $\pi(i) \neq r_k$, hence

$$\pi(i) \neq r_k \text{ and } x_i > 0 \Rightarrow f_i(x) > 0. \tag{2.30}$$

A solution can be found by arbitrarily fixing $w_{i^*} > 0$ for some $i^* \in N^+(G^k) \cap B^k(x)$. Note that, since the subtree G^k has exactly one node adjacent to the root, $\pi(i^*) = r_k$ implies $|N^+(G^k) \cap B^k(x)| = 1$, in which case (2.27) places no further restrictions on the weight vector. If $|N^+(G^k) \cap B^k(x)| > 1$, we use (2.27) to determine w_j for every $j \in N^+(G^k) \cap B^k(x) \setminus \{i^*\}$. For the players in $N^+(G^k) \setminus B^k(x)$, we determine the weights from (2.28) in a recursive manner.

In the rest of the proof we will use the following result. Let $y := \eta^\beta(\mathcal{G})$. Then, for $i \in N(G^k)$ and $1 \leq k \leq p$,

$$x_i > 0 \Leftrightarrow w_i > 0 \Leftrightarrow y_i > 0 \qquad \forall i \in N, \tag{2.31}$$

where the first equivalence follows from (2.30) and the construction of w described above. Since y is a result of Algorithm 2.5.1, where only players with positive weights have to pay anything, we have $y_i > 0 \Rightarrow w_i > 0$. Finally, $w_i > 0$ implies $x_i > 0$, and from $x \in C(c_\mathcal{G})$ and Proposition 2.3.1(iv), there must exist some edge $e \in E(\tilde{P}_i)$ such that $c(e) > 0$. Then, since y has been constructed using Algorithm 2.5.1, we have $w_i > 0 \Rightarrow y_i > 0$.

Claim 2: $\beta \in \mathcal{B}(\mathcal{G})$.
Clearly, $\mathcal{T} = \mathcal{T}(x)$ is a partition of G into pseudo subtrees. Also, $x \geq 0$, together with (2.31), imply $w \geq 0$. Let $c_i > 0$ for some $i \in N(G^k)$, $1 \leq k \leq p$. Since $x^{N(G^k)} \in C(c_{\mathcal{G}^k})$ by Proposition 2.3.4(ii), we must have

$\tilde{O}_i(x) = \sum_{j\in\tilde{F}(i)}(x_j - c_j) \geq 0$ by Proposition 2.3.1(iii). Since $x_j \geq 0$ and $c_j \geq 0$ for all $j \in \tilde{F}(i)$, there must exist some $\ell \in \tilde{F}(i)$ such that $x_\ell > 0$, and $w_\ell > 0$ then follows from (2.31).

Claim 3: $x = \eta^\beta(\mathcal{G})$.
We have

$$y(N(G^k)) = c_{\mathcal{G}^k}(N(G^k)) = x(N(G^k)), \tag{2.32}$$

for $1 \leq k \leq p$, where the first equality follows from Algorithm 2.5.1 and $\beta \in \mathcal{B}(\mathcal{G})$, and the second from $x \in C(c_\mathcal{G})$ and Proposition 2.3.4(ii). If $N(G^k) = \{i\}$ for some $i \in N$, then $x_i = y_i$ follows directly, so we will assume in the following that $|N(G^k)| > 1$. Suppose, contrary to our claim, that $x^{N(G^k)} \neq y^{N(G^k)}$. By (2.32), there must exist $i, j \in N(G^k)$ such that $x_i < y_i$ and $x_j > y_j$.

Consider node i. We will first show that

$$x_\ell \leq y_\ell \qquad\qquad \forall \ell \in \tilde{F}(i). \tag{2.33}$$

To prove (2.33), first note that $x_\ell = 0 \Leftrightarrow y_\ell = 0$ follows from (2.31). Next, note that w satisfies (2.27)-(2.29) with respect to x, by definition, and with respect to y, since (2.27)-(2.29) follows from Algorithm 2.5.1. Also, because of (2.31), we have $B^k(x) = B^k(y)$, and the definitions of $\pi^+(\bullet)$ and $N^+(G^k)$ are unambiguous. Then (2.28) implies that $f_m(x) < f_m(y)$ for every $m \in (\pi^+)^{-1}(i)$, so there must exist some $\ell \in \tilde{F}(i) \setminus \{i\}$ such that $x_\ell < y_\ell$. The argument can be repeated for $i = \ell$, and by continuing in this manner, we will eventually have shown that there is a leaf $\ell \in \tilde{F}(i)$ such that $x_\ell < y_\ell$. Now we will show that $x_\ell \leq y_\ell$ for every leaf $\ell \in \tilde{F}(i)$. Suppose, on the contrary, that this was not true, i.e., there exists a leaf $m \in \tilde{F}(i)$ such that $x_m > y_m$. Then it must be possible to find two branches $\tilde{B}_{pq}$ and $\tilde{B}_{ps}$, both rooted at $p \in \tilde{F}(i)$, such that $m \in N(\tilde{B}_{pq})$ and $x_t \geq y_t$ for all $t \in N(\tilde{B}_{pq})$, and such that $x_t \leq y_t$ for all $t \in N(\tilde{B}_{ps})$. In order to see that this yields a contradiction, note that (2.28) and $x_m > y_m$ implies

$$x_t > y_t \; \forall t \in N(\tilde{B}_{pq}) \Rightarrow f_q(x) > f_q(y) \Rightarrow x_p > y_p,$$

and that

$$x_t \leq y_t \; \forall t \in N(\tilde{B}_{pq}) \Rightarrow f_q(x) \leq f_q(y) \Rightarrow x_p \leq y_p.$$

Now, since every $\ell \in \tilde{F}(i)$ such that ℓ is a leaf of G^k satisfies $x_\ell \leq y_\ell$, we can use (2.28) in a recursive manner to prove (2.33). Then $x_i < y_i$ and (2.33) together imply $f_i(x) < f_i(y)$, which, by (2.28), implies $x_{\pi^+(i)} < y_{\pi^+(i)}$. Setting $i := \pi^+(i)$, we can successively repeat this argument until we have $i \in N^+(G^k) \cap B^k(x)$.

We have thus shown that there exists $i \in N^+(G^k) \cap B^k(x)$ such that $x_i < y_i$. Using the same line of argument for node j, we can show that there exists $j \in N^+(G^k) \cap B^k(x)$ such that $x_j > y_j$. If $|N^+(G^k) \cap B^k(x)| = 1$, this is in itself a contradiction, otherwise the contradiction follows from (2.27). □

In the same way that Proposition 2.4.5 enabled us to prove Theorem 2.4.6, Proposition 2.5.5 enables us to prove that the set of neighbour-home allocations equals the core.

Theorem 2.5.6 *For any maintenance problem $\mathcal{G}$ the set of all neighbour-home allocations equals $C(c_\mathcal{G})$, i.e. $\{\eta^\beta(\mathcal{G}) \mid \beta \in \mathcal{B}(\mathcal{G})\} = C(c_\mathcal{G})$.*

2.6 Conclusion

We have shown that every core point can be obtained from weighted versions of the painting stories of Maschler et al. (1995), and we have given explicit formulas for computing the corresponding weights. The alternative description of the core provided by our results may be useful when choosing between different core points.

The weighted down-home allocation rule corresponds to the weighted Shapley value. We also know, from Maschler et al. (1995), that $\eta^{(\mathcal{T},w)}(\mathcal{G})$ is equal to the nucleolus of $c_\mathcal{G}$ if we set $\mathcal{T} = \{G\}$ and $w_i = \frac{1}{|N|}$ for all $i \in N$. In Yanovskaya (1992) , the *weighted nucleolus* is defined by replacing the ordinary excess function by a weighted excess function, and it is shown that every point in the relative interior of the core can be obtained as a weighted nucleolus. For the game (N, g), and some pre-imputation x, this weighted excess function is given by, for any $S \neq N, \emptyset$, $\epsilon^p(S, x) := p_S(g(S) - x(S))$, where $p_S > 0$. Let $\beta := ((G^1, \ldots, G^p), w) \in \mathcal{B}(\mathcal{G})$ for some maintenance problem $\mathcal{G}$, and let $k = 1, \ldots, p$. Suppose we set $p_S := f(w^{N(G^k)})$ for all $S \subset N(G^k)$ such that $S \neq \emptyset$, where $f : \mathbf{R}^{N(G^k)} \to \mathbf{R}$. An interesting open problem is whether we can pick the function f such that $\eta^\beta(\mathcal{G})$, when restricted to the members of $N(G^k)$, is the weighted nucleolus of the game $c_{\mathcal{G}^k}$.

Chapter 3

Bankruptcy and Flow Sharing Problems

3.1 Introduction

The purpose of this chapter is to link two classical problems, namely, the *bankruptcy problem* and the *flow sharing problem.* The former problem concerns the allocation of an estate among several claimants, where the total amount of the claims exceeds the value of the estate, whereas the latter deals with the allocation of a network flow among the terminal (claimant) nodes of a capacitated network. The situations motivating both problems[1] can be characterized as *emergency situations*, arising in connection with, e.g., corporate bankruptcies, deaths, strikes or famines. In order to avoid damaging conflicts in such situations, allocation rules, satisfying the participants' notions of *fairness*, should be specified in advance.

In the literature on bankruptcy problems, many different allocation rules have been suggested, some of which will be presented in the next section, and the focus has been on *characterizing* these rules, such as in Aumann and Maschler (1985), Dagan (1996), Dagan et al. (1997), Dagan and Volij (1997), Herrero et al. (1999), Hokari and Thomson (2000), and Chun (1999). The literature on the flow sharing problem, on the other hand, has focused entirely on egalitarian solutions, and the challenge there has been how to

[1]See O'Neill (1982), Aumann and Maschler (1985), and Brown (1979).

compute the solutions. The papers on flow sharing differs slightly with respect to what is meant by an egalitarian solution. In Megiddo (1974) and Megiddo (1977), as well as in Fujishige (1980), an egalitarian solution is found by maximizing, in a lexicographically manner, the vector consisting of the (weighted) amounts received by the terminal nodes. Brown (1979) uses a maximin formulation, while Ichimori et al. (1982) study the problem of simultaneously maximizing the smallest (weighted) amount and minimizing the largest amount received.

In this chapter, we will exploit the similarities between the two problems, and show how the theory on one of them can be used in connection with the other. First, flow sharing problems contribute to the theory on bankruptcy problems in that a *generalized bankruptcy problem*, i.e., with more than one estate, can be modeled as a flow sharing problem where edge capacities represent the size of claims. Second, the theory on bankruptcy problems will be used to develop solutions to the flow sharing problems. We do this by forming, in a natural way, a reduced flow sharing problem for every pair of claimant nodes (Section 3.4). The reduced problems can be interpreted as bankruptcy problems, and by assuming that every pair of claimants believe in some specific notion of fairness, as represented by an allocation rule for their two-person bankruptcy problem, a unique solution to the n-person flow sharing problem results. In the same manner as has been done for bankruptcy problems[2], we will also formulate a flow sharing *game* in the form of a cooperative TU-game, and show that the unique solutions obtained for the flow sharing problem correspond to important game-theoretic solution concepts.

In Section 3.2 we give a formal introduction to bankruptcy problems involving a single estate, including some of the solutions that have been suggested. Then, in Section 3.3, we introduce flow sharing problems, and show how they are related to generalized bankruptcy situations. In Section 3.4 we formulate reduced two-person flow sharing problems, and develop two unique consistent solutions to the n-person problem by applying solution concepts from Section 3.2. Then, in Section 3.5, we introduce the *flow sharing game*, and show that important solution concepts for this game coincides with the so-

[2] Aumann and Maschler (1985).

lution concepts of Section 3.4. In Section 3.6 we present transfer schemes that converge to the solutions presented previously. Finally, in Section 3.7, we generalize one of the solutions from Section 3.4 by introducing weights for the claimant nodes.

3.2 Bankruptcy situations with one estate

In the bankruptcy problem, an estate with value E is to be divided among n individuals with claims given by the vector $d = (d_1, d_2, \ldots, d_n)$. An *allocation* for the bankruptcy problem (E, d) is given by a vector $x \in \mathbf{R}^n$ such that $0 \leq x_i \leq d_i$ for every $i \in N$, where $N := \{1, \ldots, n\}$. We shall say that an allocation x is *feasible* if, in addition, $\sum_{i \in N} x_i = E$.

Numerous solutions to this problem have been put forward. In practice the *proportional* rule, whereby the estate is divided among the claimants in proportion to their claims, is commonly used. The *constrained equal award* rule tries to give equal amounts to all claimants, subject to the constraint that no individual should get more than his claim. Mathematically, the amount allocated to claimant i is

$$CEA_i(E, d) := \min\{d_i, \lambda\},$$

where λ is chosen such that the allocation is feasible. The constrained equal award rule focuses on the amounts that the claimants are awarded. In some cases, e.g., if the estate is nearly large enough to cover all claims, the claimants may be more concerned with losses, i.e., the part of their claims that are not covered. In such at situation, it may be sensible to equalize losses instead of awards. The *constrained equal loss* rule does this, by allocating to claimant i the amount

$$CEL_i(E, d) := \max\{0, d_i - \lambda\},$$

where λ is chosen such that the allocation is feasible. A hybrid between the constrained equal award rule and the constrained equal loss rule is the *Talmud* rule[3], where awards are considered when the estate is smaller than

[3]Aumann and Maschler (1985) attributes this rule to Rabbi Nathan, the author of a particular passage of the Talmud, a 2000-year old document that forms the basis for Jewish civil, criminal, and religious law.

one-half the total size of the claims, and losses are considered otherwise. Mathematically, the Talmud rule is given by

$$T_i(E,d) := \begin{cases} \min\{d_i/2, \lambda\} & \text{if } E \leq \sum d_j/2, \\ \max\{d_i/2, d_i - \lambda\} & \text{otherwise,} \end{cases}$$

where λ is chosen such that feasibility is obtained.

Aumann and Maschler (1985) develop several interesting characterizations of the Talmud rule. One of them has to do with the *contested garment principle*, which is an allocation rule defined only for bankruptcy problems with two claimants. In order to explain this principle, let $(E, (d_1, d_2))$ be a bankruptcy problem with two claimants. According to the contested garment principle, claimant i should concede

$$(E - d_i)_+,$$

i.e., the part of the estate not claimed by himself, to j. If this applies to both claimants, then the disputed (remaining) amount will be

$$E - (E - d_1)_+ - (E - d_2)_+,$$

which should be divided equally among the two claimants. Hence, the amount to be received by i will be

$$x_i = \frac{E - (E - d_1)_+ - (E - d_2)_+}{2} + (E - d_j)_+.$$

In the n-claimant case, the contested garment principle cannot be directly applied. However, if x is a feasible allocation, we may apply the principle to the two-claimant problem $(x_i + x_j, (d_i, d_j))$ formed by picking two distinct individuals i and j. Aumann and Maschler (1985) show that every bankruptcy problem has a unique solution x such that (x_i, x_j) is the solution to every two-claimant problem $(x_i + x_j, (d_i, d_j))$. This solution, called the *contested garment consistent* solution, turns out to be exactly the Talmud rule.

In Aumann and Maschler (1985), a cooperative game with transferable utility is formulated, where

$$v_{E,d}(S) := (E - d(N \setminus S))_+$$

is the value of coalition $S \subseteq N$. This value represents a pessimistic view of how much the members of S could get, since the members of $N \setminus S$ are allowed to take as much as they want. Aumann and Maschler (1985) show that the nucleolus of the bankruptcy game $v_{E,d}$ is precisely the Talmud rule for the bankruptcy problem (E, d). The method of "recursive completion" in O'Neill (1982) is the Shapley value of this game, and Curiel et al. (1987) show that the τ-value of this game coincides with an adjusted version of the proportional rule.

3.3 Flow sharing problems and bankruptcy situations with more than one estate

The situation modeled by the bankruptcy problem of the previous section is similar to those motivating the *flow sharing problem* of Brown (1979), who studies the problem of allocating the (maximal) flow of a capacitated network among its sink nodes. One of the real-world examples mentioned by Brown concerns the transportation of coal from a set of coal mines to a set of power producers. During a coal strike, only non-union mines will be producing, and the limited production capacity of these mines must be shared by the users of coal, i.e., the power companies. Each power company "claims" an amount of coal equal to the amount used during normal (no strike) times. The solution studied by Brown (1979) is one where the smallest (weighted) amount of coal is maximized, subject to the capacities of the coal mines and the transportation network. Another example is the distribution of food during a famine, where the food supplies has to go through a transportation network with limited capacities.

Both in the bankruptcy situation, and the situations underlying the flow sharing problem, a limited amount of some resource has to be allocated among a set of agents in an emergency situation, according to some prespecified rule. In the following, we will use a network approach similar to that of Brown (1979) in order to model bankruptcy situations with more than one estate. In the rest of this section, we will describe the flow sharing problem formally. In order to stress the similarities with bankruptcy problems, we shall show how the flow sharing model can be used to describe a bankruptcy situation with more than one estate. However, the model can of course be used in connection with other situations where the limited availability of a resource can be represented by a capacitated network.[4]

In a bankruptcy situation with more than one estate, there may be several types of *interdependencies* among estates:

(i) A creditor may have claims on several estates.

(ii) The estates may have claims on each other.

In order to model such interdependencies, we will use a network model. Let the set of estates be given by $M := \{1, \dots, m\}$, and the set of claimants by $N := \{1, \dots, n\}$. The sets M and N will correspond to *intermediate (estate) nodes* and *sink (claimant) nodes*, respectively, in our network model.[5] The entire set of nodes $V := \{s\} \cup M \cup N$ also contains the artificial *source node* s. The nodes are connected by a set of edges $A \subseteq V \times V$. A directed edge (i,j) is an ordered pair, and it has a finite capacity k_{ij}. In the network model corresponding to a bankruptcy situation, an edge (i,j) may belong to one of the following types:

(i) $i = s$ and $j \in M$, in which case (i,j) represents those assets of estate j that are not claims on estates in $M \setminus \{j\}$. In order to make things simple, we shall refer to such assets as *cash*, although in practice they can take many forms. The value of these assets will be determined exogenously, and is given by the edge capacity k_{ij}.

(ii) $i \in M$ and $j \in N$, in which case k_{ij} is the value of j's claim on estate i.

(iii) $i \in M$ and $j \in M$, in which case k_{ij} is the value of estate j's claim on estate i.

[4]The assumption of *transferable utility*, i.e., that a unit of the disputed resource has the same value to every agent, limits the applicability of the model, especially if the disputed resource is a physical one, such as in the examples mentioned by Brown (1979).

[5]In order to distinguish between estate nodes and claimant nodes in figures, we shall denote the node of estate j by e_j.

Example 3.3.1 [Figure 3.1] In order to illustrate the above concepts, we present the example with 2 estates and 3 claimants shown below. Estate 1 contains cash worth 40, while the cash amount in estate 2 is 60. In addition to cash, estate 2 consists of a claim of 60 on estate 1. Consider a solution

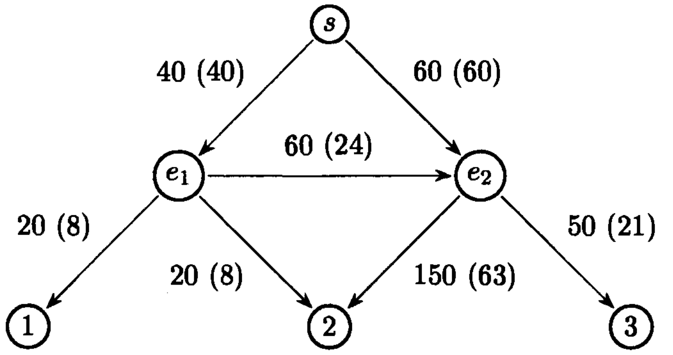


Figure 3.1: Example 3.3.1

based on the proportional rule. Estate 1 would then be split according to the vector

$$40 \cdot (1/5, 1/5, 3/5) = (8, 8, 24),$$

where the claim of estate 2 is awarded 24. The value to be shared by 2 and 3 in estate 2 would then be 84 (= 60 + 24), giving the allocation

$$84 \cdot (3/4, 1/4) = (63, 21)$$

The total amounts awarded to the three claimants is

$$(8, 71, 21)$$

In Figure 3.1(a), the divisions of the two estates are illustrated as flows (numbers in parentheses) over the edges. ◁

From Example 3.3.1 we see that an allocation of the bankruptcy situation can be interpreted as a network flow. Mathematically, a flow is a function $f : A \to \mathbf{R}$, where f_{ij} is the flow from node i to node j. In order to deal with network flows, we need to introduce a few concepts concerned with networks. If P and Q are subsets of the set V of network nodes, let

$$(P, Q) := \{(p, q) \in A : p \in P \text{ and } q \in Q\},$$

i.e., the set of edges going from P to Q. The case $(P, V \setminus P)$ is of special interest, and is called a *cut*. For a network flow f, let the flow from P to Q be denoted by

$$f(P,Q) := \sum_{(p,q)\in(P,Q)} f_{pq},$$

and the capacity $k(P,Q)$ can be defined in an analogous manner.

The net flow into a node $i \in V$ is given by

$$\text{net}(f,i) := f(V,\{i\}) - f(\{i\},V).$$

In order for a flow f to be feasible, it should satisfy

$$\text{net}(f,i) \begin{cases} \leq 0 & \text{if } i = s, \\ \geq 0 & \text{if } i \in N, \\ = 0 & \text{if } i \in M. \end{cases} \tag{3.1}$$

I.e., the direction of the net flow should be out of the sink node and into the source (claimant) nodes. For the intermediate (estate) nodes, outflow should balance inflow exactly. Moreover, the flow should be nonnegative (i.e., the flow should not go backwards), and should not violate the edge capacities, i.e.,

$$0 \leq f_{ij} \leq k_{ij} \qquad\qquad (i,j) \in A. \tag{3.2}$$

In a model of a bankruptcy situation, where edge capacities represent the size of claims, (3.2) implies that no claimant should get more than the size of his claim. Let the set of feasible flows, i.e., those satisfying (3.1) and (3.2) be denoted by F. Also, let

$$X := \{x \in \mathbf{R}^n : x_i = \text{net}(f,i)\ \forall i \in N,\ f \in F\}$$

denote the set allocations corresponding to feasible flows. For any set $S \subseteq N$ of sink nodes, let the total net inflow of S be given by

$$\text{net}(f,S) := \sum_{i\in S} \text{net}(f,i) = f(V,S) - f(S,V).$$

Then the set of *maximal flows* can be defined as

$$F^* := \{f \in F : \text{net}(f,N) \geq \text{net}(f',N)\ \forall f' \in F\},$$

i.e., the set of flows for which the total net flow into the sink nodes in N is maximal. The corresponding set of allocations is

$$X^* := \{x \in \mathbf{R}^n : x_i = \mathrm{net}(f, i)\ \forall i \in N,\ f \in F^*\},$$

and a member of X^* will be referred to as a *feasible allocation*, in line with the terminology for single-estate bankruptcy problems. The total amount awarded is maximal, given the restrictions caused by the limited amount of cash available in the estates, and the additional restriction that no claimant should be awarded more than the size of his claim. In Example 3.3.1, the allocation $(8, 71, 21)$ is a feasible allocation.

3.4 Bilateral comparisons and consistent solutions

The set of feasible allocations in a flow sharing problem can be quite large. E.g., in Example 3.3.1, the allocations

$$(20, 40, 40) \qquad \text{and} \qquad (10, 65, 25)$$

are both feasible allocations, since they correspond to maximal flows in the network of Figure 3.1. The former allocation seems to be an attempt to equalize the amounts received by the respective claimants, given that the allocation should be feasible. It may be argued that this allocation is unfair to claimant 2, who gets no more than claimant 3, although his claims are much higher. The latter allocation awards 2 a larger portion of his total claims.

In this section we shall impose additional restrictions on the candidate allocations by requiring that they should be considered *fair* by the claimants. Fairness involves comparisons between distinct individuals, and we will limit ourselves to pairwise comparisons. We will make assumptions with regard to what a fair allocation rule in two-person problems is, and develop a set of equations that the solution to the n-person problem must satisfy in order to be *consistent* with the solution of every two-person problem. Of course, the resulting equation system may not be solvable, or have many solutions, depending on the allocation rule that is used for the two-person problems.

Consider some allocation $x \in X^*$. Pick two claimants i and j, and assume that the other claimants are awarded according to x, i.e., an amount of $x_i + x_j$ is to be shared by i and j. Will the allocation (x_i, x_j) be considered fair by i and j? In order to study this question, we form a *reduced flow sharing problem* for i and j by defining, for any subset $S \subset N$,

$$F(S,x) := \{f \in F : \mathrm{net}(f,i) = x_i \ \forall i \in S\}.$$

Then $F(N \setminus \{i,j\}, x)$ is the set of flows available to i and j, given that the claimants in $N \setminus \{i,j\}$ are rewarded according to x. The maximal flow that i can receive in the reduced problem will be

$$d_{ij}(x) := \max_{f \in F(N\setminus\{i,j\},x)} \mathrm{net}(f,i),$$

which may be interpreted as i's claim in a two-person bankruptcy given by

$$(x_i + x_j, d(x)),$$

where $d(x) := (d_{ij}(x), d_{ji}(x))$. Note that

$$x \in X^* \Rightarrow \max\{d_{ij}(x), d_{ji}(x)\} \leq x_i + x_j \leq d_{ij}(x) + d_{ji}(x), \tag{3.3}$$

where the first inequality means that neither i nor j claims more than the value of the entire estate, and the second inequality says that they together claim at least the value of the entire estate. As an illustration, consider Example 3.3.1 when $x = (10, 65, 25)$ is proposed for the entire problem. Given that claimant 1 is awarded 10, claimant 2 can secure himself $d_{23}(x) = 90$, and 3 can get $d_{32}(x) = 50$. Other claims are $d_{13}(x) = d_{12}(x) = 20$, $d_{21}(x) = 75$, and $d_{31}(x) = 35$.

Lemma 3.4.1 *Let $x \in X^*$, and suppose y satisfies $y_k = x_k$ for all $k \in N \setminus \{i,j\}$, and that (y_i, y_j) is a feasible allocation for the two-person bankruptcy problem $(x_i + x_j, d(x))$. Then $y \in X^*$.*

Proof. Consider the allocation x^i, where claimant i gets his entire claim $d_{ij}(x)$, i.e.,

$$x^i_k = \begin{cases} d_{ij}(x) & k = i \\ x_j - [d_{ij}(x) - x_i] = x_i + x_j - d_{ij}(x) & k = j \\ x_k & \text{otherwise.} \end{cases}$$

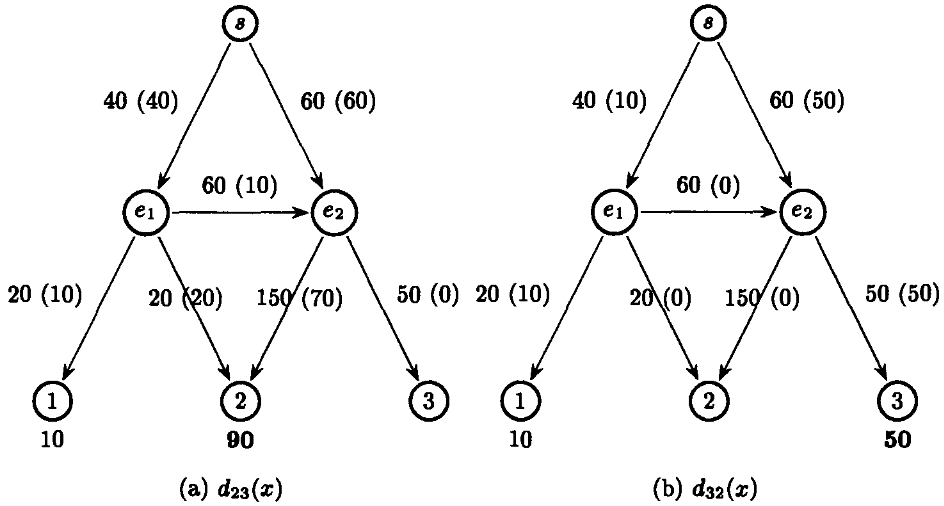


Figure 3.2: The reduced flow sharing problem $(x_2 + x_3, d(x))$ in Example 3.3.1, when $x = (10, 65, 25)$

The allocation x^i is obtained from x by increasing the net flow to i as much as possible, and reducing the flow to j accordingly. Let x^j be defined in a similar manner. Clearly, $x^i \in X^*$ and $x^j \in X^*$, and, since X^* is a convex set, every convex combination of x^i and x^j must also belong to X^*. Since y is such a convex combination, we must have $y \in X^*$. □

Suppose the *contested garment principle* from Section 3.2 is accepted as fair by both i and j. According to this principle, i concedes the part of the estate not claimed by him, namely $x_i + x_j - d_{ij}(x)$, to j. Claimant j, on the other hand, concedes $x_i + x_j - d_{ji}(x)$ to i. The disputed amount is thus

$$x_i + x_j - [x_i + x_j - d_{ij}(x)] - [x_i + x_j - d_{ji}(x)],$$

and this amount should be shared equally. Thus, i should receive the amount

$$\begin{aligned} & x_i + x_j - d_{ji}(x) + \frac{x_i + x_j - [x_i + x_j - d_{ij}(x)] - [x_i + x_j - d_{ji}(x)]}{2} \\ = & \frac{x_i + x_j - d_{ij}(x) - d_{ji}(x)}{2}. \end{aligned}$$

Requiring that the contested garment solution should be consistent with x

for claimant i means that

$$x_i = \frac{x_i + x_j - d_{ij}(x) - d_{ji}(x)}{2} \Leftrightarrow x_i - d_{ij}(x) = x_j - d_{ji}(x).$$

If x is consistent with the contested garment principle for every pair of individuals, we shall say that x is *contested garment consistent*[6] (CG-consistent). Formally, CG-consistency means that x must be a solution to the equation system

$$x_i - d_{ij}(x) = x_j - d_{ji}(x) \qquad \{i,j\} \subset N. \tag{3.4}$$

In Example 3.3.1, a solution of (3.4) is $x = (10, 65, 25)$.

Theorem 3.4.2 *Every flow sharing problem has a unique CG-consistent solution.*

Proof. In Theorem 3.5.4 we will show that (3.4) is exactly the equation system describing the nucleolus of a certain cooperative TU-game. Since we know from e.g. Maschler et al. (1979) that the nucleolus exists and consists of a single point, the result follows. □

Another view of fairness, frequently advocated in discussions on e.g. income distribution, is that every individual should get the same amount. By adding a feasibility requirement, we get the *constrained equal award solution* from Section 3.2. Consider two individuals i and j, together receiving $x_i + x_j$. The constrained equal award solution is shown in Figure 3.3.[7]

A requirement for x to be considered fair for the pair consisting of i and j is thus

$$x_i = \min\left\{\max\left\{x_i + x_j - d_{ji}(x), \frac{x_i + x_j}{2}\right\}, d_{ij}(x)\right\}.$$

If a solution x is consistent with the constrained equal award rule for every pair of claimants, we shall say that x is *CEA-consistent.* Then x needs to satisfy

$$\min\left\{\max\left\{x_j - d_{ji}(x), \frac{x_j - x_i}{2}\right\}, d_{ij}(x) - x_i\right\} = 0 \quad \{i,j\} \subset N. \tag{3.5}$$

[6]This term is borrowed from Aumann and Maschler (1985), and can be seen as a generalization of the consistency requirement that is studied in their paper.

[7]Note that the case $\frac{x_i + x_j}{2} > \max\{d_{ij}(x), d_{ji}(x)\}$ is not of interest, because of (3.3).

Case	$CEA_i(x_i + x_j, d(x))$	$CEA_j(x_i + x_j, d(x))$
$\frac{x_i+x_j}{2} \leq \min\{d_{ij}(x), d_{ji}(x)\}$	$\frac{x_i+x_j}{2}$	$\frac{x_i+x_j}{2}$
$d_{ji}(x) \leq \frac{x_i+x_j}{2} \leq d_{ij}(x)$	$x_i + x_j - d_{ji}(x)$	$d_{ji}(x)$
$d_{ij}(x) \leq \frac{x_i+x_j}{2} \leq d_{ji}(x)$	$d_{ij}(x)$	$x_i + x_j - d_{ij}(x)$

Figure 3.3: The constrained equal award solution for the bankruptcy problem $(x_i + x_j, d(x))$

In Example 3.3.1, a solution of (3.5) is $x = (20, 40, 40)$.

Theorem 3.4.3 *Every flow sharing problem has a unique CEA-consistent solution.*

Proof. In Section 3.5.5 we will show that (3.5) is exactly the equation system describing the constrained egalitarian solution of a certain cooperative TU-game. We know from Dutta and Ray (1989) that, for convex games, this solution exists and consists of a single point. Since Proposition 3.5.2 shows that the game in question is indeed convex, the result follows. □

3.5 Flow sharing games

Using the same pessimistic view of the situation as in Aumann and Maschler (1985), we let the value of a coalition in the game (N, v) be given by the amount that the coalition can secure for itself "without going to court"[8]. In order to define this value for an arbitrary coalition $S \subseteq N$, we first need to define the maximal amount that can be taken by $R = N \setminus S$. This value is given by[9]

$$m(R) := \begin{cases} 0 & \text{if } R = \emptyset, \\ \max_{f \in F} \operatorname{net}(f, R) & \text{otherwise.} \end{cases}$$

[8]Aumann and Maschler (1985), Section 6.

Since the total value of the bankruptcy (flow sharing) situation is $m(N)$, the worst-case scenario for S is given by the value

$$v(S) := m(N) - m(N \setminus S).$$

It is easily checked that for a single-estate bankruptcy problem (E, d), we have $v(S) = (E - d(N \setminus S))_+$. The values of m and v are shown for Example 3.3.1 in Figure 3.4.

S			$m(S)$	$v(S)$
1			20	0
	2		100	30
		3	50	0
1	2		100	50
1		3	70	0
	2	3	100	80
1	2	3	100	100

Figure 3.4: The game v for Example 3.3.1

Theorem 3.5.1 *The core of a flow sharing game*[10] *is exactly the set of*

[9]The function $m : 2^N \to \mathbf{R}$ is sometimes referred to, e.g., in Megiddo (1974), as the *characteristic function* of the network. The characteristic function m can also be related to *flow games* where the players control edges of the network, as studied by Curiel et al. (1989). In their model, the control over an edge $e \in A$ is modeled by a *control game* w_e, where $w_e(S) = 1$ if the coalition S controls e, and $w_e(S) = 0$ otherwise. By extending our network, and introducing control games for the edges, we can relate our network situation to that studied in their paper. Let $V' := V \cup \{t\}$ and $A' := A \cup \{(i,t) : i \in N\}$ be the node and edge set that we get by adding the sink node t. Let the capacity of $(i,j) \in A$ be given by

$$k'_{ij} := \begin{cases} k_{ij} & i,j \in V, \\ \infty & i \in N, j = t, \end{cases}$$

and the control game w_{ij} be given by

$$w_{ij}(S) := \begin{cases} 1 & i \in N \cup S, j = t, \\ 0 & i \in N \setminus S, j = t, \\ 1 & \text{otherwise.} \end{cases}$$

The value $m'(S)$ of the maximal flow from the sink s to the source t in the network (V', A', k'), when only the edges controlled by S are used, satisfies $m'(S) = m(S)$. In Curiel et al. (1989), this value is interpreted as a *profit*, as opposed to a *cost*, implying that e.g. the core $C(m)$ of our flow sharing game (see Theorem 3.5.1) is the *anti-core* of their flow game, where the anti-core is the set of allocations obtained by reversing the direction of the core inequalities. It is easily seen from e.g. (1.4) that the anti-core is empty if and only if the core is nonempty.

[10]A similar result is shown for single-estate bankruptcy games in Theorem 2 of Curiel et al. (1987).

feasible allocations for the flow sharing problem, i.e.,

$$C(v) = C(m) = X^*.$$

Proof. The first equality follows from $v(N) = m(N)$, and, for $x \in \mathbf{R}^n$ such that $x(N) = v(N)$, and $P \subset N$,

$$\begin{aligned} &x(P) \geq v(P) \\ \Leftrightarrow &x(P) \geq m(N) - m(N \setminus P) \\ \Leftrightarrow &x(N \setminus P) \leq m(N \setminus P). \end{aligned}$$

In order to show the second equality[11], we assume that $x \in X^*$, and will prove that $x \in C(m)$. First note that $x(N) = m(N)$ follows from the definition of X^*. It then remains to show that $x(S) \leq m(S)$ for all $S \subset N$. Since $x \in X^*$, we know that there exists $f' \in F^*$ such that $x_i = \text{net}(f', i)$ for all $i \in N$. Then we must have

$$x(S) = \sum_{i \in S} \text{net}(f', i) \leq \max_{f \in F} \sum_{i \in S} \text{net}(f, i) = m(S).$$

To complete the proof of the second equality, we assume that $x \in C(m)$, and will prove that $x \in X^*$. Since $x(N) = m(N)$, the proof can be completed by constructing $\hat{f} \in F^*$ such that

$$x_i = \text{net}(\hat{f}, i) \qquad\qquad i \in N.$$

In order to do this, form a new network (V', A', k') by adding a new sink node t, i.e., $V' = V \cup \{t\}$. We also add an edge from every $i \in N$ to t, i.e., $A' = A \cup \{(i, t) : i \in N\}$. The new capacities are given by

$$k'_{ij} = \begin{cases} x_i & \text{if } i \in N \text{ and } j = t, \\ k_{ij} & \text{if } (i, j) \in A. \end{cases}$$

Let s and t be the single source and sink node, respectively, of the new network.

[11]The remaining part of the proof is, with the exception of differences in notation, almost identical to that of Lemma 4.1 of Megiddo (1974). For the sake of completeness we repeat the proof here.

We will now show that $(V, \{t\})$ is a minimal cut separating s and t. In order to do this, let $P \subset V'$ be such that $s \in P$ and $t \in V' \setminus P$. Then

$$\begin{aligned} k'(P, V' \setminus P) &= k'(P, V \setminus P) + k'(P \cap N, \{t\}) \\ &= k(P, V \setminus P) + x(P \cap N) \\ &\geq m(N \setminus P) + x(P \cap N) \\ &\geq x(N) = k'(V, \{t\}). \end{aligned}$$

In order to construct the flow $\hat{f}$ for the network (V, A, k), let f' be a maximal flow for (V', A', k'). Then, since $(V, \{t\})$ is a minimal cut, we have $f'_{it} = x_i$ for all $i \in N$. Let $\hat{f}$ be the restriction of f' to A. Then, for any $i \in N$, we have

$$\begin{aligned} \text{net}(\hat{f}, i) &= f'(V, \{i\}) - f'(\{i\}, V) \\ &= \text{net}(f', i) + f'_{it} = f'_{it} = x_i \end{aligned}$$

□

Proposition 3.5.2 *A flow sharing game v is*

(i) monotonic:

$$S \subseteq T \Rightarrow v(S) \leq v(T)$$

(ii) convex[12]*:*

$$v(S \cup T) \geq v(S) + v(T) - v(S \cap T)$$

Proof. Both statements follow from Lemma 3.2 in Megiddo (1974), where it is shown that the characteristic function m is monotonic and *concave.* That v is monotonic if, and only if, m is monotonic follows immediately from the definition of v. That concavity of m is equivalent to convexity of v is shown in Lemma 3.5.3 below. □

[12] See Curiel et al. (1987) for a similar result for single-estate bankruptcy games.

Lemma 3.5.3 *A game v is convex if and only if its dual game*[13] *v^* is concave.*

Proof. Concavity of v^* means that, for any $S, T \subseteq N$,

$$v^*(S \cup T) \leq v^*(S) + v^*(T) - v^*(S \cap T) \tag{3.6}$$

Letting $P := N \setminus S$ and $Q := N \setminus T$, we get

$$\begin{aligned} v^*(S \cup T) &= v(N) - v(N \setminus S \setminus T) \\ &= v(N) - v(P \cap Q), \end{aligned}$$

and

$$\begin{aligned} &v^*(S) + v^*(T) - v^*(S \cap T) \\ =&v(N) - v(N \setminus S) + v(N) - v(N \setminus T) - v(N) + v(N \setminus (S \cap T)) \\ =&v(N) - v(P) - v(Q) + v(P \cup Q) \end{aligned}$$

Hence (3.6) is equivalent to

$$v(P \cup Q) \geq v(P) + v(Q) - v(P \cap Q),$$

which means that v is convex. □

In Section 3.4, by specifying what is meant by fairness in the two-person setting, we were able to develop a unique solution for the n-person flow sharing problem. We characterized two such solutions, the CG-consistent solution given by (3.4), and the CEA-consistent solution given by (3.5). Note that these two solutions for the flow sharing *problem* were characterized without any reference to game-theoretic concepts. However, as we show below, they coincide with two well-known solutions for the flow sharing *game*. We first state the results, and then go on to prove them.

The *nucleolus*[14] is defined via the excess values

$$e(v, S, x) = v(S) - x(S) \qquad\qquad S \subset N,$$

[13]For any $S \subseteq N$, the value of the dual game is $v^*(S) = v(N) - v(N \setminus S)$. Since $v^*(N) = v(N)$, the relationship between the two games will be symmetric, i.e., the dual of v^* is v.

[14]See Section 1.3.

and seeks to equalize these values by minimizing, in a lexicographical manner, the vector of excess values.

Theorem 3.5.4 *The CG-consistent solution for a flow sharing problem coincides with the nucleolus of the corresponding flow sharing game.*

The set of *constrained egalitarian solutions*, introduced by Dutta and Ray (1989), assumes that equity is a desirable social goal, but recognizes that private preferences may induce selfish behavior. In order to define this solution concept, we need the *Lorenz-ordering*. Consider some set $B \subset \mathbf{R}^k$ such that $\sum_{i=1}^{k} = b$ for any $x \in B$, where b is a constant. For any vector $x \in B$, let σ_x represent a non-decreasing order of x, i.e.,

$$i < j \Rightarrow x_{\sigma_x(i)} \leq x_{\sigma_x(j)}.$$

Then, for $x, y \in B$, we say that x Lorenz-dominates y, written $x >_{LD} y$ if $\sum_{i=1}^{j} x_{\sigma_x(i)} \geq \sum_{i=1}^{j} y_{\sigma_y(i)}$ for all $j = 1, \dots, k$, with strict inequality for at least one j. The *Lorenz map* E picks the undominated elements of any set, i.e.,

$$EB := \{x \in B : \text{ there is no } y \in B \text{ such that } y >_{LD} x\}. \tag{3.7}$$

The set of constrained egalitarian solutions for a game v is defined by constructing, in a recursive way, the *Lorenz core* $L(S)$ for any coalition $S \subseteq N$. For a singleton coalition, this is the set $L(i) := \{v(i)\}$. Consider some coalition $S \subseteq N$, and suppose that the Lorenz cores for any coalition T such that $|T| < |S|$ has been defined. Then

$$L(S) := \{x \in \mathbf{R}^{|S|} : x(S) = v(S), \nexists T \subset S \text{ and } y \in EL(T) \text{ s.t. } y > x_T\}, \tag{3.8}$$

where $y > x_T$ means $y \geq x_T$ and $y \neq x_T$. The set of constrained egalitarian solutions is then $EL(N)$. In general we may have $EL(N) = \emptyset$. However, as shown by Dutta (1990), for convex games the set is nonempty and consists of a unique point. This point is the unique core point that Lorenz-dominates every other core point, denoted $CES(v)$. As flow sharing games are convex, we can use this property.

Theorem 3.5.5 *The CEA-consistent solution for a flow sharing problem coincides with the constrained egalitarian solution of the corresponding flow* *sharing game.*

In order to prove Theorems 3.5.4 and 3.5.5, we will use the fact that both these solution concepts satisfy the *converse reduced game property.*[15] For a game (N, v), the allocation x and the player subsets $R \subseteq P \subseteq N$, the value of the reduced game $(P, v^{P,x})$ á la Davis and Maschler (1965) is given by

$$v^{P,x}(R) := \begin{cases} x(P) & \text{if } R = P, \\ 0 & \text{if } R = \emptyset, \\ \max_{Q \subseteq N \setminus P}\{v(R \cup Q) - x(Q)\} & \text{otherwise.} \end{cases} \tag{3.9}$$

A solution concept σ satisfies the converse reduced game property if

$$x_P \in \sigma(P, v^{P,x}) \; \forall P \subset N s.t. \; |P| = 2 \Rightarrow x \in \sigma(N, v),$$

where x_P is the restriction of x to P. In order to prove Theorems 3.5.4 and 3.5.5, we will first show that (3.9) takes a particularly simple form when v is a flow sharing game.

Lemma 3.5.6 *If v is a flow sharing game, and x is a core allocation, then, for $R \subseteq P \subseteq N$,*

$$v^{P,x}(R) = x(P) - \max_{f \in F(N \setminus P, x)} net(f, P \setminus R).$$

Proof.

$$\begin{aligned} v^{P,x}(R) &= \max_{Q \subseteq N \setminus P}\{v(R \cup Q) - x(Q)\} \\ &= \max_{Q \subseteq N \setminus P}\{m(N) - m(N \setminus R \setminus Q) - x(Q)\} \\ &= \max_{Q \subseteq N \setminus P}\{x(N \setminus Q) - m(N \setminus R \setminus Q)\} \\ &= x(P) - \min_{Q \subseteq N \setminus P}\{m(N \setminus R \setminus Q) - x(N \setminus P \setminus Q)\}, \end{aligned}$$

[15]Introduced by Peleg (1986). See Driessen (1991) for an overview of consistency (reduced game) properties in cooperative game theory.

where the second last equality follows from $x \in C(v) = C(m) \Rightarrow x(N) = m(N)$. Letting $Q' := N \setminus P \setminus Q$, we can show that

$$\begin{aligned} v^{P,x}(R) &= x(P) - \min_{Q' \subseteq N \setminus P} \{m(P \setminus R \cup Q') - x(Q')\} \\ &= x(P) - \min_{Q' \subseteq N \setminus P} \max_{f \in F} \{\mathrm{net}(f, P \setminus R) + \mathrm{net}(f, Q') - x(Q')\} \\ &= x(P) - \max_{f \in F(N \setminus P, x)} \mathrm{net}(f, P \setminus R), \end{aligned}$$

where the last equality follows from Lemma 3.5.7 below. □

Lemma 3.5.7 *Let $x \in X^*$, and $S \subseteq P \subseteq N$. Then*

$$\min_{Q \subseteq N \setminus P} \max_{f \in F} \{net(f, S) + net(f, Q) - x(Q)\} = \max_{f \in F(N \setminus P, x)} net(f, S). \quad (3.10)$$

Proof. In order to prove (3.10), it is sufficient to show that there exists an optimal solution $(\hat{Q}, \hat{f})$ to the left-hand side problem such that

$$\begin{array}{lll} (i) & \mathrm{net}(\hat{f}, i) = x_i & i \in \hat{Q}, \\ (ii) & \hat{Q} = N \setminus P. & \end{array}$$

In order to prove (i), let $(\hat{Q}, \hat{f})$ be optimal. We will first show that this solution must satisfy

$$\mathrm{net}(\hat{f}, i) \leq x_i \qquad i \in \hat{Q}. \quad (3.11)$$

Suppose this is not the case, i.e., there exists some $i \in \hat{Q}$ and

$$f' \in \arg\max_{f \in F} \{\mathrm{net}(f, S) + \mathrm{net}(f, \hat{Q} \setminus \{i\}) : \mathrm{net}(f, S \cup \hat{Q}) = \mathrm{net}(\hat{f}, S \cup \hat{Q})\} \quad (3.12)$$

such that

$$\mathrm{net}(f', i) > x_i. \quad (3.13)$$

Note that, since $\hat{f}$ is an optimal solution to the left-hand side of (3.10), the maximal net flow into $S \cup \hat{Q} \setminus \{i\}$ cannot be larger than $\mathrm{net}(\hat{f}, S \cup \hat{Q})$, and we must therefore have

$$\mathrm{net}(f', S) + \mathrm{net}(f', \hat{Q} \setminus \{i\}) = \max_{f \in F} \{\mathrm{net}(f, S) + \mathrm{net}(f, \hat{Q} \setminus \{i\})\}.$$

Then, because of (3.13), we must have

$$\max_{f\in F}\{\mathrm{net}(f,S)+\mathrm{net}(f,\hat{Q}\setminus\{i\})-x(\hat{Q}\setminus\{i\})\}<\max_{f\in F}\{\mathrm{net}(f,S)+\mathrm{net}(f,\hat{Q})-x(\hat{Q})\},$$

contradicting the optimality of $\hat{Q}$. Now, since $x \in X^*$, i.e., x corresponds to a feasible flow, we can choose $\hat{f}$ such that the inequalities in (3.11) are all satisfied as equalities.

In order to show (ii), we use (i) to rewrite the left-hand side of (3.10) as

$$\min_{Q\subseteq N\setminus P}\ \max_{f\in F(Q,x)} \mathrm{net}(f,S).$$

The value of the inner maximization problem will always decrease when we add more players to Q, since $F(Q\cup\{i\},x) \subseteq F(Q,x)$ for $i \in N\setminus P\setminus Q$. Hence $\hat{Q} = N\setminus P$ will be an optimal solution. □

Now we are ready to prove Theorems 3.5.4 and 3.5.5. The proof will be done by describing the nucleolus and the constrained egalitarian solution for reduced games with two players, and showing that this description leads exactly to the descriptions of the CG-consistent solution, i.e., (3.4), and the CEA-consistent solution, i.e., (3.5). Let $(\{i,j\},u)$ be a game with two players i and j. It is easily shown that the nucleolus for this game is the *standard solution.*[16] A solution (x_i,x_j) is the standard solution iff

$$x_i = u(i) + \frac{u(i,j)-u(i)-u(j)}{2}. \tag{3.14}$$

Under constrained egalitarianism, the allocation $\frac{u(i,j)}{2}$ is seen as the ideal solution, but one also has to make sure that i does not get less than $u(i)$, and that j does not get less than $u(j)$. This means that a solution (x_i,x_j) will be the constrained egalitarian solution iff

$$\begin{aligned} x_i &= \begin{cases} \max\left\{u(i),\frac{u(i,j)}{2}\right\} & \text{if } u(j)\le \frac{u(i,j)}{2},\\ u(i,j)-u(j) & \text{otherwise} \end{cases} \\ &= \min\left[\max\left\{u(i),\frac{u(i,j)}{2}\right\},u(i,j)-u(j)\right]. \end{aligned} \tag{3.15}$$

[16]For two-player games, many other solutions, such as the Shapley value and the τ-value, also coincide with the standard solution.

Proof. [Theorem 3.5.4] Peleg (1986) shows that the nucleolus satisfies the converse reduced game property. For a generalized bankruptcy problem, let $x \in C(v) = X^*$. For the players i and j, the characteristic function of their reduced game is, from Lemma 3.5.6,

$$v^{\{i,j\},x}(R) = \begin{cases} x_i + x_j & R = \{i,j\}, \\ x_i + x_j - d_{ji}(x) & R = \{i\}, \\ x_i + x_j - d_{ij}(x) & R = \{j\}. \end{cases} \tag{3.16}$$

Requiring that (x_i, x_j) should be the standard solution of the reduced game gives the equation

$$\begin{aligned} x_i &= x_i + x_j - d_{ji}(x) + \frac{x_i + x_j - [x_i + x_j - d_{ji}(x)] - [x_i + x_j - d_{ij}(x)]}{2} \\ &= \frac{x_i + x_j - d_{ji}(x) + d_{ij}(x)}{2} \\ &\Leftrightarrow 2x_i = x_i + x_j - d_{ji}(x) + d_{ij}(x) \\ &\Leftrightarrow x_i - d_{ij}(x) = x_j - d_{ji}(x), \end{aligned}$$

which, if imposed for every pair of players, is equivalent to (3.4). □

Proof. [Theorem 3.5.5] Dutta (1990) shows that the constrained egalitarian solution satisfies the converse reduced game property for convex games. Since the generalized bankruptcy game is convex (Proposition 3.5.2), we can use this property. We insert the values of $v^{\{i,j\},x}$ in (3.15), and get

$$\begin{aligned} x_i &= \min\left[\max\left\{x_i + x_j - d_{ji}(x), \frac{x_i + x_j}{2}\right\}, d_{ij}(x)\right] \\ \Leftrightarrow 0 &= \min\left[\max\left\{x_j - d_{ji}(x), \frac{x_j - x_i}{2}\right\}, d_{ij}(x) - x_i\right], \end{aligned}$$

which, if imposed for every pair of players, is equivalent to (3.5). □

3.6 Convergent transfer schemes

The equation systems (3.4) and (3.5) not only describes ideal[17] solutions to the allocation problem. They also immediately suggest methods to compute the solutions, through bilateral transfers.

[17] Given the respective views of fairness involved.

where the first inequality follows from

$$\begin{aligned}
&x \in X^* \\
\Rightarrow & x_j - d_{ji}(x) \leq 0 \leq d_{ij}(x) - x_i \\
\Leftrightarrow & x_j - d_{ji}(x) - x_i + d_{ij}(x) \leq 2(d_{ij}(x) - x_i) \\
\Leftrightarrow & \frac{x_j - d_{ji}(x) - x_i + d_{ij}(x)}{2} \leq d_{ij}(x) - x_i,
\end{aligned}$$

and the second inequality follows from

$$\begin{aligned}
& x_j - v(j) = x_j - m(N) + m(N \setminus \{j\}) = m(N \setminus \{j\}) - x(N \setminus \{j\}) \\
\geq & \underbrace{d_{ij}(x) + x(N \setminus \{i,j\})}_{\leq m(N \setminus \{j\}), \text{since } x \in C(m) = X^*} - x(N \setminus \{j\}) = d_{ij}(x) - x_i.
\end{aligned}$$

□

Example 3.6.2 [Figure 3.5] In order to illustrate the convergence of the sequences, we shall consider the example shown Figure 3.5(a). Given the allocation vector in step t, denoted x^t, we compute $\alpha^*(x^t) := \max_{i \neq j} \alpha_{ij}(x^t)$, where the maximum occurs for the nodes i^* and j^*. Likewise, we compute $\beta^*(x^t) := \max_{i \neq j} \beta_{ij}(x^t)$. The CG-consistent solution is

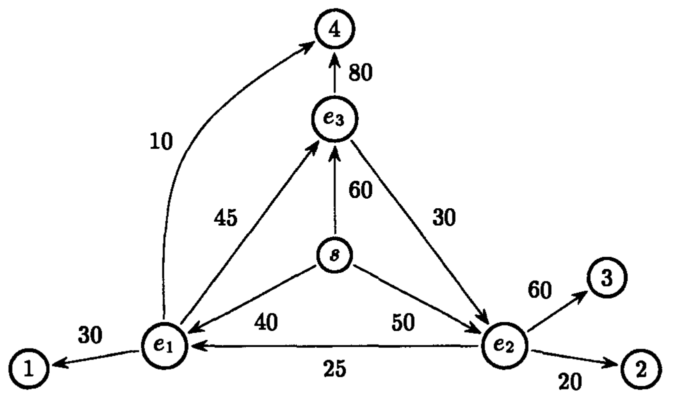
$$(16\ 2/3, 10, 46\ 2/3, 76\ 2/3),$$

and the CEA-consistent solution is

$$(30, 20, 50, 50).$$

The α-sequence shown in Figure 3.5(b) does seem to converge to the CG-consistent solution, as predicted by Theorem 3.6.1, although not in a finite number of steps. The β-sequence shown in Figure 3.5(c), on the other hand, finds the CEA-consistent solution in only four steps.[19] ◁

[19]It is easy to find examples for which also β-sequences need an infinite number of iterations in order to reach the limit point.



(a)

t	x_1^t	x_2^t	x_3^t	x_4^t	$\alpha^*(x^t)$	i^*	j^*
0	10.000	0.000	60.000	80.000	10.000	1	3
1	20.000	0.000	50.000	80.000	10.000	2	1
2	10.000	10.000	50.000	80.000	5.000	1	3
3	15.000	10.000	45.000	80.000	2.500	1	4
4	17.500	10.000	45.000	77.500	1.250	3	1
5	16.250	10.000	46.250	77.500	0.625	1	4
6	16.875	10.000	46.250	76.875	0.312	3	1
7	16.562	10.000	46.562	76.875	0.156	1	4
8	16.719	10.000	46.562	76.719	0.078	3	1
9	16.641	10.000	46.641	76.719	0.039	1	4
10	16.680	10.000	46.641	76.680	0.020	3	1
11	16.660	10.000	46.660	76.680	0.010	1	4
12	16.670	10.000	46.660	76.670	0.005	3	1
13	16.665	10.000	46.665	76.670	0.002	1	4
14	16.667	10.000	46.665	76.667	0.001	3	1
15	16.666	10.000	46.666	76.667	0.001	1	4

(b) α-sequence

t	x_1^t	x_2^t	x_3^t	x_4^t	$\beta^*(x^t)$	i^*	j^*
0	10.000	0.000	60.000	80.000	20.000	1	3
1	30.000	0.000	40.000	80.000	20.000	2	3
2	30.000	20.000	20.000	80.000	30.000	3	4
3	30.000	20.000	50.000	50.000	0.000	-	-

(c) β-sequence

Figure 3.5: Example 3.6.2

3.7 Weighted generalization of the CEA-consistent solution

In the previous sections we assumed that claimants are identical, except with respect to the size of their claims. However, different claims may not be considered equally important. Consider, e.g., the case where one claim is put forward by a group of individuals, while another represent the claim of one individual. In this case, the equity of various allocations could be judged on a per-capita basis. In order to model such asymmetries, we introduce a weight vector $\omega \in \mathbf{R}^n_{++}$. In this new setting, what is a fair allocation? We shall follow the same line of thought as in Section 3.4, i.e., we shall first define a fair allocation principle for two-person reduced flow sharing problems, and show that this implies a unique allocation for the n-person problem.

Let $x \in X^*$, i.e., x is a feasible allocation, and consider again the two-person bankruptcy problem $(x_i + x_j, d(x))$ of Section 3.4. We assume that a fair division of $x_i + x_j$ between i and j is to give claimant i an amount proportionate to his relative weight, i.e., the weighted constrained equal award solution shown in Figure 3.6.

Case		CEA_i^ω	CEA_j^ω
$\frac{\omega_i(x_i+x_j)}{\omega_i+\omega_j} \leq d_{ij}(x)$	$\frac{\omega_j(x_i+x_j)}{\omega_i+\omega_j} \leq d_{ji}(x)$	$\frac{\omega_i(x_i+x_j)}{\omega_i+\omega_j}$	$\frac{\omega_j(x_i+x_j)}{\omega_i+\omega_j}$
$\frac{\omega_i(x_i+x_j)}{\omega_i+\omega_j} \leq d_{ij}(x)$	$\frac{\omega_j(x_i+x_j)}{\omega_i+\omega_j} \geq d_{ji}(x)$	$x_i + x_j - d_{ji}(x)$	$d_{ji}(x)$
$\frac{\omega_i(x_i+x_j)}{\omega_i+\omega_j} \geq d_{ij}(x)$	$\frac{\omega_j(x_i+x_j)}{\omega_i+\omega_j} \leq d_{ji}(x)$	$d_{ij}(x)$	$x_i + x_j - d_{ij}(x)$

Figure 3.6: The weighted constrained equal award solution for the reduced problem $(x_i + x_j, d(x))$

The generalization of (3.5) is easily shown to be

$$\beta_{ij}^\omega(x) := \min\left\{\max\left\{x_j - d_{ji}(x), \frac{\omega_i x_j - \omega_j x_i}{\omega_i + \omega_j}\right\}, d_{ij}(x) - x_i\right\} = 0 \qquad \{i,j\} \subset N. \qquad (3.17)$$

In order to relate the solution of (3.17) to the flow sharing game, we will introduce the *weighted constrained egalitarian solution*, a natural generalization of the constrained egalitarian solution of Section 3.5. For a generalization of the rather lengthy definition given in Section 3.5, we refer to Hokari (1998) and Koster (1999). For the class of convex games, Hokari

(1998) shows that this solution is the unique Lorenz-maximal core point, denoted $CES^{\omega}(v)$, where the Lorenz ordering introduced in Section 3.5 is generalized in order to take the weights of players into account as in Ebert (1999). For $x \in \mathbf{R}^n$ take an ordering σ_x^{ω} of the players such that

$$i < j \Rightarrow \frac{x_{\sigma_x^{\omega}(i)}}{\omega_{\sigma_x^{\omega}(i)}} \leq \frac{x_{\sigma_x^{\omega}(j)}}{\omega_{\sigma_x^{\omega}(j)}},$$

i.e., σ_x^{ω} sorts the weighted elements of x in a non-decreasing order. Then, the points defining the piece-wise linear Lorenz curve $L_x^{\omega} : [0,1] \to [0,1]$ are given by

$$L_x^{\omega}\left(\frac{\sum_{j=1}^{i} \omega_{\sigma_x(j)}}{\omega(N)}\right) := \frac{\sum_{j=1}^{i} x_{\sigma_x(j)}}{x(N)} \qquad i = 0, \dots, n,$$

i.e., $L_x^{\omega}(p)$, for $0 \leq p \leq 1$, is the fraction of the total amount $x(N)$ received by the fraction p of the richest individuals.

In Example 3.3.1, suppose the weight vector $\omega = (2,3,1)$ is used. Compare the allocations $x = (20,40,40)$ and $y = (20,60,20)$. The vectors $x^{\omega} = (10, 13\ 1/3, 40)$ and $y^{\omega} = (10,20,20)$ are obtained by dividing with the weights, e.g. $x_2^{\omega} = x_2/\omega_2 = 13\ 1/3$. Since the ranking of the elements in neither vector changes as a result of this transformation, we have $\sigma_x^{\omega} = \sigma_y^{\omega}$. The corresponding Lorenz curves are shown in Figure 3.7. Notice for example that $L_x^{\omega}(5/6) = 60/100 = 0.6$, i.e., the vector x allocates 60% of the amount to the two worst-off claimants, i.e., 1 and 2. On the other hand we have $L_y^{\omega}(5/6) = 80/100 = 0.8$, i.e., y allocates 80% of the total amount to 1 and 2. Since $L_y^{\omega}(p) \geq L_x^{\omega}(p)$ for all $0 \leq p \leq 1$, and with strict inequality for some p, y is considered a more egalitarian allocation than x.

In general, for $x, y \in \mathbf{R}^N$ such that $x(N) = y(N)$, and for some weight vector $\omega \in \mathbf{R}_{++}$, we will say that y *weakly ω-Lorenz dominates* x, denoted $y \geq_{LD}^{\omega} x$, if and only if $L_y^{\omega}(p) \geq L_y^{\omega}(p)$ for all $0 \leq p \leq 1$. If $x \geq_{LD}^{\omega} y$ and not $y \geq_{LD}^{\omega} x$, then we say that y *strongly ω-Lorenz dominates* x, written $x >_{LD}^{\omega} y$. Note that the Lorenz-ordering is not complete, i.e., it is not always possible to rank allocations. E.g., the allocation $z = (0,75,25)$, for which the corresponding curve L_z^{ω} is shown in Figure 3.7, cannot be compared to x, since we have neither $x \geq_{LD}^{\omega} z$ nor $z \geq_{LD}^{\omega} x$.

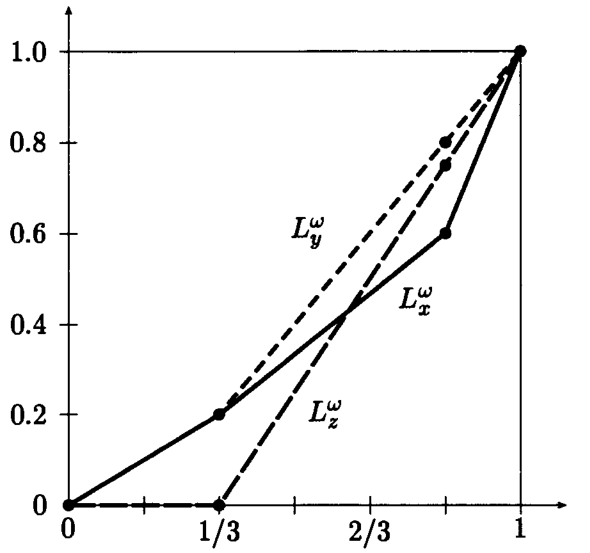


Figure 3.7: Lorenz curves for Example 3.3.1

The Lorenz-ordering is related to the lexicographic ordering. Since the elements of x^ω represent the slope of the Lorenz-curve L_x^ω, we must have that

$$x >_{LD}^{\omega} y \Rightarrow x >_{LEX}^{\omega} y \tag{3.18}$$

The lexicographic ordering imposed by $>_{LEX}^{\omega}$ is, in contrast to the Lorenz-ordering, complete. In the example illustrated by Figure 3.7, we have $y >_{LEX}^{\omega} x >_{LEX}^{\omega} z$. A maximal, with respect to the ordering imposed by $>_{LEX}^{\omega}$, element in X^* will in the sequel be referred to as a *lexicographically optimal solution* with respect to the weight vector ω. This solution is studied by Megiddo (1974) and Megiddo (1977) for the special case where $\omega_i = 1$ for all $i \in N$. Fujishige (1980) utilizes polymatroid theory to solve the problem for general weight vectors. Algorithms for solving the problem are provided in Megiddo (1977) and Fujishige (1980).

Theorem 3.7.1 *The following statements are true with respect to the system (3.17):*

(i) The system has a unique solution, which coincides with the weighted constrained egalitarian solution of the flow sharing game and the lexicographically optimal solution of the flow sharing problem.

(ii) A β^ω-sequence $\{x^t\}_{t=0}^{\infty}$, where $x^0 \in X^*$, will converge to the solution.

Proof. Hokari (1998) shows that, for convex games, the weighted constrained egalitarian solution satisfies the converse reduced game property. Then, in order to show that (3.17) coincides with the weighted constrained egalitarian solution of the flow sharing game, we need to show that (3.17) is equivalent to the solution obtained by solving all two-person reduced games. For a two-person game $(\{i,j\},u)$, the amount allocated to player i according to the weighted constrained egalitarian solution is given by

$$x_i = \min\left[\max\left\{u(i), \frac{\omega_i u(i,j)}{\omega_i+\omega_j}\right\}, u(i,j)-u(j)\right], \tag{3.19}$$

where a special case is (3.15). Given an allocation $x \in C(v)$, the reduced game for $\{i,j\}$ is given by (3.16), and by applying the solution (3.19) to every two-player reduced game, (3.17) easily results.

Koster (1999) shows that the weighted constrained egalitarian solution for a convex game exists, and consists of the unique core point that ω-Lorenz dominates every other core point. Then, since the lexicographically optimal solution of the flow sharing problem is also unique, and since $C(v) = X^*$ from Theorem 3.5.1, coincidence of these two solutions follows from (3.18).

In order to show (ii), first, note that, because $x^0 \in X^*$ and Lemma 3.4.1, every element of the β^ω-sequence will satisfy $x^t \in X^* = C(v)$. Suppose y results from x by transferring $\beta^\omega_{ij}(x) > 0$ from j to i. I.e., $y_i = x_i + \beta^\omega_{ij}(x)$ and $y_j = x_j - \beta^\omega_{ij}(x)$, and $y_k = x_k$ for all $k \neq i,j$. Then

$$\frac{x_j}{\omega_j} > \frac{x_i}{\omega_i} \qquad \frac{y_i}{\omega_i} > \frac{x_i}{\omega_i} \qquad \frac{y_j}{\omega_j} < \frac{x_j}{\omega_j} \qquad \frac{y_j}{\omega_j} \geq \frac{y_i}{\omega_i} \tag{3.20}$$

The first inequality implies that a β^ω-transfer will always go from a relatively rich individual to a poor one. To see this, note that since $x \in X^*$ implies $x_j \leq d_{ji}(x)$, we must have both

$$x_i < d_{ij}(x) \qquad \text{and} \qquad \omega_i x_j > \omega_j x_i \Leftrightarrow \frac{x_j}{\omega_j} > \frac{x_i}{\omega_i}$$

in order to have $\beta^\omega_{ij}(x) > 0$. The next two inequalities of (3.20) says that i is better off, and that j is worse off, after the transfer, and the last inequality says that the ranking of i and j has not changed as a result of the transfer. To see why this last inequality holds, let $\gamma := \frac{\omega_i x_j - \omega_j x_i}{\omega_i+\omega_j}$. Note that $\beta^\omega_{ij}(x) > 0 \Rightarrow \beta^\omega_{ij}(x) = \min\{\gamma, d_{ij}(x) - x_i\}$, hence $y_i \leq x_i + \gamma$ and $y_j \geq x_j - \gamma$. Then

$$\frac{\omega_i y_j - \omega_j y_i}{\omega_i+\omega_j} \geq \frac{\omega_i(x_j-\gamma) - \omega_j(x_i+\gamma)}{\omega_i+\omega_j} = \frac{\omega_i x_j - \omega_j x_i}{\omega_i+\omega_j} - \gamma\frac{\omega_i+\omega_j}{\omega_i+\omega_j} = 0,$$

from which the last inequality in (3.20) follows immediately. Since (3.20) implies that $y >^{\omega}_{LEX} x$, convergence follows. □

3.8 Conclusion

The contribution of this chapter has been to relate two well-known problems. We showed in Section 3.3 how the flow sharing problem may be seen as a generalized bankruptcy problem, where there are more than one estate. In Section 3.4 we formed two-person flow sharing problems, and showed that by applying a particular principle of fairness to each two-person problem, we get a unique solution to the n-person problem. One of the solutions studied, the CEA-consistent solution, corresponds to the lexicographically optimal solution, which has been studied by other authors. The CG-consistent solution is new, and offers a different view of fairness. These two solutions can also be obtained from the flow sharing game of Section 3.5, as the constrained egalitarian solution and the nucleolus, respectively. We also showed, in Section 3.6, how the solutions can be computed via transfer schemes. In Section 3.7 we generalized the CEA-consistent solution by introducing a weight for each player.

The *CG-consistent* solution, which corresponds to the nucleolus of the flow sharing game, can also be given a weighted generalization, e.g., as in Hokari and Thomson (2000). An interesting question is whether these weighted generalizations exhaust the core, as was the case for the standard fixed tree games of Chapter 2. Another interesting problem is, as in Chapter 2, how we can find weights such that the weighted versions of the CG- and CEA-consistent solutions coincides with a given core point. In the case of the CEA-consistent solution, this problem has already been solved by Fujishige (1980).

Chapter 4

Cost Allocation in a Bank ATM Network†

4.1 Introduction

Through Automated Teller Machines (ATMs), financial organizations (hereafter called banks) provide service, e.g. cash withdrawals, to their customers. For various reasons, networks of ATMs have formed, consisting of several banks, where customers of one bank may use ATMs of any bank in the network. In such a system, there is a difference between the costs that are incurred by a bank, and the costs that are actually caused by that bank's customers. Such imbalances in network usage may be compensated for by setting *interchange fees.*[1] Every time a customer of bank i uses an ATM of bank j, bank i has to pay a fee f_{ij} to bank j. Setting interchange fees is equivalent to allocating the total cost arising in such a network, and the fee structure will be the result of a negotiation process involving the participating banks.

The (transaction) costs arising in such a network depends on how transactions are processed. If the ATMs of the network are not easily accessible for the customers of the member banks, the customers will tend to use alternative means of processing their transactions, e.g. withdrawing cash over the counter. Also, the cost of processing a transaction will be higher if the

†This chapter is based on Bjørndal, Hamers and Koster (1999).
[1]See Gow and Thomas (1998) for an example from the UK.

processing involves linking computer systems of different institutions, than if no such links are necessary. The availability, for a particular customer, of ATMs belonging to the network to which his bank is affiliated, and in particular, of ATMs belonging to his own bank, depends on the physical location of the customer as well as of the various ATMs in the network, since the customer need to be physically present at the site of an ATM in order to be able to use it.

We model this cost allocation problem as a cooperative game with transferable utility, and in doing so, we explicitly model the location of customers (transactions) and ATMs. For examples of other applications of cooperative game theory to cost allocation, see e.g. Littlechild (1977) and Nouweland et al. (1996). A key question is whether there exist cost allocations that insure against break-up of the network. Given such an allocation, it should not be possible for any groups of banks to lower their costs by leaving the network. This requirement is related to the *core* of the corresponding game. Since finding a core allocation means checking a very large number of core inequalities, we would like to be able to deduce such allocations directly from the problem data, i.e., not explicitly considering the game. By relating such "natural" allocations to other solution concepts, such as the Shapley value (Shapley (1953)), the nucleolus (Schmeidler (1969)), and the τ-value (Tijs (1981)), we can learn something about e.g. the location of the allocation within the core. Another interesting question is the properties of the allocation method in a dynamic context. Assuming that there are benefits resulting from cooperation, i.e., the game has a nonempty core, we would like the allocation methods to be such that it facilitates the enlargement of the network. When a new bank wants to join the network, the existing members should not loose by accepting it as a new member. This is related to the concept of ***population monotonic allocation schemes***, Sprumont (1990).

This chapter is similar in spirit to Gow and Thomas (1998), but our approach differs from theirs in that they do not consider explicitly the location of ATMs and transactions. Another difference is that we do not consider fixed costs. In fact, our *cost savings game* would not be influenced by the inclusion of fixed costs in the manner of Gow and Thomas (1998).[2]

In Section 4.2, we introduce the *ATM-game*. This cost savings game is defined by aggregating ***single-location games*** over the set of locations. In Section 4.3 we show that the single-location games correspond to ***information market games***, as defined by Muto et al. (1989) and Potters and Tijs (1989). This correspondence yields many useful results about these games, and makes us able to study the more general ATM-games in Section 4.4. We introduce two allocation rules, the *equal-split rule* and the ***transaction-based rule***. Both rules involve aggregating, over the set of locations, allocations proposed for single-location games in Section 4.3, and they only differ with respect to locations where only one bank have ATMs. The equal-split rule yields a core element that coincides with the τ-value, but is not, in general, a population monotonic allocation scheme. The transaction-based rule also yields a core element, and moreover, is always population monotonic.

4.2 ATM-games

Let N denote the set of banks (players). We define a *location* to be a city or parts thereof, and let L denote the set of locations. Let n_i^ℓ represent the number of transactions of bank $i \in N$ in location $\ell \in L$. Let A^ℓ be the set of banks that have ATMs in location l. Further, let $L_1 := \{\ell \in L : |A^\ell| = 1\}$ be the set of locations where only one bank is represented, and let $L_M := \{\ell \in L : |A^\ell > 1\}$ be the set of locations where multiple banks are represented. We will assume that $L = L_1 \cup L_M$, i.e., that there are ATMs in all locations.

With regard to the behaviour of customers, we assume that, if $S \subseteq N$ have formed a network:

A1 Transactions in a particular location will be processed by an ATM if one or more members of S have an ATM there.

A2 When a customer of bank $i \in S$ performs a transaction in a location ℓ, and if bank i has ATMs in location ℓ, the customer will use one of

[2]This would not be the case, however, if the number and location of ATMs were endogenously determined in our model.

the ATMs of bank i.

The transaction costs are assumed to be the same for all banks. The transaction cost will be α if the customer uses an ATM of his own bank. If he uses an ATM of another bank the transaction cost will be β, where $\beta > \alpha$. The cost of non-ATM transactions is complex, since there exist several alternatives to using ATMs, such as withdrawing money over the counter, writing a check to a third person in exchange for cash, or using a cashback facility. In the four-bank example of Gow and Thomas (1998), the cost of cash withdrawal over the counter is used as an approximation of this cost. We shall assume that the cost of a non-ATM transaction is γ, where $\gamma > \beta$.

Suppose S forms a network. Assumptions A1 and A2 imply, for any location $\ell \in L$, that the total amount of transaction costs in location ℓ is given by

$$c^\ell(S) := \begin{cases} \sum_{i\in S\cap A^\ell} \alpha n_i^\ell + \sum_{i\in S\setminus A^\ell} \beta n_i^\ell & \text{if } S\cap A^\ell \neq \emptyset, \\ \sum_{i\in S} \gamma n_i^\ell & \text{otherwise.} \end{cases}$$

S	$\{A\}$	$\{B\}$	$\{C\}$	$\{A,B\}$	$\{A,C\}$	$\{B,C\}$	N
$c^\ell(S)$	100	1500	200	400	300	500	600

Figure 4.1: The values for c^ℓ in Example 4.2.1

Example 4.2.1 [Figure 4.2] Consider a location ℓ where the three banks A, B, and C, have customers. The numbers of customers of these banks at ℓ are $n_A^\ell = 100, n_B^\ell = 150$, and $n_C^\ell = 200$. Banks A and C have ATMs at ℓ. The cost of an ATM transaction is $\alpha = 1$ for customers serviced by their own bank, and $\beta = 2$ otherwise. Every non-ATM transaction involves cost $\gamma = 10$. In the previous terminology we have $A^\ell = \{A, C\}$. Then these parameters fix the coalitional cost game c^ℓ. The cost at which the coalition $\{A\}$ is able to service all its customers is $c^\ell(\{A\}) = \alpha n_A^\ell = 100$, since $A \in A^\ell$. Similarly, we calculate $c^\ell(\{C\}) = \alpha n_C^\ell = 200$. Since $B \notin A^\ell$, we have $c^\ell(\{B\}) = \gamma n_B^\ell = 1500$. The cost of serving the customers of A

and B together are $c^\ell(\{A,B\}) = \alpha n_A^\ell + \beta n_B^\ell = 100 + 300 = 400$. The customers of A and C are all serviced by the ATM of their own bank, hence $c^\ell(\{A,C\}) = \alpha(n_A^\ell + n_C^\ell) = 300$. In this way we compute the cost associated with each coalition, as shown in Figure 4.2. $\triangleleft$

In order to relate our game to existing literature, it will be convenient to study the corresponding *cost savings game* v^ℓ. Let $s_i^\ell := (\gamma - \beta) n_i^\ell$ denote the cost savings that occur if transactions of bank $i \in N \setminus A^\ell$ can be processed via an ATM of another bank. The *single-location ATM-game* v^ℓ is given by, for any $S \subseteq N$,

$$v^\ell(S) := \sum_{i \in S} c^\ell(\{i\}) - c^\ell(S) = \begin{cases} \sum_{i \in S \setminus A^\ell} s_i^\ell & \text{if } S \cap A^\ell \neq \emptyset, \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

Since all the solution concepts that we will study are relatively invariant under strategic equivalence[3], all results for a cost savings game can easily be translated into the setting of a cost game.

S	$\{A\}$	$\{B\}$	$\{C\}$	$\{A,B\}$	$\{A,C\}$	$\{B,C\}$	N
$v^\ell(S)$	0	0	0	1200	0	1200	1200

Figure 4.2: The values for v^ℓ in Example 4.2.2

Example 4.2.2 [Figure 4.2] Now we return to Example 4.2.1. The coalitional cost savings for that example are specified in Figure 4.2. Notice that the cost savings, i.e., the values of v^ℓ, arise from transactions of banks that do not have ATMs in location ℓ, i.e., bank B in this case. Notice also the zero values of single player coalitions. Single players save no costs, regardless of whether they have ATMs or not. $\triangleleft$

The *ATM-game* v is obtained by aggregating over the set of locations, i.e.,

[3]A solution concept σ is said to be *relatively invariant under strategic equivalence* iff, whenever v is a game with $\sigma(v) \neq \emptyset$, $a \in \mathbb{R}^N$, and $b > 0$, then $\sigma(a + bv) = a + b\sigma(v)$. If c is a cost game, and v is the corresponding cost savings game, then $x \in \sigma(c) \Leftrightarrow y \in \sigma(v)$, where $y_i = c(\{i\}) - x_i$ for all $i \in N$. Note that in going from the cost game c to the cost savings game v, we also change the interpretation (sign) of the game, and the definition of the solution concepts must be changed accordingly, cf. Section 1.3.

let, for every $S \subseteq N$,

$$v(S) := \sum_{\ell \in L} v^{\ell}(S) = \sum_{\ell \in L: S \cap A^{\ell} \neq \emptyset} s^{\ell}(S \setminus A^{\ell}). \tag{4.2}$$

where, for any $R \subseteq N$, $s^{\ell}(R) := \sum_{i \in R} s_i^{\ell}$. In order to study some properties and solutions of single-location ATM-games, we recall the solution concepts core, Shapley value, and the nucleolus, defined in Section 1.3. Here we will use an alternative definition of the Shapley value as the average of the *marginal vectors* of the game. In order to define this concept, consider a game with characteristic function g and player set N. Let Π be the set of all orderings of the player set. Take a player $i \in N$ and an order $\pi \in \Pi$. The ith coordinate of the marginal vector $m^{\pi}(v)$ is given by

$$m_i^{\pi}(g) := g(\{\pi(1), \dots, \pi(k-1), \pi(k)\}) - g(\{\pi(1), \dots, \pi(k-1)\}),$$

where $i = \pi(k)$. The Shapley value is equal to the average, over the set Π, of the marginal vectors, i.e.,

$$\Phi(g) = \frac{1}{|\Pi|} \sum_{\pi \in \Pi} m^{\pi}(g).$$

If the game v is convex, then we know from Shapley (1971) that $C(g) = \text{conv}\{m^{\pi}(g) : \pi \in \Pi\}$, and hence that $\Phi(g) \in C(g)$.

We will also study the *τ-value* (Tijs (1981)), a compromise between the *minimum right vector* $m(g)$ and the *utopia vector* $M(g)$. Note that player i cannot get more than $M_i(g) := g(N) - g(N \setminus \{i\})$ in any core allocation, since otherwise, the core inequality for $N \setminus \{i\}$ would be violated. On the other hand, he cannot get less than

$$m_i(g) := \max_{S \ni i} \left(g(S) - \sum_{j \in S \setminus \{i\}} M_j(g) \right), \tag{4.3}$$

since, for $x \in C(g)$, and for any S such that $i \in S$, we have $x_i = x(S) - x(S \setminus \{i\}) \geq g(S) - \sum_{j \in S \setminus \{i\}} M_j(g)$. The utopia vector and the minimum rights vector form upper and lower bounds, respectively, for the core, i.e.,

$$x \in C(g) \Rightarrow m(g) \leq x \leq M(g). \tag{4.4}$$

The τ-value of g is the (unique) convex combination of $m(g)$ and $M(g)$ that satifies $\sum_{i \in N} \tau_i(g) = g(N)$.

4.3 Properties and solutions of single-location games

Because of (4.2), an allocation x will satisfy $x \in C(v)$ if $x_i = \sum_{\ell \in L} x_i^\ell$ for all $i \in N$, and $x^\ell \in C(v^\ell)$ for all $\ell \in L$. This, and the additivity of the Shapley value, suggests that we study the games v^ℓ, $\ell \in L$, in order to learn more about the properties of the game v.

Our game can be related to the class of *information market games*, see Muto et al. (1989) and Potters et al. (1989). An information market game consists of a set of players N, where a subset $I \subset N$ possesses information about a (patented) new technology, necessary for producing a new product. The total market for this new product can be partitioned into sub-markets, and the profit realized by a coalition depends on which sub-markets the coalition has access to. Let M_T denote the set of sub-markets that the coalition T has access to, and let r_T denote the profit that can be realized from these sub-markets. A coalition S can realize the profit r_T if it has at least one member with access to the sub-markets M_T, i.e., $S \cap T \neq \emptyset$, as well as at least one member with knowledge of the patented technology, i.e., $S \cap I \neq \emptyset$. Therefore, the total profit that can be realized by the members of S is given by

$$v_{I,r}(S) = \begin{cases} \sum_{T: T \cap S \neq \emptyset} r_T & \text{if } S \cap I \neq \emptyset \\ 0 & \text{otherwise,} \end{cases} \tag{4.5}$$

thus defining the information market game $(N, v_{I,r})$. Muto et al. (1989) show that, if $|I| = 1$, then the nucleolus and the τ-value coincides. If $|I| = 1$ and $r_T = 0$ for all $T \subseteq N$ such that $|T| \geq 2$, then $(N, v_{I,r})$ is convex, and the Shapley value coincides with the nucleolus and the τ-value. Potters and Tijs (1989) show that, if $|I| \geq 2$, and if $r_T = 0$ for all $T \subseteq N$ such that $|T| \geq 2$, then the core consists of a single point.

Proposition 4.3.1 *A single-location ATM-game is an information market games.*

Proof. We obtain $v^\ell = v_{I,r}$ by setting $I := A^\ell$ and

$$r_T := \begin{cases} s_i^\ell & \text{if } T = \{i\} \subset N \setminus A^\ell, \\ 0 & \text{otherwise.} \end{cases}$$

□

4.3.1 Locations where only one bank has ATMs

Because of Proposition 4.3.1, and since $|A^\ell| = 1$, this case is covered by Muto et al. (1989). Moreover, since $r_T = 0$ for all $T \subseteq N$ such that $|T| \geq 2$, we get the following result.

Proposition 4.3.2 *If $\ell \in L_1$, then the game v^ℓ is convex.*

We denote the bank having ATMs in location ℓ by i^ℓ.

Theorem 4.3.3 *If $\ell \in L_1$, then*

(i) $C(v^\ell) = \mathrm{conv}\{m^\pi(v^\ell) : \pi \in \Pi\}$

(ii) $C(v^\ell) = \{x \in \mathbf{R}^N : x(N) = v^\ell(N),\ 0 \leq x_i \leq s_i^\ell \ \forall i \in N \setminus \{i^\ell\}\}$

Proof. The convexity of v^ℓ and Shapley (1971) imply (i), and (ii) follows from Muto et al. (1989). □

Theorem 4.3.3(i) is useful here, because the vectors of marginal contributions have a simple structure in our case. For any $\pi \in \Pi$, let

$$S_\pi^\ell := \{i \in N : \pi^{-1}(i) < \pi^{-1}(i^\ell)\},$$

i.e., S_π^ℓ is the set of players that precede i^ℓ in the order π.

Proposition 4.3.4 *If $\ell \in L_1$, then*

$$m_i^\pi(v^\ell) = \begin{cases} 0 & \text{if } i \in S_\pi^\ell, \\ s^\ell(S_\pi^\ell) & \text{if } i = i^\ell, \\ s_i^\ell & \text{if } i \in N \setminus S_\pi^\ell. \end{cases} \tag{4.6}$$

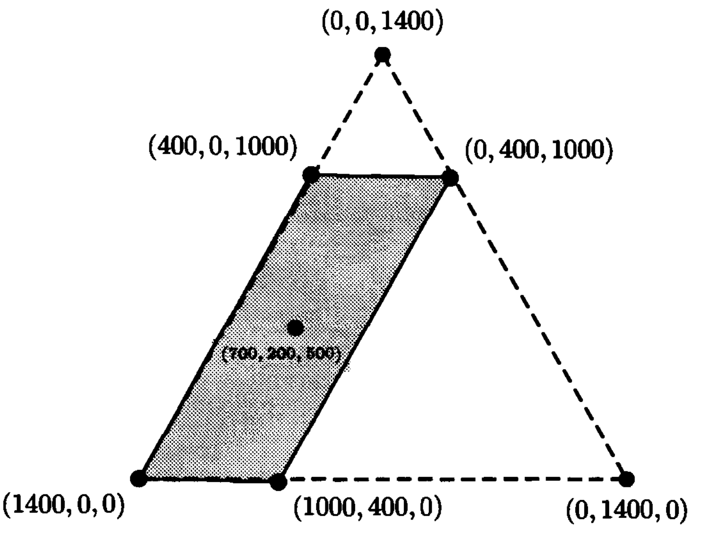
Proof. Consider an arbitrary player $i \in N$. If $i \neq i^\ell$, and i joins a coalition S, then an additional cost saving of s_i^ℓ will be realized, but only if i^ℓ is already a member of S. If $i = i^\ell$, then i will provide cost savings for all the players that are already in S. □

S	$\{A\}$	$\{B\}$	$\{C\}$	$\{A,B\}$	$\{A,C\}$	$\{B,C\}$	N
$v^{\ell}(S)$	0	0	0	400	1000	0	1400

(a) The game

π	$m_A^{\pi}(v^{\ell})$	$m_B^{\pi}(v^{\ell})$	$m_C^{\pi}(v^{\ell})$
A,B,C	0	400	1000
A,C,B	0	400	1000
B,A,C	400	0	1000
B,C,A	1400	0	0
C,A,B	1000	400	0
C,B,A	1400	0	0
Average	700	200	500

(b) Marginal vectors



(c) The core

Figure 4.3: Example 4.3.5

Example 4.3.5 [Figure 4.3] Consider a situation where the banks A, B, and C have customers in the location ℓ. Let $n_A^\ell = 200, n_B^\ell = 50$, and $n_C^\ell = 125$. Also, as in Example 4.2.1, let $\alpha = 1$, $\beta = 2$, and $\gamma = 10$. Only bank A has ATMs in the location. The values of the game v^ℓ are shown in Figure 4.3(a), and the marginal vectors are shown in Figure 4.3(b). From the picture of the core shown in Figure 4.3(c), the coincidence of the extreme points of the core with the marginal vectors can indeed be verified. Observe that, although there are six different orderings of the player set, there are only four distinct marginal vectors. Both the marginal vectors for which bank A comes first (last) coincides, as is easily seen from (4.6). The average of the marginal vectors, i.e., the Shapley value, is a member of the core, which follows from Proposition 4.3.2 and Theorem 4.3.3(i). $\triangleleft$

Theorem 4.3.6 *If $\ell \in L_1$, then $\Phi(v^\ell) = w$, where*

$$w_i = \begin{cases} \dfrac{s^\ell(N \setminus \{i\})}{2} & \text{if } i = i^\ell, \\ \dfrac{s_i^\ell}{2} & \text{otherwise.} \end{cases} \tag{4.7}$$

Proof. By taking the average, over the set of all orderings of the players, of the marginal vectors given by (4.6), we get the Shapley value. Note that, if $i \neq i^\ell$, the number of orderings such that i precedes i^ℓ equals the number of orderings such that i comes after i^ℓ, and the average payoff of i in the marginals is $\frac{s_i^\ell}{2}$. On the other hand, a player $j \in N \setminus \{i^\ell\}$ will precede i^ℓ in half the orderings, and the average cost savings that i^ℓ will be rewarded from the transactions of j will be $\frac{s_j^\ell}{2}$. The result could also have been found by using Muto et al. (1989), and the fact that v^ℓ is a convex game. $\square$

By Theorem 4.3.3(i), the Shapley value is a core allocation of v^ℓ. Another core allocation is the τ-value, which is is related to the core characterization of Theorem 4.3.3(ii), and is a compromise between the utopia vector $M(v^\ell)$ and the minimum rights vector $m(v^\ell)$. If $i = i^\ell$, then the utopia payoff of i

is

$$M_i(v^\ell) = v^\ell(N) - v^\ell(N \setminus \{i\}) = s^\ell(N \setminus \{i^\ell\}) - 0$$
$$= s^\ell(N \setminus \{i^\ell\}),$$

i.e., bank i can not hope to get more than the total cost savings for banks without ATMS. On the other hand, if $i \neq i^\ell$, i.e., bank i does not have an ATM in location ℓ, then

$$M_i(v^\ell) = s^\ell(N \setminus \{i^\ell\}) - s^\ell(N \setminus \{i^\ell, i\}) = s_i^\ell,$$

i.e., bank i can not hope to get more than his own cost savings. If $i = i^\ell$, then a lower bound on i's share of the cost savings is

$$m_i(v^\ell) = \max_{S \ni i} \left[v^\ell(S) - \sum_{j \in S \setminus \{i\}} M_j(v^\ell) \right]$$
$$= \max_{S \ni i} \left[s^\ell(S \setminus \{i^\ell\}) - s^\ell(S \setminus \{i\}) \right] = 0,$$

i.e., the worst-case scenario is that i does not get any part of the cost savings at all. If $i \neq i^\ell$, then $m_i(v^\ell)$ is the maximum of

$$\max_{\substack{S \ni i \\ S \ni i^\ell}} \left[v^\ell(S) - \sum_{j \in S \setminus \{i, i^\ell\}} M_j(v^\ell) - M_{i^\ell}(v^\ell) \right]$$
$$\max_{\substack{S \ni i \\ S \ni i^\ell}} \left[s^\ell(S \setminus \{i^\ell\}) - s^\ell(S \setminus \{i, i^\ell\}) - s^\ell(N \setminus \{i^\ell\}) \right]$$
$$= s_i^\ell - s^\ell(N \setminus \{i^\ell\}) = -s^\ell(N \setminus \{i, i^\ell\}) \leq 0$$

and

$$\max_{\substack{S \ni i \\ S \not\ni i^\ell}} \left[v^\ell(S) - \sum_{j \in S \setminus \{i\}} M_j(v^\ell) \right] = \max_{\substack{S \ni i \\ S \not\ni i^\ell}} \left[0 - s^\ell(S \setminus \{i\}) \right] = 0,$$

i.e., we have $m_i(v^\ell) = 0$. To sum up, we have

$$M_i(v^\ell) = \begin{cases} s^\ell(N \setminus \{i\}) & \text{if } i = i^\ell \\ s_i^\ell & \text{otherwise, and} \end{cases} \tag{4.8}$$

$$m_i(v^\ell) = 0. \tag{4.9}$$

Theorem 4.3.7 *If $\ell \in L_1$, then $\tau(v^\ell) \doteq \frac{1}{2}M(v^\ell) = w$, where w is given by* (4.7).

Proof. The result follows from $w = \frac{1}{2}M(v^\ell) \Rightarrow w(N) = v^\ell(N)$. Note that the result could have been proved by using Muto et al. (1989), but we have included the proof here for the sake of completeness. □

Theorem 4.3.8 *If $\ell \in L_1$, then $NU(v^\ell) = w$, where w is given by (4.7).*

Proof. The result follows from Muto et al. (1989), but the proof provided here is simpler, due to the special structure of ATM-games. Let

$$\mathcal{B}_0 := \{S \subset N : |S| = 1 \text{ and } w(S) = v^\ell(S)\},$$

and, for $k \geq 1$,

$$\mathcal{B}_k := \{S \subset N : e(v^\ell, S, w) = \max[e(v^\ell, T, w) : T \notin \cup_{j<k}\mathcal{B}_j]\}.$$

Suppose there are p such sets, and let, for $1 \leq k \leq p$, $\mathcal{D}_k := \cup_{j \leq k}\mathcal{B}_j$. Kohlberg (1971) proved that w is the nuceolus iff, for every $1 \leq k \leq p$, there exists a *balanced collection* $C \subset \mathcal{D}_k$, i.e., there exists weights $\{\lambda\}_{S \in C}$ such that $\sum_{\substack{S \in C \\ S \ni i}} \lambda_S = 1$ for all $i \in N$ and $\lambda_S > 0$ for all $S \in C$. In order to show that such balanced collections exists for the allocation w, we will show that, for all $S \subset N$,

$$e(w, v^\ell, S) = e(w, v^\ell, N \setminus S). \tag{4.10}$$

Hence, for any $1 \leq k \leq p$, a balanced collection is given by $C = \{S, N \setminus S\}$ for some $S \in \mathcal{D}_k$. To see that (4.10) holds, note that, if $i^\ell \in S$, then

$$\begin{aligned} &e(v^\ell, S, w) = v^\ell(S) - w_{i^\ell} - w(S \setminus \{i^\ell\}) \\ =&s^\ell(S \setminus \{i^\ell\}) - \frac{s^\ell(N \setminus \{i^\ell\})}{2} - \frac{s^\ell(S \setminus \{i^\ell\})}{2} = -\frac{s^\ell(N \setminus S)}{2}. \end{aligned} \tag{4.11}$$

If $i^\ell \notin S$, then

$$e(v^\ell, S, w) = 0 - \frac{s^\ell(S)}{2} = -\frac{s^\ell(S)}{2}. \tag{4.12}$$

Since $i^\ell \in S \Leftrightarrow i^\ell \notin N \setminus S$, (4.10) follows. □

4.3.2 Locations where multiple banks have ATMs

In order to investigate the structure of the core for this case, we compute the utopia vector and the minimum rights vector, which form upper and lower bounds, respectively, for core allocations. We will shown that the bounds coincide, hence the core consists of a single point. For every $i \in N$ we have

$$M_i(v^\ell) = v^\ell(N) - v^\ell(N \setminus \{i\}) = \begin{cases} 0 & \text{if } i \in A^\ell, \\ s_i^\ell & \text{otherwise.} \end{cases} \tag{4.13}$$

If $i \in N \setminus A^\ell$, then $m_i(v^\ell)$ is the maximum of

$$\begin{aligned} &\max_{\substack{S \cap A^\ell \neq \emptyset \\ S \ni i}} \left[v^\ell(S) - \sum_{j \in S \setminus \{i\}} M_j(v^\ell) \right] \\ &= \max_{\substack{S \cap A^\ell \neq \emptyset \\ S \ni i}} \left[s^\ell(S \setminus A^\ell) - s^\ell(S \setminus A^\ell \setminus \{i\}) \right] = s_i^\ell, \end{aligned}$$

and

$$\max_{\substack{S \cap A^\ell = \emptyset \\ S \ni i}} \left[v^\ell(S) - \sum_{j \in S \setminus \{i\}} M_j(v^\ell) \right] = \max_{\substack{S \cap A^\ell = \emptyset \\ S \ni i}} \left[0 - s^\ell(S \setminus \{i\}) \right] = 0,$$

hence

$$m_i(v^\ell) = s_i^\ell \tag{4.14}$$

If $i \in A^\ell$, then

$$\begin{aligned} m_i(v^\ell) &= \max_{S \ni i} \left\{ v^\ell(S) - \sum_{j \in S \setminus \{i\}} M_j(v^\ell) \right\} \\ &= \max_{S \ni i} \{ s^\ell(S \setminus A^\ell) - s^\ell(S \setminus A^\ell \setminus \{i\}) \} = 0. \end{aligned} \tag{4.15}$$

From (4.13)-(4.15) follows that $M(v^\ell) = m(v^\ell)$. This gives us useful information about the core, which in this case turns out to consist of a single point.

Theorem 4.3.9 *If $\ell \in L_M$, then $C(v^\ell) = \{x\}$, where x is given by*

$$x_i = \begin{cases} 0 & \text{if } i \in A^\ell, \\ s_i^\ell & \text{otherwise.} \end{cases} \tag{4.16}$$

Proof. Theorem 4.3.9 also follows from Proposition 4.3.1 and Potters and Tijs (1989). In order to see that the core is nonempty, note that, for any $S \subseteq N$,

$$x(S) = s^\ell(S \setminus A^\ell) \geq v^\ell(S),$$

and that $x(N) = s^\ell(N \setminus A^\ell) = v^\ell(N)$. Hence $x \in C(v^\ell)$.

Uniqueness follows from (4.4) and $m(v^\ell) = x = M(v^\ell)$. □

Example 4.3.10 Recall the situation described in Example 4.2.1 and Example 4.2.2. Bank B is the only one that does not have ATMs, so the only cost savings are those involving B's transactions. According to (4.16), B will be allowed to keep the entire cost savings himself, and this yields the unique core allocation $(0, 1200, 0)$. The game v^ℓ is not convex, since

$$v^\ell(\{A,B\}) - v^\ell(\{B\}) = 1200 > v^\ell(\{A,B,C\}) - v^\ell(\{B,C\}) = 0,$$

and the Shapley value, given by $\Phi(v^\ell) = (200, 800, 200)$, is not a core element. ◁

Theorem 4.3.11 *If $\ell \in L_M$, then $x = M(v^\ell) = \tau(v^\ell) = NU(v^\ell)$, where x is given by* (4.16).

The result also follows from Proposition 4.3.1 and Potters and Tijs (1989).

Proof. Since $x(N) = v^\ell(N)$, the second equality follows from $m(v^\ell) = M(v^\ell)$. The third equality follows since the nucleolus is always a core point, if the core is nonempty. □

4.4 Two allocation rules for multiple-location games

Now we turn to the ATM-game v defined by (4.2). In Section 4.3 we proposed allocation rules for single-location ATM-games. In the following we will discuss two allocation rules for situations with multiple locations. These rules aggregate, over the set of locations, the allocation vectors proposed for single locations. We shall relate the resulting solutions to solution concepts such as the core and the τ-value. Also, we will investigate whether these allocations rule correspond to *population monotonic allocation schemes*, as defined by Sprumont (1990). Let $P(N)$ denote the set of nonempty subsets of N.

Definition 4.4.1 A vector $d = (d_{iS})_{i \in S, S \in P(N)}$ is a population monotonic allocation scheme of the game v if and only if

$$\sum_{i \in S} d_{iS} = v(S) \qquad \forall S \in P(N), \tag{4.17}$$

$$d_{iS} \leq d_{iT} \qquad \forall i \in S \subseteq T \in P(N). \tag{4.18}$$

Hence, d specifies an allocation for every game corresponding to the population $S \subseteq N$, where the characteristic function is given by the restriction of v to the members of S. Condition (4.17) expresses that the entire cost $v(S)$ should be covered, and (4.18) that no player should be made worse off as new players enter the games. In the ATM network situation, population monotonicity ensures that members of an existing network will not object to new banks joining the network.

4.4.1 The equal-split rule

We first propose an allocation rule that splits the cost savings equally between owners of transactions and the owner of the ATMs. Thus, for $\ell \in L_1$ we use w^ℓ given by (4.7), and for $\ell \in L_M$ use x^ℓ given by (4.16). This yields the allocation y given by, for every $i \in N$,

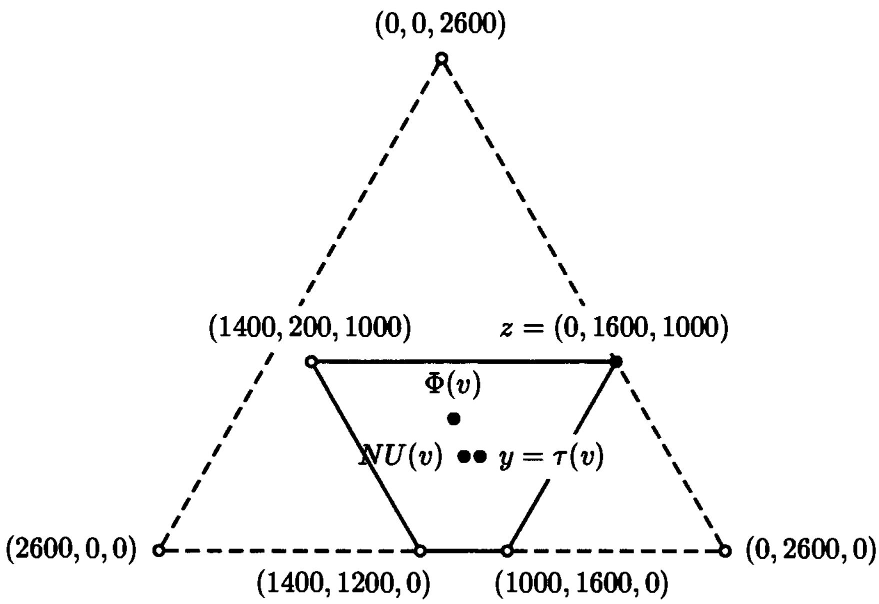
$$\begin{aligned} y_i &= \sum_{\ell \in L_1} w_i^\ell + \sum_{\ell \in L_M} x_i^\ell \\ &= \sum_{\ell \in L_1 : i \notin A^\ell} \frac{s_i^\ell}{2} + \sum_{\ell \in L_1 : i \in A^\ell} \frac{s^\ell(N \setminus \{i\})}{2} + \sum_{\ell \in L_M : i \notin A^\ell} s_i^\ell. \end{aligned} \tag{4.19}$$

Since the equal-split rule adds core elements of the games v^ℓ, the following result is obvious.

Theorem 4.4.2 *The equal-split rule gives a core element of the game v.*

S	$\{A\}$	$\{B\}$	$\{C\}$	$\{A,B\}$	$\{A,C\}$	$\{B,C\}$	N
$v(S)$	0	0	0	1600	1000	1200	2600

(a) The game



(b) The core (area enclosed by solid lines)

Figure 4.4: Example 4.4.3

Example 4.4.3 [Figure 4.4] Consider a situation with two locations, i.e., $L := \{1, 2\}$, and three banks, i.e., $N = \{A, B, C\}$. In location 1 the banks have 100, 150, and 200 transactions, respectively. Here, bank A and C have ATMs. In location 2 the banks have 200, 50, and 125 transactions, respectively, and only bank A has ATMs there. The locations correspond to those described in Example 4.2.1 and 4.3.5, respectively. As before, $\alpha = 1$, $\beta = 2$, and $\gamma = 10$. The values of the resulting game v are shown in Figure 4.4(a), and a picture of the core in Figure 4.4(b). In location 1, where both bank A and C have ATMs, the equal-split rule prescribes the allocation $(0, 1200, 0)$, and in location 2, where only bank A has ATMs, the allocation $(700, 200, 500)$. Summing the allocation vectors, we get $y = (700, 1400, 500)$. From Figure 4.4(b) it can be seen that this is a point in the relative interior of the core. It coincides with the τ-value, but not with the Shapley value or the nucleolus, given by the allocation vectors $(900, 1000, 700)$ and $(950, 1150, 500)$, respectively. $\triangleleft$

Theorem 4.4.4 *If y results from the equal-split rule, then $y = \tau(v)$.*

Proof. In general the τ-value is not additive, hence we must show that $\tau(v) = \sum_{\ell \in L} \tau(v^\ell)$ holds. Theorem 4.3.7 states that $\tau(v^\ell) = \frac{1}{2}M(v^\ell)$ for $\ell \in L_1$, and Theorem 4.3.11 that $\tau(v^\ell) = M(v^\ell)$ for $\ell \in L_M$. Hence, the τ-values for single-location games depend only on the utopia vectors, for which $M(v) = \sum_{\ell \in L} M(v^\ell)$ holds. $\square$

The equal-split does not always satisfy population monotonicity, as the following example illustrates.

S	$\{A\}$	$\{B\}$	$\{C\}$	$\{A,B\}$	$\{A,C\}$	$\{B,C\}$	N
$v(S)$	0	0	0	p	$q+p$	q	$q+2p$

Figure 4.5: The game v of Example 4.4.5

Example 4.4.5 [Figure 4.4.1] Consider a situation with two locations, i.e., $L = \{1, 2\}$, and three banks, i.e., $N = \{A, B, C\}$. A has ATMs in both locations, B in location 1, and C in neither location. Hence, $A^1 = \{A, B\}$, and $A^2 = \{A\}$. The number of transactions for each bank $i \in N$ and each location $\ell \in L$ is

$$n_i^\ell = \begin{cases} q & \text{if } i = C \text{ and } \ell = 1, \\ p & \text{otherwise,} \end{cases}$$

where $p < q$. Let $\gamma - \beta = 1$. The cost savings realized by the various coalitions $S \subseteq N$, i.e., the values of the game v, are shown in Figure 4.4.1.

Suppose that B and C have formed a network. The cost savings that they realize are q, involving only the transactions of C in location 1, where B is present. According to the equal-split rule, they will both get a payoff of

$$\frac{q}{2}.$$

Suppose that A wants to join the network. It will provide cost savings for both B and C in location 2, where it is the only bank present, and the total cost savings now increase to $2p+q$. The equal-split rule yields the allocation

$$\left(p, \frac{p}{2}, q+\frac{p}{2}\right).$$

Hence, bank B, since $p < q$, will have its payoff reduced, and it is therefore likely that it will oppose A being accepted as a new participant in the network. ◁

However, there are situations in which even the equal-split rule satisfies population monotonicity, as the next example shows.

Example 4.4.6 Consider Example 4.4.5 again, but now with the assumption that $p = q$, i.e., all the banks have the same number of transactions, in all locations $\ell \in L$. We will check that the allocation scheme $(y_{iS})_{S \in P(N)}$ corresponding to the equal-split rule is population monotonic. If a bank operate on its own, i.e., if $|S| = 1$, no cost savings will be realized, and we have $y_{iS} = 0$ for all $i \in N$. If all the banks participate in the network, the total cost savings will be $3p$, and the equal-split rule yields the allocation

$$y_N = \left(p, \frac{p}{2}, \frac{3p}{2}\right).$$

Since the equal-split rule always assigns positive payoffs to all banks, we only need to check, for each $S \subset N$ such that $|S| = 2$, that $y_{iS} \leq y_{iN}$. The equal-split rule, when applied to the network formed by S, gives bank $i \in S$ the payoff

$$y_{iS} = \begin{cases} \frac{p}{2} & \text{if } S = \{A, B\}, \\ p & \text{if } S = \{A, C\}, \\ \frac{p}{2} & \text{if } S = \{B, C\}, \end{cases}$$

hence population monotonicity is satisfied. ◁

4.4.2 The transaction-based rule

Suppose that banks with ATMs are given no reward for the cost savings that they provide for the banks without ATMs. The cost savings are rewarded to the bank owning the *transactions* for which the savings are realized. Thus, for every $\ell \in L$, we choose x^ℓ as defined by (4.16), and then we sum over the set of locations. This yields the allocation vector z given by, for every $i \in N$,

$$z_i = \sum_{\ell \in L} x_i^\ell = \sum_{\substack{\ell \in L \\ i \notin A^\ell}} s_i^\ell. \tag{4.20}$$

Theorem 4.4.7 *The transaction-based rule gives a core element of v.*

Proof. We know that $x^\ell \in C(v^\ell)$ for all $\ell \in L$, which follows from Theorem (4.3.3)(ii) and Theorem 4.3.9. □

Example 4.4.8 We apply (4.16) to location 1 and 2, respectively, and get the allocations vectors $(0, 1200, 0)$ and $(0, 400, 1000)$. By summing these, we get the allocation vector $z = (0, 1600, 1000)$. From Figure 4.4(b) we see that this corresponds to one of the extreme points of the core.[4] ◁

We saw in Section 4.4.1 that the equal-split rule is not necessarily population monotonic. The transaction-based rule is better in this respect.

Theorem 4.4.9 *The transaction-based rule is a population monotonic allocation scheme.*

Proof. Let $L^S := \{\ell \in L : S \cap A^\ell \neq \emptyset\}$, and, for every $i \in S$, $z_{iS} = \sum_{\substack{\ell \in L^S \\ i \notin A^\ell}} s_i^\ell$. Since $s_i^\ell \geq 0$ for all $i \in N$, $S \subseteq T$ implies $L^S \subseteq L^T$, we have $z_{iS} \leq z_{iS}$ for $i \in S \subseteq T$. □

[4]That this is not so in general can be verified by e.g. adding a location 3 to the example, where $n_i^3 = 100$ for all $i \in N$, and $A^3 = \{B\}$.

4.5 Conclusion

This chapter presents a simple model of a bank ATM network. Assumptions A1 and A2 of Section 4.2 imply that the single-location games studied in Section 4.3 are information market games, which have a particularly simple structure. In the case where only one bank has ATMs in a location, the core is relatively large, and several important solution concepts coincide with a central point in the core. In the case where more than one bank has ATMs, the core consists of a single point. By combining allocations for single-location games via the *equal-split rule* of Section 4.4, we were able to obtain τ-value, which is a core point. The *transaction-based rule* also yields a core point, and, in contrast to the equal-split rule, this rule is population-mononotonic, i.e., no bank will lose as a result of a new bank entering the network.

The simplistic assumptions with respect to the behaviour of the bank customers may be responsible for the elegant results obtained, and should be made more realistic, e.g., by introducing distances.

Chapter 5

Lower and Upper Bounds for Linear Production Games

5.1 Introduction

We study a model of a production economy, in which the production technology is given by linear relationships, and where every group of agents have access to the same technology. There is a set of resources R that can be used to produce a set of products P. The production technology is given by a matrix A, where a_{ij} is the amount of resource i needed to produce one unit of product j. It is assumed that an infinite amount of product j can be sold at the price c_j, giving the price vector c. The resources available is given by a vector b, where b_i is the amount available of resource i. The maximal profit that can be made from the resource bundle b is given by

$$\max\left\{c^T x : Ax \leq b,\ x \in \mathbf{R}_+^p\right\}, \tag{5.1}$$

where x_j denotes the amount of product j that is produced.

The resources are owned by a set N of agents, and ownership of the resources is shared among the agents. The agents may operate on their own, or they may combine their resources in order to increase the total profit. Assuming that cooperation leads to an increase in the total profit, the agents need to agree on how to share this profit among themselves, and the way in which they do this will influence the incentives for them to cooperate or not. It may, for example, be the case that a group of agents receives so little of the total profit that they will be better off by forming their own production facility than putting their resources into the joint production facility. The problem of finding an allocation of the profit can be modeled as a TU-game, such as in Owen (1975), providing us with solution concepts such as the core. Generalizations, with respect to how resources are controlled by various subsets (coalitions) of agents, have been studied by Granot (1986) and Curiel et al. (1989).

To describe a solution to a TU-game, we need to know not only the profit that can be made by N, but also the corresponding values for some or all of the subsets $S \subset N$. Since there are $2^n - 1$ such subsets, the amount of computational work involved can be prohibitive. In this chapter we present a method that provides us with lower and upper bounds on $v(S)$ for any $S \subseteq N$, while requiring less computational effort than actually computing $v(S)$. Our method is related to aggregation of columns and rows in linear programming problems, as in Zipkin (1980b) and Zipkin (1980a), respectively.

In Section 5.2 we define linear production processes and linear production games, as well as some concepts related to cooperative game theory. Section 5.3 describes how lower and upper bounds for linear production games can be found by aggregating columns and rows, respectively, and in Section 5.4 we give a method to find bounds on the error resulting from the aggregation. The method involves solving a mixed integer programming problem, and the solution from this problem also suggests how the weight matrix of the aggregated game may be updated in order to improve the bound. Finally, in Section 5.5, we investigate, using numerical examples, how the performance of the aggregation approach depends on the structure of the problem data.

5.2 Linear production games

The set of *agents* (players) is denoted by N, the set of *resources* by R, and the set of *products* by P, where $n := |N|$, $r := |R|$, and $p := |P|$. The production technology is described by the matrix $A \in \mathbf{R}^{R \times P}$, where a_{ij} is the amount of resource i needed to produce one unit of product j. The profit per unit sold of product j is c_j, making up the column[1] vector $c \in \mathbf{R}^P$. The ownership of the resources is described by the vector-valued function[2] $b : 2^N \to \mathbf{R}^R$, where $b_i(S)$ is the amount of resource i that the subset $S \subseteq N$ controls.

Definition 5.2.1 The triple (A, b, c) is a *linear production process* if

(i) $a_{ij} \geq 0$ for all $i \in R$ and $j \in P$,

(ii) $b_i(S) \geq 0$ for all $i \in R$ and $S \in 2^N$,

(iii) if $c_j > 0$, then there exists some resource i such that $a_{ij} > 0$.

The above assumptions ensures that that the linear programs that we will define below have finite optimal solutions. For a linear production process (A, b, c), and for every $S \in 2^N$, the maximal profit that the agents in S can obtain by pooling their resources is given by

$$v^{(A,b,c)}(S) := \max \left\{ c^T x : Ax \leq b(S),\ x \in \mathbf{R}^P_+ \right\}. \tag{5.2}$$

We will refer to the LP-problem given by (5.2) as $LP(A, b, c, S)$, or, if this is unambiguous, just $LP(S)$. From the Duality Theorem of Linear Programming follows that we can also compute the value of $LP(S)$ from

$$v^{(A,b,c)}(S) = \min \left\{ u^T b(S) : A^T u \geq c,\ u \in \mathbf{R}^R_+ \right\}. \tag{5.3}$$

For every linear production process (A, b, c) we define a *linear production game* $(N, v^{(A,b,c)})$, where N is the set of *players*, and $v^{(A,b,c)} : 2^N \to \mathbf{R}$ is the *characteristic function*. We will mostly skip the superscript and just write v for the characteristic function.

Several variations on linear production games, with respect to how the function b is defined, exist in the literature. Owen (1975) studies the situation where the resources are controlled by individual players, where b_{ik} denotes the amount of resource i controlled by player k. Owen assumes that a group of players can pool their resources by simply adding the individual amounts, i.e., $b_i(S) = \sum_{k \in S} b_{ik}$. In this case, an allocation in the core can be deduced

[1] If nothing else is stated, a vector is assumed to consist of one column.

[2] Let 2^N denote the set of all subsets of N.

from an optimal solution to the dual of $LP(N)$. If u is such an optimal dual solution then y is in the core of v, where $y_k := \sum_{i\in R} b_{ik}u_i$ for every $k \in N$. Gellekom et al. (1999) provide alternative characterizations of this allocation rule.

Granot (1986) generalizes this model, and studies the core of the linear production game $(N, v^{(A,b,c)})$ by looking at the *resource games* (N, b_i), $i \in R$. If the cores of all the resource games are nonempty, then the core of $(N, v^{(A,b,c)})$ is also nonempty. Moreover, if t^i is a core allocation for the resource game (N, b_i) for every $i \in R$, and u is an optimal dual solution to $LP(N)$, then a core allocation for the game $(N, v^{(A,b,c)})$ is given by the vector y, where the amount allocated to player k is $y_k := \sum_{i\in R} t^i_k u_i$.

Curiel et al. (1989) assumes that each resource $i \in R$ is divided into d_i portions. The amount of resource i belonging to portion q, $1 \leq q \leq d_i$, is b_i^q. Portion q of resource i is controlled by a committee $R \subseteq N$, meaning that a coalition $S \subseteq N$ can only use this portion if it contains R. Formally, this is modeled using a simple game[3] (N, w_i^q), where $w_i^q(S) = 1$ only if $R \subseteq S$. The amount of resource $i \in R$ controlled by coalition S is given by $b_i(S) := \sum_{q=1}^{d_i} b_i^q w_i^q(S)$. Curiel et al. show that the core of a linear production game is nonempty if all the games w_i^q, where $i \in R$ and $q \in \{1, \ldots, d_i\}$, have nonempty cores. Moreover, if z_i^q is in the core of the game (N, w_i^q) for every $i \in R$ and $q \in \{1, \ldots, d_i\}$, and if u is an optimal dual solution to $LP(N)$, then y is a core allocatin for $(N, v^{(A,b,c)})$, where $y_k := \sum_{i\in R} u_i \sum_{q=1}^{d_i} b_i^q (z_i^q)_k$ for every $k \in N$.

5.3 Aggregation of columns and rows

Reducing the size of (each of) the linear programs that must be solved in order to compute v can be done by aggregating over columns or rows (or both), as in Zipkin (1980b) and Zipkin (1980a), respectively.

In Zipkin (1980b), column aggregation is performed by specifying a partition of the set of columns. The columns belonging to each partition member are combined using a pre-specified weight vector. After the aggregated problem

[3]A game (N, g) is *simple* if $g(N) = 1$ and $g(S) \in \{0, 1\}$ for every $S \subseteq N$.

has been solved, a feasible solution to the original problem can be obtained by disaggregating using the same weight vectors. Our approach is a generalization[4] of that of Zipkin, and the aggregation is performed by multiplying A and c with the matrix $G \in \mathbb{R}_+^{P \times \bar{P}}$, where $\bar{P}$ is the set of "products" of the resulting linear production process $(AG, b, G^T c)$. Our purpose is to reduce the size of the LP-problems to be solved when computing the values of the linear production game, so in most cases we will have $\bar{p} < p$. The values of the resulting linear production game, which we label v^G, is given by, for every $S \subseteq N$,

$$v^G(S) := v^{(AG,b,G^T c)}(S) = \max\left\{c^T GX : AGX \leq b(S),\ X \in \mathbb{R}_+^{\bar{P}}\right\}. \quad (5.4)$$

The linear program to be solved by coalition S will be denoted $LP^G(S) = LP(AG, b, G^T c, S)$. In order to distinguish between the solutions of the original and the aggregated LP-problem, we will use uppercase letters to denote solutions to the latter problem. In order to illustrate how v^G is constructed, we provide an example.

Example 5.3.1 [Figures 5.1 and 5.2] There are four products ($p = 4$) and two resources ($r = 2$), and the production technology and the profits that can be made are given by

$$A = \begin{bmatrix} 2 & 1 & 3 & 1 \\ 1 & 2 & 2 & 1 \end{bmatrix}, \text{ and } c^T = \begin{bmatrix} 6 & 6 & 8 & 5 \end{bmatrix}.$$

The resources are controlled by three players ($n = 3$), and, as in Owen (1975), we assume that $b(S) := Be_S^N$ for every $S \subseteq N$, where

$$B = \begin{bmatrix} 9 & 0 & 6 \\ 1 & 8 & 3 \end{bmatrix}.$$

[4]In Zipkin (1980b), column aggregation is performed by specifying a partition $\sigma = \{P_k : k = 1, \ldots, K\}$ of P, and a weight vector g^k for each member of this partition. To illustrate how our approach relates to that of Zipkin, consider an example with four products, where $\sigma = \{\{1,2\},\{3,4\}\}$, and where $g_1^1 = g_2^1 = g_1^2 = g_2^2 = 0.5$. In our case this corresponds to the matrix

$$G = \begin{bmatrix} 0.5 & 0 \\ 0.5 & 0 \\ 0 & 0.5 \\ 0 & 0.5 \end{bmatrix}$$

Note that our approach is more general than that of Zipkin, in that aggregation is done with respect to *coverings* of the set of columns, since each row of G can have more than one nonzero element (see Section 4 of Zipkin (1980b)).

The value of coalition S is computed as

$$\begin{aligned} v(S) = \quad \max \quad & 6x_1 + 6x_2 + 8x_3 + 5x_4 \\ \text{s.t.} \quad & 2x_1 + 1x_2 + 3x_3 + 1x_4 + s_1 = b_1(S) \\ & 1x_1 + 2x_2 + 2x_3 + 1x_4 + s_2 = b_2(S) \\ & x_j \geq 0 \text{ for } j = 1,2,3,4 \\ & s_i \geq 0 \text{ for } i = 1,2 \end{aligned}$$

and the (unique) optimal solutions of the primal problems are shown in Figure 5.1.

S			x_1	x_2	x_3	x_4	s_1	s_2	$v(S)$	$v^G(S)$	$v^{G'}(S)$
1			1	0	0	0	7	0	6	5.25	6
	2		0	0	0	0	0	8	0	0	0
		3	3	0	0	0	0	0	18	15.75	18
1	2		0	0	0	9	0	0	45	37.8	45
1		3	4	0	0	0	7	0	24	21	24
	2	3	0	5	0	1	0	0	35	25.2	30
1	2	3	3	0	0	9	0	0	63	63	63

Figure 5.1: Optimal primal solutions for Example 5.3.1

Suppose now that we combine the columns of A using one of the solutions shown in Figure 5.1. Choosing the solution corresponding to the grand coalition, i.e.,

$$G = \begin{bmatrix} 3 \\ 0 \\ 0 \\ 9 \end{bmatrix},$$

gives the new linear production process $(AG, b, G^T c)$, where

$$AG = \begin{bmatrix} 15 \\ 12 \end{bmatrix} \text{ and } c^T G = \begin{bmatrix} 63 \end{bmatrix}.$$

Since the aggregated game has a single column, its value for a particular coalition can be computed by solving a continuous knapsack problem, e.g., for the grand coalition the value is

$$\begin{aligned} v^G(N) &= \max\{63X : 15X \leq 15,\ 12X \leq 12,\ X \in \mathbf{R}^1_+\} \\ &= 63 \times \min\left\{\frac{15}{15}, \frac{12}{12}\right\} = 63 = v(N). \end{aligned}$$

Not surprisingly, for the grand coalition, from which we obtained the aggregation weights, the game v^G coincides with v. For the other coalitions, having smaller amounts of resources than N, the value of the aggregated game is obtained by scaling down the value of the grand coalition. E.g., for coalition $\{1,3\}$,

$$\begin{aligned} v^G(1,3) &= \max\{63X : 15X \le 15,\ 12X \le 4,\ X \in \mathbf{R}^1_+\} \\ &= 63 \times \min\left\{\frac{15}{15}, \frac{4}{12}\right\} = 21 < v(1,3) = 24. \end{aligned}$$

We note that for all coalitions, the game v^G forms a lower bound for v.

Had we instead chosen the weight matrix

$$G' = \begin{bmatrix} 0 & 4 \\ 0 & 0 \\ 0 & 0 \\ 9 & 0 \end{bmatrix},$$

i.e., the columns of G' correspond to the optimal solutions of $LP(1,2)$ and $LP(1,3)$, the game $v^{G'}$, also shown in Figure 5.1, would result. The games v and $v^{G'}$ coincide for all but one coalition, namely $\{2,3\}$. An interesting point is that coincidence occurs even for coalitions for which we did not include the optimal solution in G'. We will show, in Proposition 5.3.2(iii) that coincidence will occur for a coalition S if and only if the optimal solution for $LP(S)$ can be obtained as a linear combination of the columns of G'. In the example, the optimal solution for the grand coalition can be obtained[5]

[5] In fact, the two problems $LP(N)$ and $LP(1,2)$ have the same optimal *basis*, hence the solution of both problems could have been obtained using the corresponding basis matrix. For any coalition S we can write the primal of $LP(S)$ as

$$\begin{aligned} v(S) = \quad & \max \quad c^T x \\ & \text{s.t} \quad Ax + Is = b(S) \\ & \qquad\; x \ge 0,\ s \ge 0, \end{aligned} \tag{5.5}$$

where s is a vector of slack variables. Letting

$$d^T := [c^T\ 0^T],\ y := \begin{bmatrix} x \\ s \end{bmatrix},\ \text{and } C := [A\ I],$$

we can rewrite (5.5) as

$$\begin{aligned} & \max \quad d^T y \\ & \text{s.t} \quad Cy = b(S) \\ & \qquad\; y \ge 0 \end{aligned} \tag{5.6}$$

by combining the solutions for $\{1,2\}$ and $\{1,3\}$ as

$$[\ 3\ \ 0\ \ 0\ \ 9\ ] = \frac{3}{4}[\ 4\ \ 0\ \ 0\ \ 0\ ] + 1\cdot[\ 0\ \ 0\ \ 0\ \ 9\ ],$$

hence we will have $v^{G'}(N) = v(N)$. The weights in this expression correspond to the optimal primal solution of $LP^{G'}(N)$, i.e., $X_1^* = \frac{3}{4}$ and $X_2^* = 1$. ◁

Proposition 5.3.2 *Let (A,b,c) be a linear production process, and $G \in \mathbb{R}_+^{P\times\bar{P}}$. Then the following statements are true:*

(i) (AG, b, G^Tc) is a linear production process.

(ii) $v^G(S) \leq v(S)$ for every $S \subseteq N$.

(iii) $v^G(S) = v(S)$ if and only if there exists $X \in \mathbb{R}^{\bar{P}}$ such that GX is an optimal primal solution of $LP(S)$.

Proof. (i) Since A and G have non-negative elements, the elements of AG must also be non-negative. Also, if $(G^Tc)_j = \sum_{k\in P} c_k g_{kj} > 0$ for some $j \in \bar{P}$, then there must exist some $k \in P$ such that $c_k > 0$ and $g_{kj} > 0$. Then, since $(A,b,c) \in \mathcal{L}$, there must exist some $i \in R$ such that $a_{ik} > 0$, and hence $(AG)_{ij} = \sum_{l\in P} a_{il}g_{lj} \geq a_{ik}g_{kj} > 0$.

(ii) For $S \subseteq N$ and an optimal solution X to the primal of $LP^G(S)$, we have $AGX \leq b(S)$, implying that GX is a feasible solution to the primal of $LP(S)$, hence we must have $v(S) \geq c^TGX = v^G(S)$.

The optimal basis matrix $\mathbf{B} \in \mathbb{R}^{R\times R}$, not to be confused with the matrix describing ownership of the resources, determines the solutions of the primal and dual, respectively, as

$$y_{\mathbf{B}}^* = \mathbf{B}^{-1}b(S) \text{ and } u^* = (d_{\mathbf{B}}^T\mathbf{B}^{-1})^T.$$

Hence, if $\mathbf{B}$ is an optimal basis also for some other coalition $R \neq S$, then $\mathbf{B}^{-1}b(R)$ is an optimal primal solution to $LP(R)$.

In the example, an optimal basis for coalitions N and $\{1,2\}$ corresponds to columns 1 and 4 of the matrix A, i.e., the basis matrix

$$\mathbf{B} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \text{ and its inverse } \mathbf{B}^{-1} = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix}.$$

The optimal solution for $LP(N)$ and $LP(1,2)$ are, respectively,

$$\mathbf{B}^{-1}b(N) = \mathbf{B}^{-1}\begin{bmatrix} 15 \\ 12 \end{bmatrix} = \begin{bmatrix} 3 \\ 9 \end{bmatrix} \text{ and } \mathbf{B}^{-1}b(1,2) = \mathbf{B}^{-1}\begin{bmatrix} 9 \\ 9 \end{bmatrix} = \begin{bmatrix} 0 \\ 9 \end{bmatrix}.$$

(iii) If GX is optimal in $LP(S)$, then

$$v(S) = c^T GX \leq v^G(S) \leq v(S) \Rightarrow v^G(S) = v(S).$$

The optimality of GX in $LP(S)$ implies $AGX \leq b(S)$, i.e., X is feasible in the primal of $LP^G(S)$, hence the first inequality. The second inequality follows from (ii).

Suppose $v^G(S) = c^T GX = v(S)$, where $X \in \mathbf{R}^{\bar{P}}$ is an optimal primal solution to $LP^G(S)$. Then clearly, $GX \in \mathbf{R}^P$ is feasible in $LP(S)$, since $GX \geq 0$ and $AGX \leq b(S)$. Then, since $v(S) = c^T GX$, the solution GX must be optimal in $LP(S)$. □

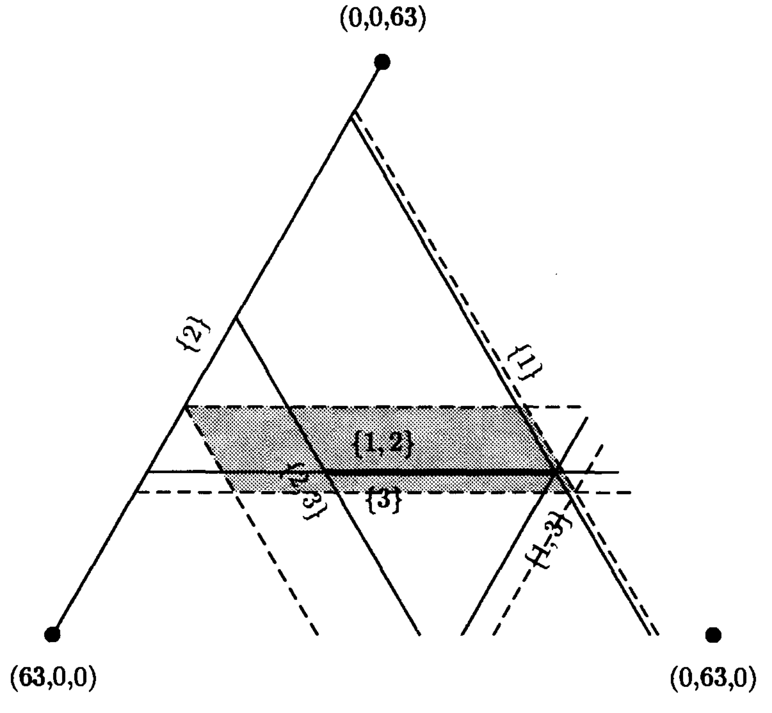


Figure 5.2: Core of v and v^G in Example 5.3.1

From Proposition 5.3.2(iii), we know that by including in G an optimal solution for the grand coalition, we can make $v^G(N) = v(N)$. Also, since $v^G \leq v$, by Proposition 5.3.2(ii), the core of v^G will contain the core of v. This is illustrated by Figure 5.2, where the solid lines represent the game v, and the dashed lines the game v^G.

We may also aggregate over the rows (resource constraints) of the LP-problem, as in Zipkin (1980a). Let $\bar{R}$ be the set of "resources" in the aggregated problem. Then, take some $H \in \mathbf{R}_{+}^{\bar{R}\times R}$ and define, for every $S \subseteq N$,

$$\begin{aligned} v^H(S) &:= v^{(HA,b^H,c)}(S) = \max\left\{c^T x : HAx \leq b^H(S),\ x \in \mathbf{R}_{+}^{P}\right\} \\ &= \min\left\{U^T Hb(S) : U^T HA \geq c^T,\ U \in \mathbf{R}_{+}^{\bar{R}}\right\}, \end{aligned}$$

where $b^H(S) := Hb(S)$ for every $S \subseteq N$. The linear program to be solved by coalition S will be denoted $LP^H(S) = LP(HA, b^H, c, S)$.

Example 5.3.3 [Figures 5.3 and 5.4] There are two products ($p = 2$) and four resources ($r = 4$), and the production technology and the profits that can be made are given by

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \\ 3 & 2 \\ 1 & 1 \end{bmatrix}, \text{ and } c^T = \begin{bmatrix} 6 & 6 \end{bmatrix}.$$

The resources are controlled by three players ($n = 3$), and $b(S) = Be_S^N$ for every $S \subseteq N$, where

$$B = \begin{bmatrix} 9 & 0 & 3 \\ 1 & 8 & 3 \\ 3 & 4 & 7 \\ 3 & 3 & 3 \end{bmatrix}.$$

The value of coalition S can be obtained as

$$\begin{aligned} v(S) = \quad \min \quad & u_1 b_1(S) + u_2 b_2(S) + u_3 b_3(S) + u_4 b_4(S) \\ \text{s.t.} \quad & 2u_2 + 1u_2 + 3u_3 + 1u_4 - s_1 = 6 \\ & 1u_1 + 2u_2 + 2u_3 + 1u_4 - s_2 = 6 \\ & u_i \geq 0 \text{ for } i = 1,2,3,4 \\ & s_j \geq 0 \text{ for } j = 1,2 \end{aligned}$$

and the optimal solutions of the dual problems are shown in Figure 5.3.

The weight matrix could e.g. be constructed from the dual solution for the grand coalition, i.e.,

$$H = \begin{bmatrix} 0 & 1.5 & 1.5 & 0 \end{bmatrix},$$

S			u_1	u_2	u_3	u_4	s_1	s_2	$v(S)$	$v^H(S)$	$v^{H'}(S)$	$v^{H''}(S)$
1			0	1.5	1.5	0	0	0	6	6	6	6
			0	6	0	0	0	6				
	2		6	0	0	0	6	0	0	18	12	0
		3	2	2	0	0	0	0	12	15	15	15
1	2		0	0	3	0	3	0	21	24	21	24
1		3	0	1.5	1.5	0	0	0	21	21	21	21
	2	3	6	0	0	0	6	0	18	33	33	18
1	2	3	0	1.5	1.5	0	0	0	39	39	39	39

Figure 5.3: Optimal dual solutions for Example 5.3.3

which produces the new linear production process[6] (HA, Hb, c), where

$$HA = \begin{bmatrix} 6 & 6 \end{bmatrix} \text{ and } HB = \begin{bmatrix} 6 & 18 & 15 \end{bmatrix}.$$

The dual of $LP^H(N)$ can now easily be solved as a continuous knapsack problem

$$\begin{aligned} v^H(N) &= \min\{39U : 6U \geq 6,\ 6U \geq 6,\ U \in \mathbf{R}^1_+\} \\ &= 39 \times \max\left\{\frac{6}{6}, \frac{6}{6}\right\} = 39 = v(N). \end{aligned}$$

Again, as for column-aggregation, the value of v^H for the grand coalition, which we used to generate H, coincides with the value of the original game. For other coalitions we get an upper bound on v, e.g.,

$$\begin{aligned} v^H(2) &= \min\{18U : 6U \geq 6,\ 6U \geq 6,\ U \in \mathbf{R}^1_+\} \\ &= 18 \times \max\left\{\frac{6}{6}, \frac{6}{6}\right\} = 18 \geq v(2). \end{aligned}$$

Note also that, since the dual constraints are the same for all coalitions, and they all have positive amounts of the single resource, $U^* = 1$ will be the optimal solution for all of them, and we have the additive structure given by $v^H(S) = \sum_{k \in S}(HB)_k$, i.e., the value for a coalition S is given by the total value of the resources owned by S, where the value is computed using the price vector included in H.

[6]It is not obvious that the aggregation actually yields a linear production process, since we may use H such that for a product j for which $c_j > 0$, we have $(HA)_{ij} = 0$ for all $i \in \bar{R}$.

A slightly better bound is obtained by using

$$H' = \begin{bmatrix} 0 & 6 & 0 & 0 \\ 0 & 0 & 3 & 0 \end{bmatrix},$$

i.e., we use the dual solutions corresponding to the coalitions $\{1\}$ and $\{1,2\}$. Note that we have $v(N) = v^{H'}(N)$, even though the optimal dual solution for the grand coalition is not included in H'. However, as we shall prove in Proposition 5.3.4(iii) below, coincidence follows from the fact that the optimal dual solution of $LP(N)$ can be written as a linear combination of the two row vectors of H', i.e.,

$$\begin{bmatrix} 0 & 1.5 & 1.5 & 0 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 0 & 6 & 0 & 0 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} 0 & 0 & 3 & 0 \end{bmatrix}.$$

The weights correspond to the optimal dual solution of $LP^{H'}(N)$, i.e., $U_1^* = 1/4$ and $U_2^* = 1/2$. ◁

Proposition 5.3.4 *Let (A, b, c) be a linear production process, and $H \in \mathbf{R}_+^{\bar{R} \times R}$.*

(i) If (HA, b^H, c) is a linear production process, then $v^H(S) \geq v(S)$ for every $S \subseteq N$.

(ii) If, for some $S \subseteq N$, there exists $U \in \mathbf{R}^{\bar{R}}$ such that $H^T U$ is optimal in the dual of $LP(S)$, then (HA, b^H, c) is a linear production process.

(iii) $v^H(S) = v(S)$ if and only if there exists $U \in \mathbf{R}^{\bar{R}}$ such that $H^T U$ is optimal in the dual of $LP(S)$.

Proof. (i) Take $S \subseteq N$ and an optimal solution U to the dual of $LP^H(S)$. Then, since the optimality of U implies $U^T HA \geq c^T$, $U^T H$ must be feasible in the dual of $LP(S)$, which implies $v^H(S) = U^T Hb(S) \geq v(S)$.

(ii) Since the elements of A and H, as well as the values returned by the function b, are non-negative, this must also be the case for the elements of HA, as well as the values returned by b^H. Also, since $H^T U$ is optimal in the dual of $LP(S)$, we must have $\sum_{i \in \bar{R}} U_i (HA)_{ij} \geq c_j$ for all $j \in P$. So if $c_j > 0$, there must exist some $i \in \bar{R}$ such that $(HA)_{ij} > 0$.

(iii) If $H^T U$ is optimal in the dual of $LP(S)$, then

$$v(S) = U^T Hb(S) \geq v^H(S) \geq v(S) \Rightarrow v^H(S) = v(S).$$

The optimality of $H^T U$ implies $U^T HA \geq c^T$ and $H^T U \geq 0$, hence U must be feasible in the dual of $LP^H(S)$, which implies the first inequality. The second inequality follows from (i).

Suppose $v^H(S) = U^T Hb(S) = v(S)$, where $U \in \mathbf{R}^{\bar{R}}$ is optimal in the dual of $LP^H(S)$. Then $U^T H \geq 0$ and $U^T HA \geq c^T$ implies that $H^T U$ is feasible in the dual of $LP(S)$, and optimality follows from $v(S) = U^T Hb(S)$. □

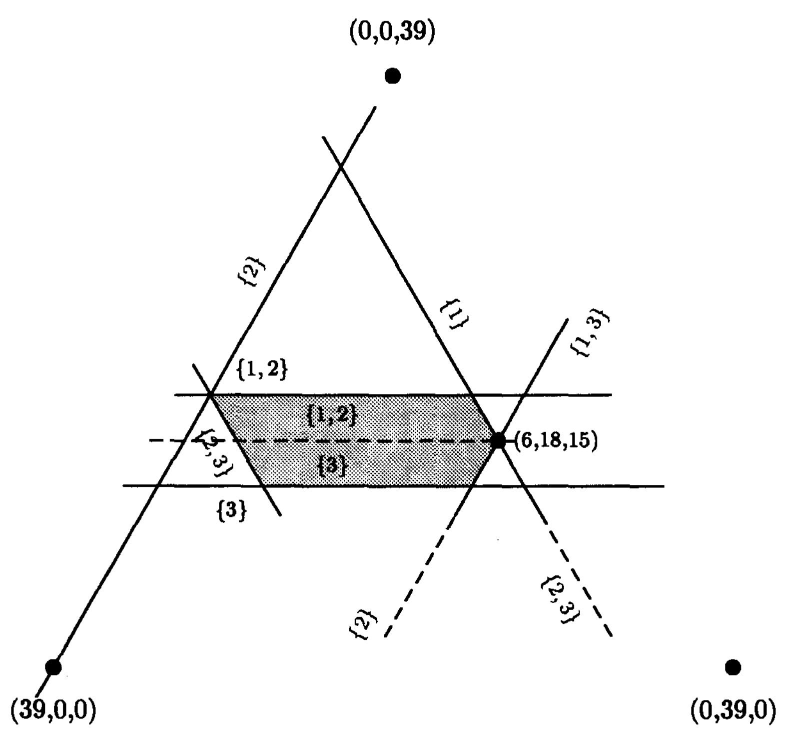


Figure 5.4: Core of v and v^H in Example 5.3.3

The cores of v and v^H in Example 5.3.3 are illustrated in Figure 5.4, where the solid (dashed) lines are hyperplanes corresponding to v (v^H). Since v^H is an upper bound for v, the core of v^H is contained in the core of v. Note that the core of v^H consists of the single point

$$\begin{bmatrix} 6 & 18 & 15 \end{bmatrix} = HB,$$

i.e., the allocation where the resources of the players are valued at the price vector corresponding to the dual solution of $LP(N)$.

Proposition 5.3.5 *Let u be an optimal dual solution to $LP(Q)$ for some $Q \subseteq N$ such that $v(Q) > 0$. Then, if $H = u^T$, we have $v^H(S) = u^T b(S)$ for every $S \subseteq N$.*

Proof. Since the aggregated problem contains only one row, i.e., $\bar{r} = 1$, the value of the game can be computed, for any S, as

$$\begin{aligned} v^H(S) &:= \min\left\{Uu^T b(S) : Uu^T A \geq c^T,\ U \in \mathbf{R}^1_+\right\} \\ &= u^T b(S) \cdot \max_{\substack{j \in P \\ u^T A^j > 0}} \frac{c_j}{u^T A^j} \end{aligned} \tag{5.7}$$

Note that the feasibility of u, for any $Q \subseteq N$, implies that $u^T A^j \geq c_j$ holds for every $j \in P$. Moreover, since $v(Q) > 0$, it must be optimal for the coalition Q to produce at least one product, hence we must have $u^T A^j = c_j$ for at least one $j \in P$. Then $v^H(S) = u^T b(S)$ follows from (5.7). □

5.4 Error bounds and ϵ-cores

The aggregated games presented in Section 5.3 enable us to analyze the original game with less computational effort. However, aggregation introduces a possible error, and the purpose of this section is to give an estimate of this error.

First, we need to make clear what we mean by "error". Since the core is one of the most widely used solution concepts for TU-games, it is natural to discuss error bounds relative to it. Suppose we use the game v^G as an approximation to the game v, where we have chosen G, according to Proposition 5.3.2, such that the core of v is contained in the core of v^G. Knowing that a pre-imputation z belongs to $C(v^G)$ thus does not guarantee that it also belongs to $C(v)$, hence there might exist some coalition S that receives less than its stand-alone value, i.e., $z(S) < v(S)$. We shall use as a "distance measure" the excess $e(v, S, z) = v(S) - z(S)$.

Suppose we know that

$$C(v^G) \subseteq C_\epsilon(v) = \{x \in I^*(v) : e(v,S,x) \leq \epsilon \ \forall S \subset N\}$$

for some ϵ. Since $z \in C_\epsilon(v)$, we know that no coalition has an excess of more than ϵ, hence no coalition receives less than $v(S) - \epsilon$. We would like to find the smallest ϵ-core containing $C(v^G)$, i.e., we need to solve

$$\min\{\epsilon : C(v^G) \subseteq C_\epsilon(v)\}. \tag{5.8}$$

Since making ϵ sufficiently high always makes $C_\epsilon(v)$ nonempty, (5.8) always has a solution.

Proposition 5.4.1 *Let (A, b, c) be a linear production process, and v be the corresponding linear production game. Let G be a matrix constructed according to Proposition 5.3.2 such that $C(v) \subseteq C(v^G)$, and let*

$$\epsilon := \max_{S \in 2^N \setminus \{N, \emptyset\}} \{v(S) - v^G(S)\}. \tag{5.9}$$

Then $C(v^G) \subseteq C_\epsilon(v)$.

Proof. If $z \in C(v^G)$, and $Q \subset N$, then

$$z(Q) \geq v^G(Q) \Rightarrow v(Q) - z(Q) \leq v(Q) - v^G(Q) \leq \epsilon.$$

□

In Example 5.3.1, (5.9) gives $\epsilon = 9.8 = v(2,3) - v^G(\{2,3\})$, and Figure 5.5 illustrates that $C_\epsilon(v) \supseteq C(v^G)$. The solid lines correspond to the sets

$$H_S^\epsilon(v) := \{z \in I^*(v) : z(S) = v(S) - \epsilon\},$$

and $C_\epsilon(v)$ is represented by the hatched area enclosed by these lines. The cores of v and v^G are given by the thick solid line and the shaded area, respectively.

Likewise, consider the games v and v^H, and suppose we know that

$$C(v^H) \supseteq C_\epsilon(v)$$

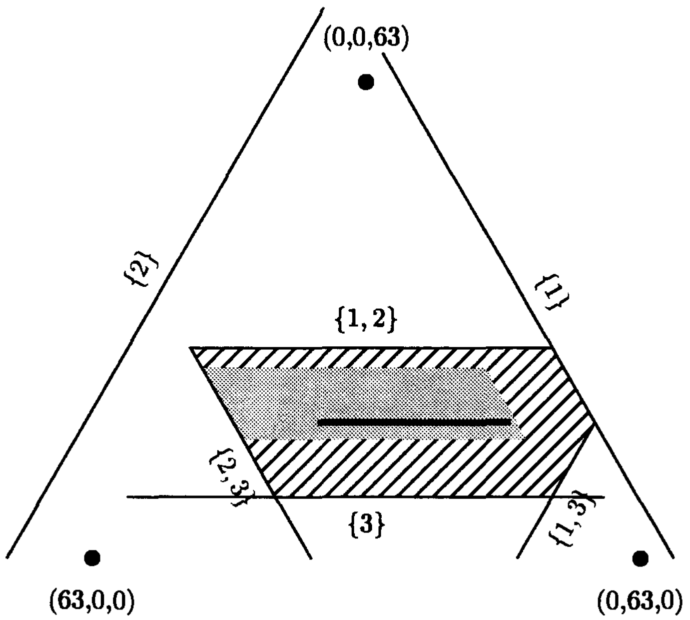


Figure 5.5: Cores of v and v^G, and the ϵ-core for $\epsilon = 12.4$, in Example 5.3.1

for some ϵ. By using v^H instead of v, we may exclude from consideration some elements of the core of v. However, we are certain to include all the points in $C_\epsilon(v)$, i.e., those with an excess less than or equal to ϵ. Of course, for $\epsilon = 0$, we exclude no core imputations, and in this case it follows that the cores of v and v^H coincide. We would like to find the largest ϵ-core that is contained in $C(v^H)$, i.e., we solve

$$\max\{\epsilon : C_\epsilon(v) \subseteq C(v^H)\}. \tag{5.10}$$

Whereas (5.8) always has a solution, (5.10) does not, since $C_\epsilon(v)$ is empty for small enough values of ϵ.

Proposition 5.4.2 *Let (A, b, c) be a linear production process, and v the corresponding linear production game. Let H be a matrix constructed according to Proposition 5.3.4 such that $C(v^H) \subseteq C(v)$, and let*

$$\epsilon := \min_{S \in 2^N \setminus \{N, \emptyset\}} \{v(S) - v^H(S)\}. \tag{5.11}$$

Then, if $C_\epsilon(v) \neq \emptyset$, we have $C_\epsilon(v) \subseteq C(v^H)$.

Proof. If $z \in C_\epsilon(v)$, and $Q \subset N$, then

$$v(Q) - z(Q) \leq \epsilon \leq v(Q) - v^H(Q) \Rightarrow z(Q) \geq v^H(Q).$$

□

In Example 5.3.3, if the weight matrix

$$H'' = \begin{bmatrix} 0 & 1.5 & 1.5 & 0 \\ 6 & 0 & 0 & 0 \end{bmatrix}$$

is used, then (5.11) gives $\epsilon = -3 = v(1,2) - v^{H'}(1,2) = v(3) - v^{H'}(3)$. In Figure 5.6 the cores of v and $v^{H''}$ are given by the shaded area and the thick solid line, respectively. The ϵ-core of v is represented by the white solid line, and we see that $C_\epsilon(v) \subseteq C(v^{H''})$. Note that we deliberately chose H'' here in order to make the ϵ-core of nonempty, given that ϵ satisfies (5.11).

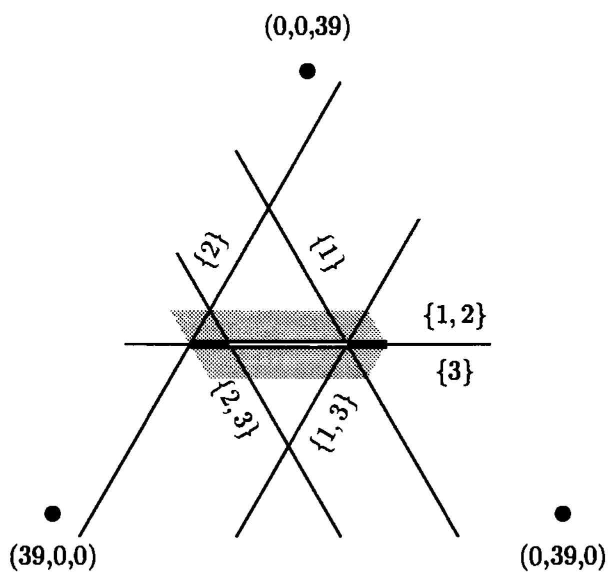


Figure 5.6: Cores of v and $v^{H''}$, and the ϵ-core, for Example 5.3.3

How can we find the error bounds given by (5.9) and (5.11) in practice? In the general case, there is not much to say about this issue, and we therefore choose to consider Owen's (1975) model. Here, player k controls b_{ik} units of resource i, where b_{ik} correspond to row i and column k of the matrix $B \in \mathbb{R}_+^{R \times N}$. The coalition S pool their resources by simply summing them, i.e., they control the resource vector $b(S) := Be_S^N$. In what follows, we will let (A, B, c) denote a linear production process, where the matrix B has replaced the function b.

Problem (5.9) may be formulated as

$$\epsilon^G := \max_{x,u,s} c^T x - u^T Bs \tag{5.12}$$
$$\text{subject to } Ax \leq Bs \tag{5.13}$$
$$u^T AG \geq c^T G \tag{5.14}$$
$$x \geq 0 \tag{5.15}$$
$$u \geq 0 \tag{5.16}$$
$$0 \leq s \leq 1 \tag{5.17}$$
$$s \text{ integer} \tag{5.18}$$

In a solution (x, u, s) to (5.12)-(5.18), $x \in \mathbf{R}^p$ is a solution to the primal of $LP(S)$, and $u \in \mathbf{R}^r$ is a solution to the dual of $LP^G(S)$. The coalition S corresponding to the solution is given by $S = \{k \in N : s_k = 1\}$. The objective function (5.12) maximizes the difference between the optimal values of the two problems. Primal feasibility of $LP(S)$ is ensured by (5.13) and (5.15), and dual feasibility of $LP^G(S)$ by (5.14) and (5.16). Problem (5.12)-(5.18) may be rewritten as:

$$\max \sum_{j\in P} c_j x_j - \sum_{i\in R}\sum_{k\in N} b_{ik} u_i s_k \tag{5.19}$$
$$\text{subject to } \sum_{j\in P} a_{ij} x_j \leq \sum_{k\in N} b_{ik} s_k \qquad \forall i \in R \tag{5.20}$$
$$\sum_{i\in R} u_i \sum_{j\in P} a_{ij} g_{j\ell} \geq \sum_{j\in P} c_j g_{\ell j} \qquad \forall \ell \in \bar{P} \tag{5.21}$$
$$x_j \geq 0 \qquad \forall j \in P \tag{5.22}$$
$$u_i \geq 0 \qquad \forall i \in R \tag{5.23}$$
$$0 \leq s_k \leq 1 \qquad \forall k \in N \tag{5.24}$$
$$s \text{ integer} \tag{5.25}$$

Finding a solution to (5.19)-(5.25) is made more difficult by the fact that (5.19) is non-concave, and because of the integrality condition (5.25). Methods to linearize such problems are given by Petersen (1971), Glover (1975), and Adams and Sherali (1990). In Petersen (1971) the product term $u_i s_k$ is replaced by the variable w_{ik}, and the following constraints are added:

$$u_i - u_i^+(1 - s_k) \leq w_{ik} \leq u_i^+ s_k \qquad \forall i \in R,\ k \in N \qquad (5.26)$$

$$w_{ik} \geq 0 \qquad \forall i \in R,\ k \in N \qquad (5.27)$$

$$w_{ik} \leq u_i \qquad \forall i \in R,\ k \in N \qquad (5.28)$$

The constant u_i^+ is an upper bound on the value of the variable u_i. If $s_k = 1$, then the first inequality of (5.26), together with (5.28) imply $u_i \leq w_{ik} \leq u_i$. In the case where $s_k = 0$, the second inequality of (5.26) together with (5.27) imply $0 \leq w_{ik} \leq 0$. Hence the equality $w_{ik} = u_i s_k$ always holds, and we may replace the objective function (5.19) by

$$\max \sum_{j\in P} c_j x_j - \sum_{i\in R}\sum_{k\in N} b_{ik} w_{ik}. \qquad (5.29)$$

Problem (5.19)-(5.25) is equivalent to the mixed-integer programming problem given by (5.20)-(5.29), hereafter referred to as MIP^G. Note that in an optimal solution, either we have $u_i = 0$ for all $i \in R$, or at least one of the constraints (5.21) is binding. Hence the upper bounds for the variable u can be set to

$$u_i^+ := \max\left\{ \max_{\substack{\ell\in \bar{P} \\ \sum_{j\in P} g_{j\ell} a_{ij} \neq 0}} \frac{\sum_{j\in P} g_{j\ell} c_j}{\sum_{j\in P} g_{j\ell} a_{ij}}, 0 \right\} \qquad \forall i \in R.$$

Problem (5.11) may be formulated as

$$\epsilon^H := \max_{x,u,s} c^T x - u^T B s \qquad (5.30)$$

$$\text{subject to } HAx \leq HBs \qquad (5.31)$$

$$u^T A \geq c^T \qquad (5.32)$$

$$x \geq 0 \qquad (5.33)$$

$$u \geq 0 \qquad (5.34)$$

$$0 \leq s \leq 1 \qquad (5.35)$$

$$s \text{ integer}, \qquad (5.36)$$

which, in a manner similar to that applied to (5.12)-(5.18), can be formulated as

$$
\begin{aligned}
\max\ & \sum_{j\in P} c_j x_j - \sum_{i\in R}\sum_{k\in N} b_{ik} w_{ik} && && (5.37)\\
\text{subject to}\ & \sum_{j\in P} x_j \sum_{i\in R} h_{\ell i} a_{ij} \le \sum_{k\in N} s_k \sum_{i\in R} h_{\ell i} b_{ik} && \forall \ell \in \bar{R} && (5.38)\\
& \sum_{i\in R} u_i a_{ij} \ge c_j && \forall j \in P && (5.39)\\
& x_j \ge 0 && \forall j \in P && (5.40)\\
& u_i \ge 0 && \forall i \in R && (5.41)\\
& 0 \le s_k \le 1 && \forall k \in N && (5.42)\\
& s \text{ integer} && && (5.43)\\
& u_i - u_i^+(1-s_k) \le w_{ik} \le u_i^+ s_k && \forall i \in R,\ k \in N && (5.44)\\
& w_{ik} \ge 0 && \forall i \in R,\ k \in N && (5.45)\\
& w_{ik} \le u_i && \forall i \in R,\ k \in N, && (5.46)
\end{aligned}
$$

where

$$
u_i^+ := \max\left\{ \max_{\substack{j\in P\\ a_{ij}\ne 0}} \frac{c_j}{a_{ij}}, 0 \right\} \qquad \forall i \in R.
$$

The mixed integer programming problem given by (5.37)-(5.46) will hereafter be referred to as MIP^H.

Example 5.4.3 [Figure 5.7] The data of this example is given by $n = 5$, $p = 5$, $r = 10$,

$$
A = \begin{bmatrix} 7 & 3 & 5 & 2 & 1\\ 6 & 9 & 9 & 5 & 10\\ 6 & 3 & 3 & 4 & 3\\ 9 & 5 & 4 & 2 & 1\\ 3 & 6 & 10 & 2 & 4\\ 4 & 5 & 1 & 3 & 8\\ 4 & 3 & 4 & 2 & 3\\ 7 & 9 & 1 & 1 & 7\\ 5 & 8 & 9 & 3 & 2\\ 2 & 6 & 3 & 10 & 2 \end{bmatrix},\ c = \begin{bmatrix} 53\\ 57\\ 49\\ 34\\ 41 \end{bmatrix},\ \text{and } B = \begin{bmatrix} 4 & 0 & 15 & 0 & 0\\ 0 & 22 & 18 & 0 & 0\\ 9 & 0 & 11 & 0 & 0\\ 0 & 17 & 0 & 5 & 0\\ 19 & 0 & 0 & 7 & 0\\ 0 & 13 & 0 & 9 & 0\\ 2 & 0 & 0 & 0 & 15\\ 0 & 22 & 0 & 0 & 4\\ 12 & 0 & 0 & 0 & 16\\ 0 & 23 & 0 & 0 & 0 \end{bmatrix}.
$$

The value of the grand coalition is $v(N) = 241.046$. We aggregate rows using the matrix

$$
H := \begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 19.0 & 0.0 & 0.0 & 0.0 \end{bmatrix},
$$

$\bar{r}$	ϵ^H	s^*					u^*									
1	226.6	0	1	1	1	0	0.0	0.0	0.0	0.0	17.7	0.0	0.0	0.0	0.0	0.0
2	220.5	0	1	1	0	1	0.0	0.0	19.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	123.7	0	1	0	1	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	26.5
4	88.5	1	0	1	1	1	13.2	0.0	0.0	0.0	0.0	3.5	0.0	0.0	0.0	0.0
5	69.0	1	1	0	0	1	0.0	0.0	0.0	3.2	2.9	3.2	0.0	0.0	0.0	1.2
6	46.3	1	1	1	0	1	9.6	3.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	21.8	1	1	0	1	1	0.0	3.3	0.0	3.0	0.0	0.0	0.0	0.3	0.5	1.0
8	0.0															

Figure 5.7: Solutions of MIP^H for Example 5.4.3

corresponding to the optimal dual solution of $LP(N)$. Solving MIP^H yields $\epsilon^H = 226.576$, corresponding to the coalition $\{2,3,4\}$. By adding the optimal dual solution to $LP(2,3,4)$, which is available as u^*, we obtain the new weight matrix

$$H := \begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 19.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.000 & 17.7 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix},$$

and by solving MIP^H again we obtain $\epsilon^H = 220.549$, corresponding to the coalition $\{2,3,5\}$. Continuing in this manner, new rows can be added to H until the value of ϵ^H is small enough. In Figure 5.7, the solutions of MIP^H, as new rows are added, are shown. After nine rows have been added, we have $\epsilon^H = 0$, implying that $v^H = v$. ◁

5.5 Numerical results

The purpose of this section is to investigate how the performance of the aggregation approach introduced in Section 5.3 depends on properties of the problem data. The analysis will be based on Owen's (1975) model, where the ownership of the resources is given by the matrix B. In our analysis, we will especially focus on the density of A, and the degree to which ownership is concentrated/dispersed, i.e., the structure of B.

A number of data sets with $n = 5$ were generated in a random manner. The nonzero elements of A were drawn from a uniform discrete distribution in the interval 1, ... ,10. The density of A, i.e., the probability that a particular element A_{ij} is nonzero, was set equal to the values 0.1, 0.4, 0.7, or 1.0. After A had been determined, we set $c_j := \sum_{i \in R} A_{ij}$ for all $j \in P$. The *total amount* of resource i was initially set to $b_{iN} := \sum_{j \in P} A_{ij}$, which was then distributed among the players according to the ownership profiles shown in Figure 5.8, where the x's indicate[7] which players are allocated positive amounts of each resource. For resource i denote these players by N_i. Let $\beta_{ik} \sim U(0,1)$ be a random number corresponding to resource i and player k. Then the amount of resource i given to player k is given by

$$\left\lceil b_{iN} \frac{\beta_{ik}}{\sum_{\ell \in N_i} \beta_{i\ell}} \right\rceil$$

Rows		Player 1	2	3	4	5
1	1-10	x	x	x	x	x
2	11-20	x	x	x	x	x
3	21-30	x	x	x	x	x
4	31-40	x	x	x	x	x
5	41-50	x	x	x	x	x
6	51-60	x	x	x	x	x
7	61-70	x	x	x	x	x
8	71-80	x	x	x	x	x
9	81-90	x	x	x	x	x
10	91-100	x	x	x	x	x

(a) Profile 1

Rows		Player 1	2	3	4	5
1	1-10	x		x		
2	11-20		x	x		
3	21-30	x		x		
4	31-40		x	x		
5	41-50	x			x	
6	51-60		x		x	
7	61-70	x			x	
8	71-80		x			x
9	81-90	x				x
10	91-100		x			x

(b) Profile 2

Rows		Player 1	2	3	4	5
1	1-10	x		x		
2	11-20		x	x		
3	21-30	x		x		
4	31-40		x	x		
5	41-50	x			x	
6	51-60		x		x	
7	61-70	x				
8	71-80		x			x
9	81-90	x				x
10	91-100		x			

(c) Profile 3

Rows		Player 1	2	3	4	5
1	1-10	x				
2	11-20	x				
3	21-30		x			
4	31-40		x			
5	41-50			x		
6	51-60			x		
7	61-70				x	
8	71-80				x	
9	81-90					x
10	91-100					x

(d) Profile 4

Figure 5.8: Ownership distribution profiles

[7]Row numbers from 1-10 refer to data sets with $r = 10$, and row numbers 1-100 to datasets with $r = 100$.

Profile 1 implies a relatively even distribution of the resources among the players, and may be seen as an extreme case. At the other extreme we find profile 4, where the entire amount of each resource is given to a single player. In the former case, the increased profits resulting from cooperation are modest, while in the latter cooperation is essential. Profiles 2 and 3 are located somewhere in between the two extremes. Note that according to these profiles, the resource bundles of player 1 and 2 are complements, and this is also the case for 3-5. Profile 3 differs from profile 2 in that player 4 does not own anything of resource 7 (61-70), and that player 4 does not own anything of resource 10 (91-100). Hence, profile 3 should, a priori, give greater benefits from cooperation than does profile 2.

Some properties/special cases regarding the data sets should be mentioned. First, note that if the ownership of resources is highly concentrated, and the density of A is high, we will have zero profits for many coalitions. In the extreme case of profile 4, where each resource has a single owner, we will have

$$v(S) = 0 \qquad\qquad \forall S \neq N, \tag{5.47}$$

if all entries of A are nonzero. Hence, positive profits can only be made if all the players pool their resources. In Figures 5.9 and 5.11, a "$\star$" after the problem name indicates that (5.47) is satisfied.

On the other hand, in the case where A is sparse, the game v will in many cases be ***additive***, i.e.,

$$v(S) + v(T) = v(S \cup T) \qquad \forall S, T \subset N \text{ s.t. } S \cap T = \emptyset. \tag{5.48}$$

To see why this is the case, consider the special case where every column of A has at most one nonzero entry. Then a unit of resource i should be used to produce the product that gives the highest profit contribution per unit that it consumes of resource i, i.e., the product, among those for which $A_{ij} > 0$, such that $\frac{c_j}{A_{ij}}$ is greatest. Assuming that there is at least on product such that $A_{ij} > 0$, the value of one unit of resource i is the constant

$$w_i := \max_{\substack{j \in P \\ A_{ij} > 0}} \frac{c_j}{A_{ij}},$$

and this constant is independent of who the owner of resource i is. Hence, the total profit that can be made by a coalition S can be found by simply summing the value of its resources, i.e.,

$$v(S) = \sum_{i \in R} w_i b_i(S) = \sum_{i \in R} \sum_{k \in S} w_i b_{ik},$$

which clearly satisfies (5.48). For additive games, the core consists of a single point. If u^* is an optimal solution to the dual of $LP(N)$, we know from Owen (1975) that the entire core is given by the point $(u^*)^T B$. In Figures 5.9 and 5.11, a "o" after the problem name indicates that (5.47) is satisfied.

For the data sets shown in Figure 5.9, where $p = 100$ and $r = 10$, column aggregation was performed. Initially, the weight matrix G consists of a single column corresponding to an optimal primal solution of $LP(N)$, and new columns were added by repeatedly solving MIP^G, where p^* is the number of columns needed in order to have $\epsilon^G = \max_{S \subset N,\ S \neq \emptyset} \{v(S) - v^G(S)\} = 0$, i.e., in order for the games v and v^G to coincide. Note that the number of coalitions is $2^n - 1 = 31$, which is an upper bound on the number of columns needed. In the table of Figure 5.9 is also reported ε_t, the value of $\epsilon^G / v(N)$ when t columns have been added to G.

The results in Figure 5.9 indicate that the effect on p^* of varying the density of A is ambiguous. If ownership is concentrated, as in profile 4, increasing density has a negative effect on p^*, whereas when ownership is dispersed, the effect is positive. Figure 5.10 can help explain this phenomenon. Let $\Omega(G)$ denote the set of coalitions that correspond to an optimal solution of MIP^G at some stage in the formation of G. For four of the data sets, we solved the aggregated problem $LP^G(S)$ for all $S \notin \Omega(G)$. Since $v = v^G$, we know from Proposition 5.3.2(iii) that for any $S \notin \Omega(G)$, it is possible to express an optimal solution to $LP(S)$ as a linear combination of the columns of G. Each row in Figure 5.10 corresponds to a coalition, and the coalitions have been sorted according to their size, as indicated by the numbers to the left of the diagrams. The $|$'s indicate nonzero entries of $(GX)^T$ for each $S \notin \Omega(G)$, where X is an optimal primal solution to $LP^G(S)$. Also, the $\bullet$'s indicate nonzero elements of each column of G, i.e., the primal solutions for the coalitions in $\Omega(G)$.

Profile	Problem		d	p^*	ε_1	ε_5	ε_{10}	ε_{20}	ε_{30}
1	P1D10A	∘	0.1	15	0.361	0.180	0.122	0.000	0.000
	P1D10B	∘	0.1	14	0.391	0.224	0.133	0.000	0.000
	P1D10C	∘	0.1	17	0.418	0.151	0.069	0.000	0.000
	P1D40A	∘	0.4	13	0.394	0.306	0.137	0.000	0.000
	P1D40B	∘	0.4	18	0.348	0.212	0.130	0.000	0.000
	P1D40C	∘	0.4	11	0.339	0.172	0.069	0.000	0.000
	P1D70A		0.7	16	0.237	0.154	0.122	0.000	0.000
	P1D70B		0.7	18	0.333	0.159	0.086	0.000	0.000
	P1D70C		0.7	21	0.446	0.197	0.134	0.009	0.000
	P1D100A		1.0	31	0.456	0.180	0.131	0.068	0.003
	P1D100B		1.0	30	0.367	0.199	0.104	0.042	0.000
	P1D100C		1.0	26	0.281	0.151	0.088	0.018	0.000
2	P2D10A	∘	0.1	23	0.780	0.593	0.463	0.207	0.000
	P2D10B	∘	0.1	25	0.747	0.620	0.481	0.219	0.000
	P2D10C	∘	0.1	23	0.630	0.602	0.439	0.213	0.000
	P2D40A		0.4	26	0.715	0.575	0.456	0.147	0.000
	P2D40B		0.4	26	0.627	0.574	0.435	0.127	0.000
	P2D40C		0.4	27	0.704	0.596	0.459	0.136	0.000
	P2D70A		0.7	25	0.752	0.573	0.294	0.039	0.000
	P2D70B		0.7	26	0.774	0.639	0.418	0.080	0.000
	P2D70C		0.7	24	0.679	0.512	0.275	0.041	0.000
	P2D100A		1.0	11	0.502	0.304	0.019	0.000	0.000
	P2D100B		1.0	7	0.606	0.107	0.000	0.000	0.000
	P2D100C		1.0	11	0.385	0.200	0.004	0.000	0.000
3	P3D10A	∘	0.1	25	0.713	0.639	0.500	0.282	0.000
	P3D10B	∘	0.1	25	0.868	0.607	0.494	0.202	0.000
	P3D10C	∘	0.1	25	0.735	0.627	0.475	0.265	0.000
	P3D40A		0.4	29	0.838	0.652	0.547	0.299	0.000
	P3D40B		0.4	29	0.850	0.636	0.497	0.258	0.000
	P3D40C		0.4	30	0.800	0.671	0.525	0.232	0.000
	P3D70A		0.7	21	0.796	0.540	0.347	0.101	0.000
	P3D70B		0.7	22	0.837	0.518	0.410	0.114	0.000
	P3D70C		0.7	21	0.707	0.483	0.271	0.039	0.000
	P3D100A		1.0	8	0.631	0.165	0.000	0.000	0.000
	P3D100B		1.0	5	0.586	0.000	0.000	0.000	0.000
	P3D100C		1.0	8	0.509	0.213	0.000	0.000	0.000
4	P4D10A		0.1	31	0.880	0.721	0.619	0.405	0.042
	P4D10B	∘	0.1	31	0.849	0.718	0.620	0.416	0.151
	P4D10C	∘	0.1	31	0.876	0.733	0.609	0.412	0.124
	P4D40A		0.4	30	0.812	0.792	0.596	0.360	0.000
	P4D40B		0.4	27	0.832	0.772	0.589	0.253	0.000
	P4D40C		0.4	27	0.815	0.787	0.590	0.327	0.000
	P4D70A		0.7	11	0.789	0.660	0.203	0.000	0.000
	P4D70B		0.7	12	0.813	0.689	0.276	0.000	0.000
	P4D70C		0.7	8	0.819	0.649	0.000	0.000	0.000
	P4D100A	⋆	1.0	1	0.000	0.000	0.000	0.000	0.000
	P4D100B	⋆	1.0	1	0.000	0.000	0.000	0.000	0.000
	P4D100C	⋆	1.0	1	0.000	0.000	0.000	0.000	0.000

Figure 5.9: Results for column aggregation, with $n = 5$, $p = 100$, and $r = 10$

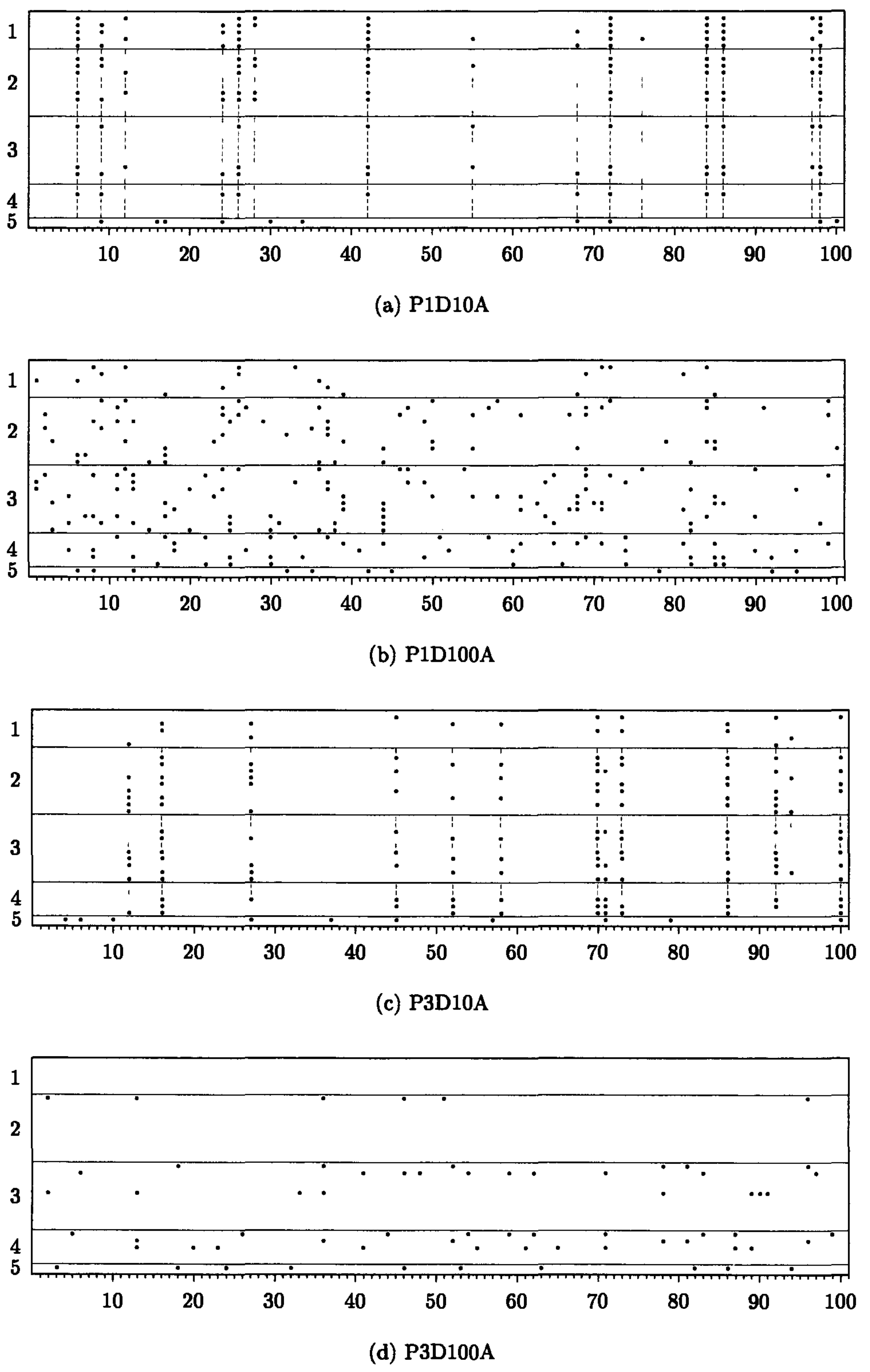


Figure 5.10: Nonzero elements of primal solutions

We see that increased density

(i) leads to more variation among the production plans of the various coalitions, and

(ii) makes it more difficult for small coalitions to produce anything at all, i.e., there are fewer nonzero entries for small coalitions.

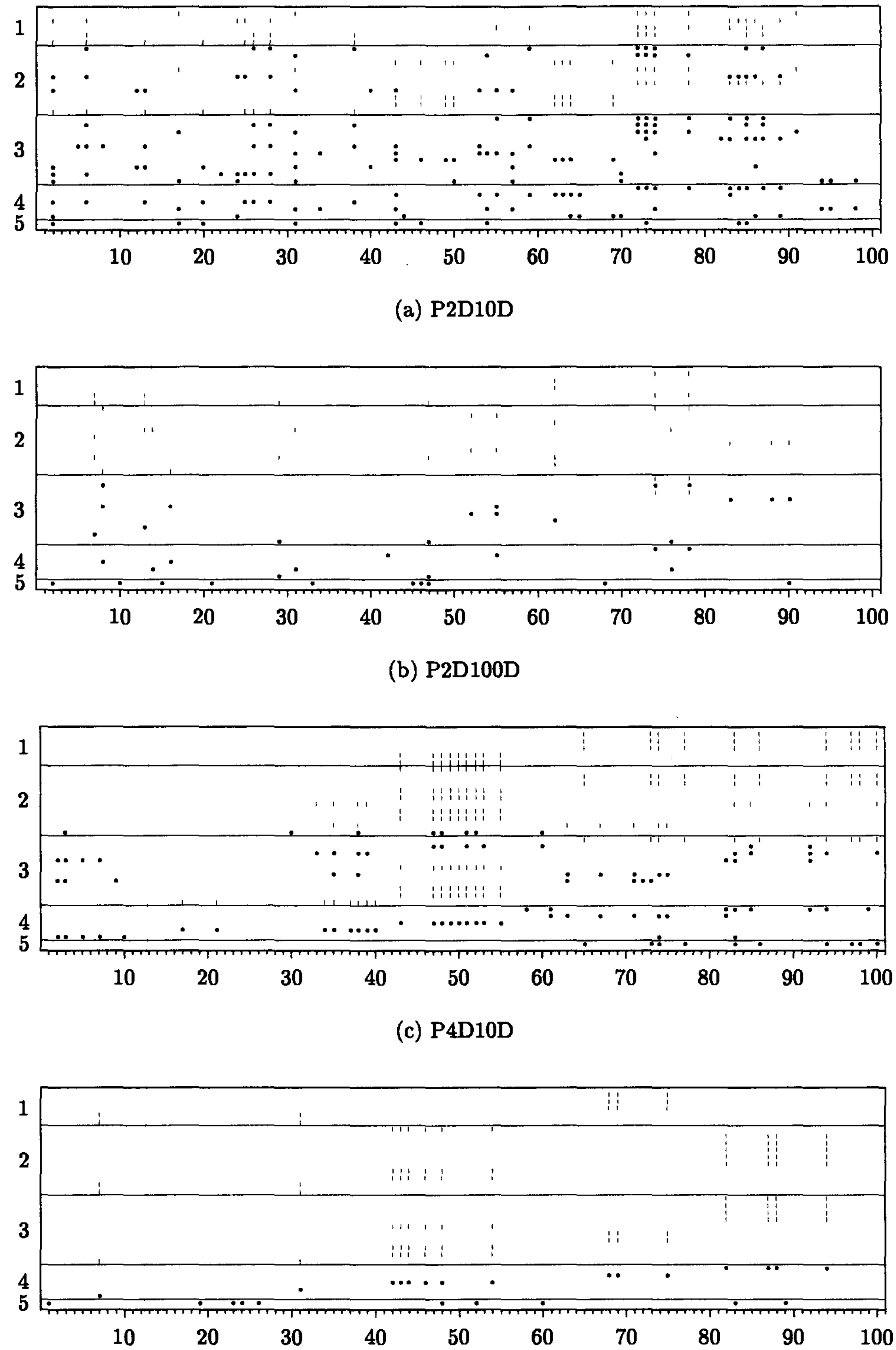
When ownership is dispersed, such as for profile 1, effect (i) is dominant. Proposition 5.3.2(iii) indicates that greater variation among the primal solutions of various coalitions makes the column aggregation approach less successful. When ownership is relatively concentrated, as for profile 3, effect (ii) dominates. If a coalition S cannot produce anything, we will have $v(S) = 0$, hence $0 \leq v^G(S) \leq v(S) \Rightarrow v^G(S) = v(S) = 0$ for any choice of the weight matrix $G \in \mathbf{R}_+^{P \times \bar{P}}$.

For the data sets of Figure 5.11, where $n = 5$, $p = 10$, and $r = 100$, row aggregation was performed. Initially, H consisted of one row corresponding to an optimal dual solution of $LP(N)$, and new rows were added by repeatedly solving MIP^H. The number r^* indicates the number of rows that had to be included in H in order to make $v = v^H$. We also report ε_t, the value of $\epsilon^H/v(N)$ when t rows have been added to H.

The results in Figure 5.11 indicate that increased density of A makes the row aggregation approach more successful, i.e., r^* decreases, except for profile 1, where r^* is close to or at the upper bound $2^n - 1$. In order to explain this phenomenon, the problems $LP^H(S)$ were solved for all $S \notin \Omega(H)$ for four of the datasets, where $\Omega(H)$ is the set of coalitions corresponding to the rows of H. In Figure 5.12, we indicate by $|'s$ the nonzero values of $U^T H$ for each coalition $S \notin \Omega(H)$, where U is an optimal dual solution to $LP^H(S)$. For all coalitions in $\Omega(H)$, the nonzero elements of the the corresponding row of H are indicated by •'s. We see that increased density of A, for the examples shown in Figure 5.12, has the effect of decreasing the number of nonzero entries. To see why this is the case, note that a relatively dense A-matrix makes decisions on different products/resources more interdependent. The number of bottlenecks, and hence the number of positive dual prices, will be fewer, as seen for dataset P2D100D and P4D100D. This makes it easier

Profile	Problem	Density	r^*	ε_1	ε_5	ε_{10}	ε_{20}	ε_{30}
	P1D10D	0.1	31	0.218	0.097	0.040	0.026	0.006
	P1D10E	0.1	31	0.153	0.088	0.050	0.020	0.001
	P1D10F	0.1	31	0.155	0.085	0.043	0.020	0.003
	P1D40D	0.4	31	0.282	0.167	0.081	0.036	0.011
	P1D40E	0.4	31	0.319	0.177	0.076	0.040	0.005
1	P1D40F	0.4	31	0.337	0.197	0.094	0.039	0.003
	P1D70D	0.7	31	0.303	0.186	0.083	0.047	0.017
	P1D70E	0.7	31	0.342	0.210	0.140	0.050	0.007
	P1D70F	0.7	31	0.382	0.202	0.141	0.046	0.009
	P1D100D	1.0	30	0.328	0.239	0.141	0.065	0.000
	P1D100E	1.0	31	0.317	0.208	0.118	0.046	0.002
	P1D100F	1.0	31	0.509	0.201	0.133	0.043	0.002
	P2D10D	0.1	20	0.646	0.404	0.156	0.000	0.000
	P2D10E	0.1	21	0.646	0.332	0.119	0.014	0.000
	P2D10F	0.1	17	0.707	0.484	0.205	0.000	0.000
	P2D40D	0.4	17	0.816	0.502	0.104	0.000	0.000
	P2D40E	0.4	17	0.865	0.443	0.065	0.000	0.000
2	P2D40F	0.4	17	0.734	0.457	0.311	0.000	0.000
	P2D70D	0.7	15	0.852	0.572	0.053	0.000	0.000
	P2D70E	0.7	15	0.867	0.581	0.063	0.000	0.000
	P2D70F	0.7	16	0.844	0.437	0.055	0.000	0.000
	P2D100D	1.0	14	0.927	0.492	0.010	0.000	0.000
	P2D100E	1.0	14	0.791	0.713	0.015	0.000	0.000
	P2D100F	1.0	12	0.942	0.397	0.002	0.000	0.000
	P3D10D	0.1	17	0.715	0.246	0.111	0.000	0.000
	P3D10E	0.1	18	0.639	0.403	0.135	0.000	0.000
	P3D10F	0.1	18	0.725	0.275	0.178	0.000	0.000
	P3D40D	0.4	11	0.743	0.540	0.017	0.000	0.000
	P3D40E	0.4	10	0.869	0.481	0.000	0.000	0.000
3	P3D40F	0.4	16	0.811	0.390	0.257	0.000	0.000
	P3D70D	0.7	9	0.880	0.464	0.000	0.000	0.000
	P3D70E	0.7	9	0.889	0.510	0.000	0.000	0.000
	P3D70F	0.7	11	0.889	0.492	0.045	0.000	0.000
	P3D100D	1.0	8	0.918	0.398	0.000	0.000	0.000
	P3D100E	1.0	9	0.959	0.589	0.000	0.000	0.000
	P3D100F	1.0	10	0.857	0.563	0.000	0.000	0.000
	P4D10D	0.1	12	1.000	0.337	0.060	0.000	0.000
	P4D10E	0.1	9	0.928	0.460	0.000	0.000	0.000
	P4D10F	0.1	10	0.935	0.388	0.000	0.000	0.000
	P4D40D ⋆	0.4	6	1.000	0.507	0.000	0.000	0.000
	P4D40E ⋆	0.4	6	0.836	0.732	0.000	0.000	0.000
4	P4D40F ⋆	0.4	6	0.859	0.730	0.000	0.000	0.000
	P4D70D ⋆	0.7	6	1.000	0.521	0.000	0.000	0.000
	P4D70E ⋆	0.7	6	0.950	0.527	0.000	0.000	0.000
	P4D70F ⋆	0.7	6	1.000	0.383	0.000	0.000	0.000
	P4D100D ⋆	1.0	6	0.996	0.211	0.000	0.000	0.000
	P4D100E ⋆	1.0	6	1.000	0.229	0.000	0.000	0.000
	P4D100F ⋆	1.0	6	0.989	0.620	0.000	0.000	0.000

Figure 5.11: Results for row aggregation, with $n = 5$, $p = 10$, and $r = 100$



(a) P2D10D

(b) P2D100D

(c) P4D10D

(d) P4D100D

Figure 5.12: Nonzero elements of dual solutions

to express the dual solutions of *all* coalitions as combinations of the dual solutions of a relatively small subset of the coalitions.

More concentrated ownership seems to work in the same direction as increased density of A, but we have no good explanation for this phenomenon at present.

5.6 Conclusion

We have shown how the dimensions of linear production games may be reduced by aggregating over columns or rows. In Section 5.3 we showed that by choosing weights corresponding to optimal solutions of the primal (dual) corresponding to particular coalitions, we can make the aggregated games coincide with the original games for those coalitions. This can be used to create a new game, easier to handle computationally, whose core form a superset (subset) of the original core. This introduces a possible error, and in Section 5.4 we provide a method, by solving a mixed integer programming problem, for quantifying this error in the special case where the players pool their resources by simply adding them. The solution of this problem can also be used to improve the bound on the original game, by adding a new column (row) to the weight matrix, and suggests a procedure by which the bound can successively be improved. In section 5.5 we test this procedure on a number of examples. The examples differ with respect to how concentrated the ownership of the resources are, and the density of the technology matrix A. The results indicate that for column aggregation, the aggregation approach is suitable for problems where ownership is relatively concentrated (dispersed) and A is dense (sparse). Row aggregation seems to be suitable for cases where ownership is relatively concentrated *and* A is dense.

We know from Owen (1975) that some core points can be obtained from the dual solution corresponding to the grand coalition. An interesting question is, if we have constructed H such that $v^H = v$, the dual solutions included in H can be given an interpretation in relation to the core of the original game.

Chapter 6

On the Computation of the Pre-Nucleolus

6.1 Introduction

This chapter has a double purpose. First, in Section 6.2, we describe Kopelowitz's (1967) procedure for computing the pre-nucleolus[1], in which an LP-problem is repeatedly solved, and where the excess values of some coalitions are fixed in every iteration. A modified version of this procedure will also be used in Chapter 7, where we compute a *restricted nucleolus*. Second, in Section 6.3, we discuss how the coalitions whose excess values are to be fixed in a particular iteration, corresponding to active constraints, can be identified using the optimal dual prices of the LP-problem. If we solve the LP-problems using a extreme-point method, we may not be able to identify all the active constraints, since the dual may have multiple optimal solutions. If, however, we use an interior-point method, we get all the active constraints. This point is illustrated in Section 6.4 via an example, and in Section 6.5 we show how the computational results change when the constraints of the LP-problem have not been specified in advance.

[1] See Section 1.3 for a definition.

6.2 Kopelowitz's procedure

The procedure[2] is based on repeatedly solving an LP-problem until a unique solution of the problem is found. The first problem, $P(1)$, seeks to maximize the smallest, with respect to the set of coalitions, of the excess values, i.e.,

$$\begin{aligned} &\max r \\ &\text{subject to } x(S) + r \leq c(S) && \forall S \in \Sigma^0 \\ &\qquad x(N) = c(N), \end{aligned}$$

where $\Sigma^0 := \{S \subset N : S \neq \emptyset\}$. Problem $P(1)$ has the optimal value r_1 and the set of optimal solutions X^1. If the solution is not unique, a new problem is formed where we fix the excess values of those coalitions corresponding to *active constraints*, i.e., the set $\Sigma_1 := \{S \in \Sigma^0 : x(S) = c(S) - r_k \; \forall x \in X^1\}$. The next problem to be solved, $P(2)$, is

$$\begin{aligned} &\max r \\ &\text{subject to } x(S) = c(S) - r_1 && \forall S \in \Sigma_1 \\ &\qquad x(S) + r \leq c(S) && \forall S \in \Sigma^1 \\ &\qquad x(N) = c(N), \end{aligned}$$

where $\Sigma^1 := \Sigma^0 \setminus \Sigma_1$. The optimal value of $P(2)$ is r_2, and the set of optimal solutions is X^2. Again we identify $\Sigma_2 := \{S \in \Sigma^1 : x(S) = c(S) - r_2 \; \forall x \in X^2\}$, and the procedure continues in this manner until the LP-problem has a unique solution. In iteration k, problem $P(k)$ is given by

$$\max r \tag{6.1}$$

$$\text{subject to } x(S) = c(S) - r_j \qquad \forall S \in \Sigma_j, j = 1, ..., k-1 \tag{6.2}$$

$$x(S) + r \leq c(S) \qquad \forall S \in \Sigma^{k-1} \tag{6.3}$$

$$x(N) = c(N), \tag{6.4}$$

and the entire procedure is summarized as Algorithm 6.2.1 below.

[2]The procedure was introduced by Kopelowitz (1967), and its properties are discussed by Maschler et al. (1979).

Algorithm 6.2.1
Begin

$\Sigma^0 := \{S \subset N : S \neq \emptyset, N\}$
$k := 0$
Repeat
$k := k + 1$
Solve $P(k)$ and store the optimal value as r_k
$X^k := \{x \in \mathbf{R}^n : x$ is an optimal solution to $P(k)\}$
$\Sigma_k := \{S \in \Sigma^{k-1} : x(S) = c(S) - r_k \ \forall x \in X^k\}$
$\Sigma^k := \Sigma^{k-1} \setminus \Sigma_k$
until $|X^k| = 1$

end

6.3 On identifying the active constraints

In order to identify the active constraints in Algorithm 6.2.1, i.e., the set Σ_k, we may use the dual solution. The dual to (6.1)-(6.4) is given by

$$\min \sum_{S \in \Sigma^{k-1} \cup \{N\}} \mu_S c(S) + \sum_{\substack{S \in \Sigma_j \\ j=1,\ldots,k-1}} \mu_S [c(S) - r_j] \tag{6.5}$$

$$\text{s.t.} \sum_{S \ni i} \mu_S = 0 \qquad \forall i \in N \tag{6.6}$$

$$\sum_{S \in \Sigma^{k-1}} \mu_S = 1 \tag{6.7}$$

$$\mu_S \geq 0 \qquad \forall S \in \Sigma^{k-1} \tag{6.8}$$

A solution (x, r, μ) is optimal if and only if it satifies (6.1)-(6.4), (6.5)-(6.8), as well as the *complementarity* condition

$$\mu_S[c(S) - x(S) - r] = 0 \qquad \forall S \in \Sigma^{k-1}. \tag{6.9}$$

Condition (6.9) implies[3] that if (x^*, r^*, μ^*) is optimal, then, for any $S \in \Sigma^{k-1}$,

$$\mu_S^* > 0 \Rightarrow c(S) - x(S) - r_k = 0 \qquad \forall x \in X^k.$$

[3]Suppose that the statement is not true, i.e., there exists some other optimal solution (x', r', μ') such that $c(S) - x'(S) - r' > 0$. Since the set of optimal solutions is convex, the solution $\frac{1}{2}(x^*, r^*, \mu^*) + \frac{1}{2}(x', r', \mu')$ must also be optimal. But this is a contradiction,

Note that a positive dual price is a sufficient, but not necessary, condition for a constraint to be active, i.e., we may have $\mu_S^* = 0$ even though the corresponding constraint is binding for all optimal solutions. Hence the set

$$\hat{\Sigma}_k := \{S \in \Sigma^{k-1} : \mu_S^* > 0\}$$

will, if there are multiple optima in the dual, be a strict subset of Σ_k.

By imposing the additional restriction that the optimal solution should be *strictly complementary*, i.e., it should satisfy

$$\mu_S + [c(S) - x(S) - r] > 0 \qquad \forall S \in \Sigma^{k-1}, \tag{6.10}$$

we avoid this problem. For, if the solution (x^*, r^*, μ^*) satisfies (6.10), then we must have, for any $S \in \Sigma^{k-1}$, that

$$\mu_S^* > 0 \Leftarrow c(S) - x(S) - r_k = 0 \qquad \forall x \in X^k.$$

Strictly complementary solutions can be obtained by interior-point methods.[4] The next section presents an example illustrating how the choice of solution methods for the LP-problems can influence the number of iterations needed in Algorithm 6.2.1.

6.4 An example – vehicle routing

The vehicle routing game was discussed by Göthe-Lundgren et al. (1996), and our example, illustrated by Figure 6.1, is identical to Example E2 in their article. There are n customers, each with a demand of d_i units, and the customers must be served from a central depot by one or more vehicles. The cost of making a trip from point i to point j is c_{ij}, where $0 \le i, j \le n$, and where point 0 is the location of the central depot. The total cost of the transportation plan is to be minimized, subject to the restriction that the entire demand of the customers should be met, and that the capacity of the vehicle(s), given by the number Q, should not be violated. Because of the capacity constraints, the optimal plan may involve more than one vehicle.

since

$$\frac{1}{2}\left([c(S) - x^*(S) - r^*] + [c(S) - x'(S) - r']\right) > 0 \text{ and } \frac{1}{2}\left(\mu_S^* + \mu'_S\right) > 0$$

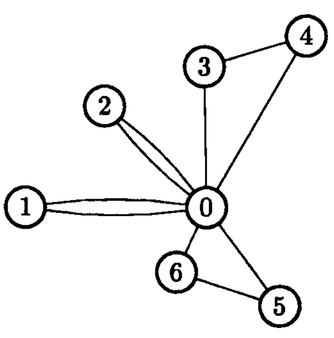
violates (6.9).

[4]See e.g. Wright (1997).

$$n = 6 \quad Q = 30 \quad d = \begin{bmatrix} 8 & 24 & 22 & 6 & 14 & 10 \end{bmatrix}$$

$$c = \begin{bmatrix} 0 & 24 & 19 & 20 & 27 & 16 & 12 \\ 24 & 0 & 17 & 31 & 44 & 36 & 23 \\ 19 & 17 & 0 & 16 & 29 & 35 & 25 \\ 20 & 31 & 16 & 0 & 15 & 34 & 28 \\ 27 & 44 & 29 & 15 & 0 & 40 & 37 \\ 16 & 36 & 35 & 34 & 40 & 0 & 13 \\ 12 & 23 & 25 & 28 & 37 & 13 & 0 \end{bmatrix}$$

(a) The data



(b) Optimal solution for the grand coalition

S	c(S)	S	c(S)	S	c(S)
{1}	48	{1,3}	75	{4,5 }	83
{2}	38	{1,4}	96	{4,6 }	76
{3}	40	{1,5}	76	{5,6 }	41
{4}	54	{1,6}	59	{1,4,5}	123
{5}	32	{2,4}	75	{1,4,6}	106
{6}	24	{3,4}	62	{4,5,6}	92

(c) Feasible coalitions/subroutes

Figure 6.1: Example E2 of Göthe-Lundgren et al. (1996)

The data of our example, which involves $n = 6$ customers, are shown in Figure 6.1(a), and the optimal plan for the grand coalition is shown in 6.1(b). The optimal route configuration has a total cost of

$$c(N) = 189 = c(1) + c(2) + c(3,4) + c(5,6) = 189,$$

and consists of four *feasible subroutes*, i.e., customer sets (coalitions) that can be served by a single vehicle. All the feasible subroutes/coalitions are listed in Figure 6.1(c). The vehicle routing problem may be seen as finding the combination of feasible subroutes such that each customer is covered by one subroute, i.e., solving the integer programming problem[5]

$$\min\left\{\sum_{t\in T} c_t y_t : \sum_{t\in T} y_t a_{it} = 1 \ \forall i \in N,\ y_t \in \{0,1\}\ \forall t \in T\right\}, \tag{6.11}$$

where T is the set of feasible subroutes, and c_t is the cost of subroute $t \in T$. The parameter a_{it} is equal to 1 if customer i is covered by route t, and 0 otherwise. The stand-alone cost of any coalition $S \subseteq N$ is the minimum cost of covering the customers belonging to S, i.e., the value

$$c(S) = \min\left\{\sum_{t\in T} c_t y_t : \sum_{t\in T} y_t a_{it} = 1 \ \forall i \in S,\ y_t \in \{0,1\}\ \forall t \in T\right\}. \tag{6.12}$$

With $n = 6$, there are $2^6 - 1 = 63$ nonempty coalitions, and the value of all these coalitions must be computed in advance. The pre-nucleolus is the vector[6]

$$\left(47\frac{1}{3}, 38, 26\frac{1}{3}, 35\frac{2}{3}, 29\frac{1}{3}, 12\frac{1}{3}\right),$$

and the computational details of Algorithm 6.2.1 are shown in Figure 6.2. The dual of all the LP-problems solved had multiple optima, and therefore an extreme-point method[7] identifies only a few active constraints in every iteration, and the algorithm terminates with a unique solution after 15 iterations. Using an interior-point method[8], on the other hand, we are able to identify *all* active constraints from one dual solution, and we use only 3 iterations.

[5]This is the formulation used by Gö the-Lundgren et al. (1996). For a discussion of solution methods for the vehicle routing problem, see Laporte (1998).

[6]This is not the same solution as in Göthe-Lundgren et al. (1996), where only the *feasible coalitions* were used. A sufficient condition for their procedure to yield the pre-nucleolus is that the core is nonempty, cf. Huberman (1980). See also Chardaire (2001).

[7]The dual Simplex-algorithm of CPLEX 7.0.

[8]The barrier algorithm of CPLEX 7.0.

k	$S \in \hat{\Sigma}_k$	x_k^*	r_k	c(S)
1	1 2 3 4 5	47.3 38.0 25.0 37.0 29.3 12.3	-0.667	176
	1 2 3 4 6			159
	5 6			41
2	1 2 6	47.3 38.0 25.0 37.0 29.3 12.3	-0.667	97
	3 4 5 6			103
3	1 6	47.3 38.0 25.0 37.0 29.3 12.3	-0.667	59
	2 3 4 5 6			141
4	1 5	47.3 38.0 27.7 34.3 29.3 12.3	-0.667	76
5	1 3 4 5	47.3 38.0 27.7 34.3 29.3 12.3	-0.667	138
6	2 5 6	47.3 38.0 27.7 34.3 29.3 12.3	-0.667	79
7	1 2 5	47.3 38.0 27.7 34.3 29.3 12.3	-0.667	114
8	1 3 4 6	47.3 38.0 25.0 37.0 29.3 12.3	-0.667	121
9	1 5 6	47.3 38.0 25.7 36.3 29.3 12.3	0.000	89
10	1 3 4 5 6	47.3 38.0 25.7 36.3 29.3 12.3	0.000	151
11	1 2 5 6	47.3 38.0 25.7 36.3 29.3 12.3	0.000	127
12	3 4	47.3 38.0 25.7 36.3 29.3 12.3	0.000	62
13	2	47.3 38.0 25.7 36.3 29.3 12.3	0.000	38
14	2 3 4	47.3 38.0 25.7 36.3 29.3 12.3	0.000	100
15	1 3 5 6	47.3 38.0 26.3 35.7 29.3 12.3	0.667	116
	2 4 5 6			116

(a) Simplex-method

k	$S \in \hat{\Sigma}_k$	x_k^*	r_k	c(S)
1	1 5	47.3 38.0 26.7 35.3 29.3 12.3	-0.667	76
	1 2 5			114
	1 3 4 5			138
	1 2 3 4 5			176
	1 6			59
	1 2 6			97
	1 3 4 6			121
	1 2 3 4 6			159
	5 6			41
	2 5 6			79
	3 4 5 6			103
	2 3 4 5 6			141
2	2	47.3 38.0 26.2 35.8 29.3 12.3	0.000	38
	3 4			62
	2 3 4			100
	1 5 6			89
	1 2 5 6			127
	1 3 4 5 6			151
3	1	47.3 38.0 26.3 35.7 29.3 12.3	0.667	48
	1 2			86
	1 3 4			110
	1 2 3 4			148
	1 2 4 5			151
	1 2 4 6			134
	1 3 5 6			116
	1 2 3 5 6			154
	2 4 5 6			116

(b) Interior-point method

Figure 6.2: Results from algorithm 6.2.1

6.5 Constraint generation

Problem $P(k)$ has $2^n - 1$ constraints. Not only will the size of the problem be prohibitive even for moderate values of n, but we also need to compute $c(S)$ for every constraint (coalition). By using *constraint generation*[9], we can reduce the size of the LP-problems to be solved.

Let Ω be some subset of the coalitions. The first LP-problem, $P(1, \Omega)$, is

$$\begin{aligned} &\max r \\ &\text{subject to } x(S) + r \leq c(S) \qquad && \forall S \in \Omega \\ &x(N) = c(N) \end{aligned}$$

Given an optimal solution (x^*, r^*), we need to check whether there are coalitions outside Ω whose constraints are violated, i.e., we compute[10]

$$z^* = \min_{S \notin \Omega \cup \{N, \emptyset\}} \{c(S) - x^*(S)\}, \tag{6.18}$$

and, if $z^* < r^*$, we should add the coalition S^* corresponding to the optimal solution of (6.18) to Ω. Then we solve $P(1, \Omega)$, now including an inequality constraint for S^*. The process of adding new constraints to $P(1, \Omega)$ continues until $z^* \geq r^*$, at which point we can proceed to stage 2 of Algorithm 6.2.1. That is, we identify the active constraints, i.e., the set Σ_1, and form the new problem $P(2, \Omega)$. This procedure continues until the solution of the

[9]See Gilmore and Gomory (1962). Göthe-Lundgren et al. (1996) apply constraint generation to the problem of computing the pre-nucleolus.

[10]In general (6.18) is a difficult problem, but for our small example, for which all the feasible coalitions have been listed in Figure 6.1(c), we use (6.13)-(6.17), where s_i^* equals one if i is a member of the optimal coalition S^*, and zero otherwise. Constraint (6.14) requires each member of S^* to be covered by the selected routes, while (6.15) limits the search to those coalitions that have not previously been included in the LP-problem.

$$\min \sum_{t \in T} c_t y_t - \sum_{i \in N} x_i^* s_i \tag{6.13}$$

$$\text{subject to } \sum_{t \in T} a_{it} = s_i \qquad \forall i \in N \tag{6.14}$$

$$\sum_{i \notin S} s_i + \sum_{i \in S} (1 - s_i) \geq 1 \qquad \forall S \in \Omega \cup \{N, \emptyset\} \tag{6.15}$$

$$s_i \in \{0, 1\} \qquad \forall i \in N \tag{6.16}$$

$$y_t \in \{0, 1\} \qquad \forall t \in T \tag{6.17}$$

LP-problem is unique. The constraint generation approach is summarized as Algorithm 6.5.1 below.

Algorithm 6.5.1

Begin

$\Sigma^0 := \{S \subset N : S \neq \emptyset, N\}$

$k := 1$

Choose $\Omega \subset \Sigma^0$

Repeat

Solve $P(k, \Omega)$ and store the optimal solution as (x^*, r^*)

Solve $PS(\Omega, x^*)$ and store the optimal solution as (z^*, S^*)

$\Omega := \Omega \cup \{S^*\}$

If $z^* \geq r^*$ **or** $\Sigma^{k-1} \setminus \Omega = \emptyset$ **then**

$X^k := \{x \in \mathbf{R}^n : x \text{ is an optimal solution to } P(k, \Omega)\}$

$\Sigma_k := \{S \in \Sigma^{k-1} \cap \Omega : x(S) = c(S) - r_k \ \forall x \in X^k\}$

$\Sigma^k := \Sigma^{k-1} \setminus \Sigma_k$

$k := k + 1$

endif

until $|X^{k-1}| = 1$

end

The LP-problem of the kth (major) iteration of the procedure, denoted $P(k, \Omega)$, is defined by (6.1), (6.2), (6.4), and

$$x(S) + r \leq c(S) \qquad\qquad \forall S \in \Sigma^{k-1} \cap \Omega, \qquad (6.19)$$

and the sub-problem $PS(\Omega, x^*)$, to be solved in order to identify constraints that are violated given the solution x^*, is given by (6.18). Since, in most applications, the sub-problem $PS(\Omega, x^*)$ is hard to solve, we choose to include the coalition S^* in Ω in every iteration, even for those where $z^* \geq r^*$.

Computational details from Algorithm 6.5.1 for our example are shown in Figures 6.3 and 6.4. Initially, Ω was set equal to the set of feasible coalitions, i.e., those shown in Figure 6.1(c). We see that the difference, with respect to the number of iterations needed, between using an extreme-point method and an interior-point method is still significant, although not as dramatic as when all coalitions where included in the LP-problem initially.

Iteration	S^*	x^*						r^*	z^*
1	1 2 3 4 5	47.1	38.3	25.3	37.0	29.1	12.1	-0.286	-0.857
2	2 3 4 5 6	46.8	38.4	25.4	37.0	28.8	12.6	-0.400	-1.200
3	2 4 5 6	47.3	37.3	24.3	38.3	29.3	12.3	-0.667	-1.333
4	2 5 6	47.3	38.7	25.0	36.3	29.3	12.3	-0.667	-1.333
5	3 4 5 6	47.3	38.0	25.0	37.0	29.3	12.3	-0.667	-0.667
	$\hat{\Sigma}_1 = \{\{1,6\},\{1,2,3,4,5\},\{2,3,4,5,6\}\}$								
6	1 2 3 4 6	47.3	38.3	25.3	37.0	28.7	12.3	-0.333	-1.333
7	1 3 4 6	47.3	38.0	25.0	37.0	29.3	12.3	-0.667	-0.667
	$\hat{\Sigma}_2 = \{\{1,5\},\{1,2,3,4,6\}\}$								
8	1 2 4 6	47.3	38.0	25.0	37.0	29.3	12.3	-0.667	-0.667
	$\hat{\Sigma}_3 = \{\{2,5,6\},\{1,3,4,6\}\}$								
9	1 2 6	47.3	38.0	25.0	37.0	29.3	12.3	-0.667	-0.667
	$\hat{\Sigma}_4 = \{\{5,6\}\}$								
10	1 2 4 5	47.3	38.0	25.0	37.0	29.3	12.3	-0.667	-0.667
	$\hat{\Sigma}_5 = \{\{1,2,6\}\}$								
11	1 3 4 5	47.3	38.0	25.0	37.0	29.3	12.3	-0.667	-0.667
	$\hat{\Sigma}_6 = \{\{3,4,5,6\}\}$								
12	1 2 5	47.3	38.0	25.0	37.0	29.3	12.3	-0.667	-0.667
	$\hat{\Sigma}_7 = \{\{1,3,4,5\}\}$								
13	1 2 4 5 6	47.3	38.0	25.0	37.0	29.3	12.3	-0.667	-0.000
	$\hat{\Sigma}_8 = \{\{1,2,5\}\}$								
14	2 3 4	47.3	38.0	25.7	36.3	29.3	12.3	0.000	0.000
	$\hat{\Sigma}_9 = \{\{2\}\}$								
15	1 2 5 6	47.3	38.0	25.7	36.3	29.3	12.3	0.000	0.000
	$\hat{\Sigma}_{10} = \{\{2,3,4\}\}$								
16	1 5 6	47.3	38.0	25.7	36.3	29.3	12.3	0.000	0.000
	$\hat{\Sigma}_{11} = \{\{1,2,5,6\}\}$								
17	1 3 4 5 6	47.3	38.0	25.7	36.3	29.3	12.3	0.000	0.000
	$\hat{\Sigma}_{12} = \{\{1,5,6\}\}$								
18	1 2	47.3	38.0	25.7	36.3	29.3	12.3	0.000	0.667
	$\hat{\Sigma}_{13} = \{\{1,3,4,5,6\}\}$								
19	1 2 3 4	47.3	38.0	25.7	36.3	29.3	12.3	0.000	0.667
	$\hat{\Sigma}_{14} = \{\{3,4\}\}$								
20	1 2 3 5 6	47.3	38.0	26.3	35.7	29.3	12.3	0.667	0.667
	$\hat{\Sigma}_{15} = \{\{1,2,3,4\}\}$								
21	1 3 4	47.3	38.0	26.3	35.7	29.3	12.3	0.667	0.667
	$\hat{\Sigma}_{16} = \{\{1,2,4,5\},\{1,2,3,5,6\}\}$								

Figure 6.3: Results for Algorithm 6.5.1 when the LP-problems are solved using the Simplex-method

Iteration	S^*	x^*						r^*	z^*
1	1 2 3 4 5	47.1	38.3	27.8	34.5	29.1	12.1	-0.286	-0.857
2	2 3 4 5 6	46.8	38.4	28.0	34.4	28.8	12.6	-0.400	-1.200
3	1 2 3 4 6	47.3	38.4	27.9	34.4	28.6	12.3	-0.667	-1.358
4	3 4 5 6	47.3	38.0	27.3	34.7	29.3	12.3	-0.667	-0.691
5	2 5 6	47.3	38.2	27.1	34.7	29.3	12.3	-0.667	-0.835
6	1 2 6	47.3	38.0	26.9	35.1	29.3	12.3	-0.667	-0.667
	$\hat{\Sigma}_1 = \{\{1,5\},\{1,6\},\{5,6\},\{1,2,3,4,5\},\{2,3,4,5,6\},\{1,2,3,4,6\},\{3,4,5,6\},\{2,5,6\}\}$								
7	1 2 5	47.3	38.0	26.5	35.5	29.3	12.3	-0.667	-0.667
	$\hat{\Sigma}_2 = \{\{1,2,6\}\}$								
8	1 3 4 6	47.3	38.0	26.5	35.5	29.3	12.3	-0.667	-0.667
	$\hat{\Sigma}_3 = \{\{1,2,5\}\}$								
9	1 3 4 5	47.3	38.0	26.5	35.5	29.3	12.3	-0.667	-0.667
	$\hat{\Sigma}_4 = \{\{1,3,4,6\}\}$								
10	1 2 5 6	47.3	38.0	26.5	35.5	29.3	12.3	-0.667	0.000
	$\hat{\Sigma}_5 = \{\{1,3,4,5\}\}$								
11	2 3 4	47.3	38.0	26.7	35.3	29.3	12.3	0.000	0.000
	$\hat{\Sigma}_6 = \{\{2\},\{3,4\},\{1,2,5,6\}\}$								
12	1 5 6	47.3	38.0	26.7	35.3	29.3	12.3	0.000	0.000
	$\hat{\Sigma}_7 = \{\{2,3,4\}\}$								
13	1 3 4 5 6	47.3	38.0	26.7	35.3	29.3	12.3	0.000	0.000
	$\hat{\Sigma}_8 = \{\{1,5,6\}\}$								
14	1 2 3 5 6	47.3	38.0	26.8	35.2	29.3	12.3	0.000	0.231
	$\hat{\Sigma}_9 = \{\{1,3,4,5,6\}\}$								
15	1 2 4 6	47.3	38.0	26.0	36.0	29.3	12.3	0.667	0.359
16	2 4 5 6	47.3	38.0	26.3	35.7	29.3	12.3	0.667	0.667
	$\hat{\Sigma}_{10} = \{\{1\},\{1,2,3,5,6\},\{1,2,4,6\}\}$								

Figure 6.4: Results for Algorithm 6.5.1 when the LP-problems are solved using an interior-point method

6.6 Conclusion

We have shown, via an example, that the number of iterations needed by Kopelowitz's (1967) procedure can be reduced by utilizing an interior-point method instead of an extreme-point method. When, in Section 6.4, all the coalition values were specified in advance, the difference in the number of iterations needed was dramatic. In Section 6.5, when the constraints of the LP-problem had to be generated in the course of the procedure, the difference was less significant.

The pre-nucleolus is indeed an interior-point of the core, if such a point exists. To illustrate this, note that $P(1)$ is equivalent to finding the *least core*[11], i.e., finding the smallest nonempty ϵ-core:

$$\max\{\epsilon : \emptyset \neq C_\epsilon(c) \subseteq C(c)\}$$

The primal solutions from an interior-point method will probably be a better "guess" as to what the final solution (the pre-nucleolus) will be. If we have to resort to constraint generation, this may be useful, since having a solution that is close to the final solution will probably prevent the generation of some unnecessary constraints.

[11] Maschler et al. (1979).

Chapter 7

Allocation of Fixed Transmission Costs in Electricity Networks

7.1 Introduction

Prior to 1988, most countries had electric power systems characterized by vertical integration and little competition. Under this regime, a single company controlled the entire production, transmission, and distribution system for a fixed geographic area. Starting with the deregulation of the British power system, many countries have created systems where production is subject to competition, but where transmission and distribution are still performed by monopoly entities. In order to facilitate competition in production, methods for allocating the cost of the transmission and distribution systems among the system users are needed.

We will in this chapter study the allocation of the cost of a transmission system. The system, as well as providing *basic transmission services*, i.e., transporting power from producers to consumers, also needs to provide *ancillary services*, such as the provision of operating reserves in order to handle generator failures, reactive support in order to maintain voltage levels within acceptable limits, and congestion management. We will focus on the basic transmission services, hence the most important part of the system cost is the cost of installing the transmission capacity, e.g. the power lines.

Existing and proposed allocation methods for allocating transmission costs can[1] be classified as belonging to one of the following groups:

(i) Embedded cost methods.

(ii) Incremental/marginal cost methods.

(iii) Combinations of (i) and (ii).

The *embedded cost* methods[2] allocate the total cost among the network users in proportion to some measure of the extent to which they make use of the network. This measure may e.g. be the amount of power transacted (the *postage-stamp rate* method), not taking into account the distance over which the power travels in the network. This method, popular in practice because of its simplicity, thus may lead to cross-subsidization of long-distance transactions by short-distance transactions. The *contract path* method also considers distances, but in a very crude way, assuming that the power flow follows the single shortest path between the seller and the buyer. Since power does not flow over a single path[3], the contract path method may also result in cross-subsidization. To alleviate this problem, the *MW-mile* method (Shirmohammadi et al. (1989).), measures the power flow that a transaction causes on every line in the system. Multiplying this flow with the length of the line, and summing over all lines, we get a measure of the transaction's use of the system.

While embedded cost methods allocate the system cost among *all* transactions in the system, the *incremental/marginal* cost methods take the view that a new transaction should cover the additional cost that it causes. An *incremental* approach would then be to calculate the total cost with and without the transaction, and let the transaction cover the difference. A *marginal* approach would be to multiply the amount of transacted power by the cost of an extra unit of power (e.g. the cost of an extra unit of power injected at some location in the network, obtained as the dual from the solution of a cost minimizing optimization problem). Note that, in contrast

[1]Marangon Lima (1996) and Shirmohammadi et al. (1996).
[2]See Pan et al. (2000) for an overview.
[3]This phenomenon is called "loop flow", and is illustrated by Example 7.2.1.

with the embedded cost methods, incremental/marginal methods may not recover the total system cost.[4] Also, the resulting allocations may depend on the ordering of the transactions.

Much of the literature on transmission cost allocation concentrates on presenting new methodology, especially new methods designed to measure a transaction's extent of usage of the system. We do not present a cost allocation method that returns a single-point solution. Rahter we present a model that can be used to evaluate cost allocations resulting from existing methods, using the core of a cooperative game. The network is given, i.e., we do not allow the construction of new lines or the closing down of existing lines. The transmission network connects a set of regional electricity markets, and the producers and consumers of a region have a choice as to whether they want to make use of the network or just trade in the local market. Given that a subset of the regions trade with each other through the network, equilibrium production and consumption quantities are determined from the demand and supply functions of the regions, as well as the capacities of the network.

Due to loop flow and limited line capacities, the combined surplus of producers/consumers will sometimes decrease when the inhabitants of a regional market (or several local markets) join the other markets via the network, although this will never happen if we also take into account the capacity charges collected by the network owner.[5] We will use the change in surplus when a set (coalition) of regions join the network as an upper bound on the amount that the network owner can charge these regions in order to cover the transmission cost. In adding payments from several regions together, we are assuming that sidepayments can be made between inhabitants of different regions. In Section 7.3, we formulate a cooperative game with transferable utility, and show that the core of this game is nonempty if there are no binding capacity constraints for the grand coalition. In Section 7.5, we illustrate, using examples, how existing cost allocation methods (postage-stamp rate methods and distribution factors á la Rudnick et al. (1995)) may fail to yield

[4]See e.g. Pérez-Arriaga et al. (1995) .

[5]This statement refers to (7.10), where it is assumed that the equilibrium corresponds to an optimal solution of (7.6)-(7.9). In practice this may not be the case, e.g., if zonal pricing (Bjørndal and Jörnsten (2001)) is used .

core elements. By combining several cost allocation methods, however, we may be able to produce core allocations, and in Section 7.6 we describe an LP-based procedure for doing this.

7.2 Regional electricity markets

We consider an electric power network that supports alternating currents (AC). The current follows a sinusoidal wave form, and the *instantaneous* current (at some node in the network) at time t is given by

$$i(t) = I_{max} \cos(\omega t),$$

where I_{max} is the amplitude of the current, and ω is its frequency measured in radians per second. The voltage may lag/lead the current by some phase angle δ, thus the instantaneous voltage is given by

$$v(t) = V_{max} \cos(\omega t + \delta)$$

Instantaneous *power* is then given by the product

$$i(t)v(t)$$

We shall in this chapter be concerned with *average* (real) power flows in a network with n nodes. Let V_i and θ_i be the voltage amplitude and voltage angle, respectively, at node i, measured relative to some reference node. Also, let r_{ij} and x_{ij} be the resistance and reactance, respectively, of the line between i and j, and define[6] $G_{ij} := -r_{ij}/(r_{ij}^2+x_{ij}^2)$ and $B_{ij} := x_{ij}/(r_{ij}^2+x_{ij}^2)$. It can be shown[7] that the average power flow from node i to node j, by applying Kirchoff's laws, is given by

$$q_{ij} = -G_{ij}V_i^2 + G_{ij}V_iV_j \cos(\theta_i - \theta_j) + B_{ij}V_iV_j \sin(\theta_i - \theta_j). \tag{7.1}$$

[6] G_{ij} and B_{ij} are known as the *conductance* and *susceptance*, respectively, of the line between i and j, and determine the elements of the complex admittance matrix Y, where $Y_{ij} := G_{ij} + \mathrm{j}B_{ij}$, and where $\mathrm{j} = \sqrt{-1}$. The elements of the diagonal are determined by setting $G_{ii} := \sum_{j\neq i} -G_{ij}$ and $B_{ii} := \sum_{j\neq i} -B_{ij}$.

[7] In Bergen and Vittal (1986), page 326, the average power injected into the system via node i is written

$$\sum_{j=1}^{n} V_iV_j(G_{ij}\cos(\theta_i - \theta_j) + B_{ij}\sin(\theta_i - \theta_j))$$

When modeling the effect of capacity constraints, as in Wu et al. (1996), resistances are often assumed to be zero, resulting in $G_{ij} = 0$. Note that this assumption gives a lossless system, where $q_{ij} = -q_{ji}$. Furthermore, since the angle difference $\theta_i - \theta_j$ is usually small, it is often assumed that $\sin(\theta_i - \theta_j) \approx \theta_i - \theta_j$. Hence (7.1) reduces to the linear expression

$$q_{ij} = B_{ij}(\theta_i - \theta_j) \tag{7.2}$$

where $B_{ij} = 1/x_{ij}$, and where we have assumed, without loss of generality, that quantities are normalized[8] so that $V_i = V_j = 1$. The angles $\theta_1, ..., \theta_n$ are unknowns, and can be determined from the power injections/withdrawals in the nodes. Let the demand for power in node i be given by q_i^d, the supply by q_i^s, and let $q_i := q_i^s - q_i^d$. Then, assuming that (7.2) properly represents line flows, the voltage angles can be determined[9] from

$$q_i = \sum_{j=1}^{n} q_{ij} = \sum_{j=1}^{n} B_{ij}(\theta_i - \theta_j) \qquad 1 \le i \le n. \tag{7.3}$$

A large current flowing through a line will cause the temperature of the line to rise. Since very high temperatures will damage the line, the line flow cannot exceed a certain level, hereafter referred to as the (thermal) capacity of the line. If the capacity of line (i, j) is CAP_{ij}, this constraint is given by

$$q_{ij} \le CAP_{ij} \qquad 1 \le i, j \le n.$$

Example 7.2.1 [Figure 7.1] Here, $n = 3$, and all line reactances are equal to one, i.e., $x_{ij} = 1$ for all i and j. One MW of power is injected in node 1, and the same amount of power is withdrawn in node 3, as indicated next to the nodes. There is no net injection/withdrawal of power in node 2. From

In order to find line flows, we rewrite

$$V_i^2(G_{ii}\cos(\theta_i - \theta_i) + B_{ii}\sin(\theta_i - \theta_i)) + \sum_{j \ne i} V_i V_j (G_{ij}\cos(\theta_i - \theta_j) + B_{ij}\sin(\theta_i - \theta_j))$$
$$= - V_i^2 \sum_{j \ne i} G_{ij} + \sum_{j \ne i} V_i V_j (G_{ij}\cos(\theta_i - \theta_j) + B_{ij}\sin(\theta_i - \theta_j)),$$

where we have used $\cos(0) = 1$, $\sin(0) = 0$, and $G_{ii} = \sum_{j \ne i} -G_{ij}$.

[8]See Section 5.5 in Bergen and Vittal (1986).

[9]In fact, only angle *differences* matter, so we may arbitrarily fix any one of the voltage angles, e.g. $\theta_1 = 0$.

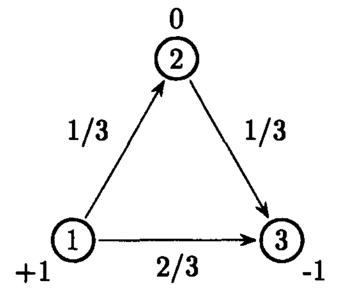


Figure 7.1: Example 7.2.1

(7.3), and by using the fact that this is a lossless system, i.e., $q_{ij} = -q_{ji}$, we get the three equations of (7.4).

$$\begin{aligned} q_{12} + q_{13} &= 1 \\ -q_{12} + q_{23} &= 0 \\ -q_{13} - q_{23} &= -1 \end{aligned} \tag{7.4}$$

Since $q_{12} = \theta_1 - \theta_2$, $q_{23} = \theta_2 - \theta_3$, and $q_{13} = \theta_1 - \theta_3$, we also have

$$q_{13} = q_{12} + q_{23}. \tag{7.5}$$

The equations in (7.4) correspond, in terms of power flows, to Kirchoff's junction rule, whereas (7.5) represents Kirchoff's loop rule. Solving (7.4) and (7.5) yields $q_{12} = q_{23} = 1/3$ and $q_{13} = 2/3$.

We see that the power flow over the path consisting of the lines (1,2) and (2,3) cannot be determined independently of the flow over line (1,3). If there is a capacity constraint limiting the flow over line (1,3) to 1/2, then the loop rule will reduce the power flow over the route consisting of (1,2) and (2,3) to 1/4. After the introduction of the capacity constraint, we cannot transport more than 1/2 + 1/4 = 3/4 units of power from node 1 to node 3. Hence, even though there is surplus capacity over the lines (1,2) and (2,3), we are not able to utilize this capacity due to the loop rule. ◁

In Example 7.2.1 the net injections/withdrawals of power were specified exogenously. In the sequel we will assume that the demand for real (average) power in node i is described by the (non-increasing) demand function $p_i^d(q)$, and the supply is likewise described by the (non-decreasing) supply function $p_i^s(q)$. A market equilibrium can be determined by solving the following (convex) problem:

$$\text{maximize} \sum_{i=1}^{n} \left[\int_0^{q_i^d} p_i^d(q)dq - \int_0^{q_i^s} p_i^s(q)dq \right] \tag{7.6}$$

subject to

$$q_i^s - q_i^d = \sum_{j=1}^{n} q_{ij} = \sum_{j=1}^{n} B_{ij}(\theta_i - \theta_j) \qquad 1 \le i \le n \tag{7.7}$$

$$B_{ij}(\theta_i - \theta_j) \le CAP_{ij} \qquad 1 \le i, j \le n \tag{7.8}$$

In all our examples, we will use the linear demand and supply functions $p_i^d(q) = a_i - b_i q$ and $p_i^s(q) = c_i q$, where $a_i, b_i, c_i > 0$. Moreover, if nothing else is stated explicitly in our examples, all reactances will be equal to one, i.e., $x_{ij} = B_{ij} = 1$ for all $i \neq j$.

We shall study situations in which only a subset of the nodes use the network. Let $N := \{1, ..., n\}$. When only the consumers and producers of $S \subseteq N$ use the network, we will assume that the inhabitants of the remaining nodes $N \setminus S$ trade only among themselves, i.e., (7.6) – (7.8) will be solved subject to the additional constraint

$$q_i^s - q_i^d = 0 \qquad i \in N \setminus S. \tag{7.9}$$

Let $q_i^s(S)$, $q_i^d(S)$, and $p_i(S)$ represent optimal values for node i in a solution to (7.6) – (7.9), and let $\Pi(S)$ be the corresponding value of (7.6). Note that because of the capacity constraints in (7.8), the equilibrium prices of two different nodes may differ.

The total surplus will always be higher when more nodes use the network, i.e., we have

$$\Pi(R) \le \Pi(S) \qquad \text{for all } R, S \text{ such that } R \subseteq S \subseteq N, \tag{7.10}$$

since the only effect of adding node i to S is to remove one of the constraints in (7.9). The structure of (7.6) allows us to define the surplus of each individual node i as

$$\begin{aligned}\Pi_i(S) &:= \text{consumer surplus} + \text{producer surplus} \\ &= \left[\int_0^{q_i^d(S)} p_i^d(q)dq - p_i(S)q_i^d(S)\right] + \left[p_i(S)q_i^s(S) - \int_0^{q_i^s(S)} p_i^s(q)dq\right] \\ &= \int_0^{q_i^d(S)} p_i^d(q)dq - \int_0^{q_i^s(S)} p_i^s(q)dq + p_i(S)\left[q_i^s(S) - q_i^d(S)\right]. \quad (7.11)\end{aligned}$$

The first two terms of (7.11) represent the area under the demand and supply curve, respectively, in node i. The last term is the net income from sale of surplus power, and may be negative. Hence the value of (7.6) may be written as

$$\Pi(S) = \sum_{i\in N} \Pi_i(S) + \sum_{i\in N} p_i(S)\left[q_i^d(S) - q_i^s(S)\right] := \sum_{i\in N} \Pi_i(S) + \Pi_{no}(S).$$

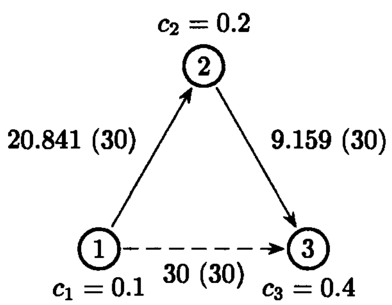
The number $\Pi_{no}(S)$ is the *merchandizing surplus*, as defined by Wu et al. (1996), and can be interpreted as short-term capacity charges collected by the network owner. This interpretation becomes clearer from the following relationships, as shown by Wu et al. (1996),

$$\Pi_{no}(S) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \left[p_j(S) - p_i(S)\right] q_{ij}(S) = \sum_{i=1}^{n}\sum_{j=1}^{n} \mu_{ij}(S)CAP_{ij}, \quad (7.12)$$

where $\mu_{ij}(S)$ is the dual price for line (i,j). It also follows[10] from Wu et al. (1996) that $\Pi_{no}(S) \geq 0$ for all $S \subseteq N$.

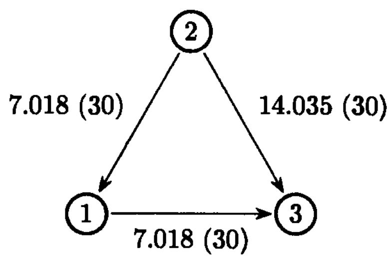
Example 7.2.2 [Figure 7.2] In this example, the demand functions are identical in all the three nodes, with $a_i = 20$ and $b_i = 0.05$. The parameters of the supply functions (c_i) are 0.1, 0.2, and 0.4 in node 1, 2, and 3, respectively, as indicated next to the nodes in part (a) of the figure. Reactances are all equal to one, and the capacities of all three lines are 30. In the equilibrium solution, there is a surplus of power in node 1, exactly matched by deficits in nodes 2 and 3. This results in a power flow from node 1 to the other two nodes. The amount that can be exported is limited by the capacity of line (1,3). Because of the binding capacity constraint, the equilibrium prices varies over the set of nodes.

[10]This is because the equilibrium solution $(p(S), q^d(S), q^s(S))$ is an optimal solution to (7.6)-(7.9). In practice (see e.g. Bjørndal and Jörnsten (2001)) this may not be the case.



i	q_i^s	q_i^d	p_i
1	150.280	99.439	15.028
2	77.664	89.346	15.533
3	40.093	79.252	16.037

(a) Market equilibrium when all nodes use the network.



i	q_i^s	q_i^d	p_i
1	133.333	133.333	13.333
2	84.211	63.158	16.842
3	42.105	63.158	16.842

(b) Market equilibrium when only 2 and 3 use the network.

S			$\Pi_1(S)$	$\Pi_2(S)$	$\Pi_3(S)$	$\sum_{i\in N} \Pi_i(S)$	$\Pi_{no}(S)$	$\Pi(S)$
1			1333.333	800.000	444.444	2577.778	0.000	2577.778
	2		1333.333	800.000	444.444	2577.778	0.000	2577.778
		3	1333.333	800.000	444.444	2577.778	0.000	2577.778
1	2		1355.372	826.446	444.444	2626.263	0.000	2626.263
1		3	1367.083	800.000	489.444	2656.528	42.500	2699.028
	2	3	1333.333	808.864	454.294	2596.491	0.000	2596.491
1	2	3	1376.414	802.730	478.520	2657.664	45.420	2703.084

(c) Surpluses for node inhabitants and the network owner.

Figure 7.2: Example 7.2.2

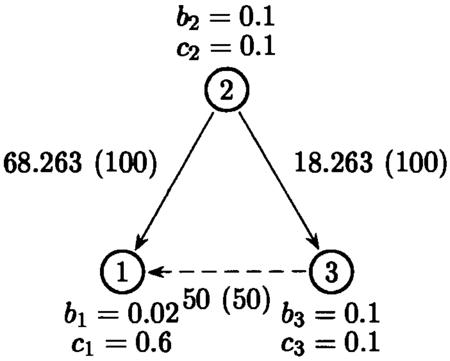
Although society as a whole, i.e., the node inhabitants and the network owner combined, are always better off as more nodes start using the network, this is not necessarily true for individual nodes. If node 1 stops using the network, we get the equilibrium shown in part (b) of the figure. The surplus of 1 drops from 1376.414 ($= \Pi_1(1,2,3)$) to 1333.333 ($= \Pi_1(2,3)$). Node 2 experiences an *increase* in surplus from 802.730 to 808.864, whereas the inhabitants of node 3 see their surplus drop from 478.520 to 454.294. In total, the surplus of all node inhabitants decreases from 2657.664 to 2596.491. The network owner looses 45.420 when node 1 stops using the network, since the network will no longer be congested. ◁

Example 7.2.3 [Figure 7.3] In Example 7.2.2, although the surplus of individual nodes in some cases increased when new users left the network, the combined surplus of the node inhabitants ($\sum_{i \in N} \Pi_i$) always decreased. To construct an example where this is not the case, let $a_i = 20$ for all i, and let the other characteristics of the network as well as the node inhabitants be as described in Figure 7.3(a). By comparing the situation in (a) and (b), we see that node 3, by using the network, causes congestion on line (1,3). Hence, if the inhabitants of node 3 does not use the network, the total surplus of *all* node inhabitants increases from 2670.284 to 2953.488, i.e., an increase of 283.204. Since the network owner looses 410.929, the effect on the total surplus is negative. ◁

7.3 The cost allocation problem and its game

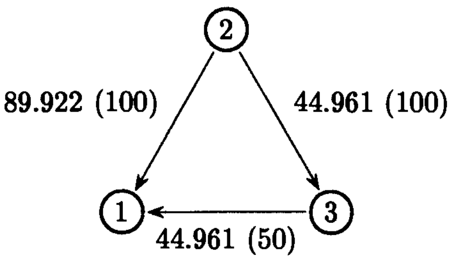
We will now consider the allocation of (fixed) transmission network costs among the users of the network. Some of the network cost is covered through short-term capacity charges, i.e., through the revenue that the network owner gets from regulating the short-term flow of the network. This revenue is given by $\Pi_{no}(N)$ if the inhabitants of all nodes are using the network. Typically[11], this income falls short of the total cost of the network, and the residual C has to be covered through additional fees.

[11] According to Pérez-Arriga et al. (1995), one should not expect more than 30% of total network cost to be recovered from short-term capacity charges.



i	q_i^s	q_i^d	p_i
1	28.443	146.707	17.066
2	143.263	56.737	14.326
3	115.868	84.132	11.587

(a) Market equilibrium when all nodes use the network.



i	q_i^s	q_i^d	p_i
1	27.907	162.791	16.744
2	167.442	32.558	16.744
3	100.000	100.000	10.000

(b) Market equilibrium when only 1 and 2 use the network.

S			$\Pi_1(S)$	$\Pi_2(S)$	$\Pi_3(S)$	$\sum_{i\in N} \Pi_i(S)$	$\Pi_{no}(S)$	$\Pi(S)$
1			322.581	1000.000	1000.000	2322.581	0.000	2322.581
	2		322.581	1000.000	1000.000	2322.581	0.000	2322.581
		3	322.581	1000.000	1000.000	2322.581	0.000	2322.581
1	2		498.648	1454.840	1000.000	2953.488	0.000	2953.488
1		3	377.016	1000.000	1140.625	2517.641	311.492	2829.133
	2	3	322.581	1000.000	1000.000	2322.581	0.000	2322.581
1	2	3	457.931	1187.172	1025.180	2670.284	410.929	3081.213

(c) Surpluses for node inhabitants and the network owner.

Figure 7.3: Example 7.2.3

By a *cost allocation* we will understand a vector $x \in \mathbf{R}^N$, where x_i is the amount to be paid by the inhabitants of node i. Let $x(S) := \sum_{i \in S} x_i$ represent the total cost allocated to the members of S. Because (7.10) implies that total surplus is maximized when all node inhabitants use the network, we would like the cost allocation to be such that all users are encouraged to use the network. Suppose a cost allocation x is such that

$$x(S) > \sum_{i \in N} [\Pi_i(N) - \Pi_i(N \setminus S)]. \tag{7.13}$$

The left-hand side of of (7.13) is the amount paid by the coalition S to the network owner, while the right-hand side is the change in the total surplus, excluding the surplus of the network owner, that would occur if the members of S stopped using the network. We will in the sequel assume that sidepayments are possible between the node inhabitants, but not between the network owner and node inhabitants. Hence, (7.13) implies that it would be better for society as a whole, except the network owner, if S left the network, meaning that the inhabitants of the set of nodes S will buy and sell power only in their local markets. Thus, a reasonable requirement for a cost allocation is that

$$x(S) \leq \sum_{i \in N} [\Pi_i(N) - \Pi_i(N \setminus S)] \qquad \forall S \subset N. \tag{7.14}$$

The owner of the network wants to recover the total cost of the network, given by the number C, i.e., he seeks an allocation vector such that[12]

$$x(N) = C. \tag{7.15}$$

The requirements (7.14) and (7.15) corresponds to the core restrictions for the (cost) game v^C, where

$$v^C(S) := \begin{cases} \sum_{i \in N} [\Pi_i(N) - \Pi_i(N \setminus S)] & S \subset N, \\ C & S = N. \end{cases} \tag{7.16}$$

[12]We will in the following assume that

$$C \leq \sum_{i \in N} [\Pi_i(N) - \Pi_i(\emptyset)].$$

If this was not the case, the grand coalition would prefer not to use the network when faced with a cost allocation x that satisfies (7.15).

In order to analyze the game v^C, we shall split it into two parts. Let $v^C := v^1 + v^2$, where, for any $S \subseteq N$,

$$v^1(S) := \sum_{i \in S} [\Pi_i(N) - \Pi_i(N \setminus S)]$$

and

$$v^2(S) := \sum_{i \in N \setminus S} [\Pi_i(N) - \Pi_i(N \setminus S)].$$

The number $v^1(S)$ can be interpreted as the value, for the members of S, of joining the network. Note that $\Pi_i(N \setminus S) = \Pi_i(\emptyset)$ for all $i \in N \setminus S$, since, if node i does not use the network, the surplus that i gets does not depend on which of the other nodes are currently using the network. Hence we have

$$v^1(S) = \sum_{i \in S} [\Pi_i(N) - \Pi_i(\emptyset)],$$

showing that v^1 corresponds to an additive game, with a unique core allocation where player $i \in N$ is allocated the amount $\Pi_i(N) - \Pi_i(\emptyset)$.

On the other hand, $v^2(S)$ measures the increased surplus that the members of $N \setminus S$ get when S joins. Note that $v^2(N) = 0$.

The games of Example 7.2.2 and 7.2.3 are shown in Figure 7.4. For Example 7.2.2 we also show the excess values for the allocation[13]

$$y = (54.545, 0.568, 24.773),$$

all of which are nonnegative, hence y is a member of the core. Thus, the core is nonempty. For Example 7.2.3, the allocation

$$z = (177.703, 160, 10)$$

is not a core member, since some of the excess values are negative. If node 3 left the network, then node 1 and 2 would gain $-v^2(3) = 308.385$, and node 3 would loose $v^1(3) - x_3 = 25.180 - 10 = 15.180$, giving a total gain of $308.385 - 15.180 = 293.205$. Note that it would not be in the interest of the network owner to have the users of node 3 leave the network, since the

[13]This is the nucleolus of the game, see Chapter 1.

S			$v^1(S)$	$v^2(S)$	$v^C(S)$	$e(S,y)$
1			43.080	18.092	61.172	6.628
	2		2.730	−1.594	1.136	0.568
		3	34.076	−2.675	31.401	6.628
1	2		45.810	34.076	79.886	24.773
1		3	77.156	2.730	79.886	0.568
	2	3	36.805	43.080	79.886	54.545
1	2	3	79.886	0.000	79.886	0.000

(a) Example 7.2.2

S			$v^1(S)$	$v^2(S)$	$v^C(S)$	$e(S,z)$
1			135.351	212.353	347.703	170.000
	2		187.172	−34.529	152.643	−7.357
		3	25.180	−308.385	−283.204	−293.204
1	2		322.523	25.180	347.703	10.000
1		3	160.531	187.172	347.703	160.000
	2	3	212.353	135.351	347.703	177.703
1	2	3	347.703	0.000	347.703	0.000

(b) Example 7.2.3

Figure 7.4: Examples of the game v^C

network would no longer be congested. If the users of node 3 left, he would loose $\Pi_{no}(N) + x_3 = 410.929 + 10 = 510.929$ in revenue. The loss of the network owner caused by node 3 leaving thus more than outweighs the gain of the node inhabitants.

In Example 7.2.3, the core is actually empty. To see this, note that if x is to be a core allocation, we need

$$x_3 \leq v^C(3) = -283.204 \text{ and } x_1 + x_2 + x_3 = v^C(N) = 347.703.$$

Combining these two conditions, we get

$$x_1 + x_2 = 347.703 - x_3 \geq 347.703 + 283.204 = 630.907,$$

which contradicts the core inequalities

$$x_1 \leq v^C(1) = 347.703 \text{ and } x_2 \leq v^C(2) = 152.643.$$

Theorem 7.3.1 *Suppose $C \leq v^1(N)$. If, for $S = N$, an optimal solution to (7.6)-(7.9) is still optimal when we remove the capacity constraints, i.e., when we set $CAP_{ij} = \infty$ for all $i, j \in N$, then the core of v^C is nonempty.*

Proof. Follows from Lemmas 7.3.2 and 7.3.3 below. □

Lemma 7.3.2 *Suppose $C \leq v^1(N)$. If $v^2 \geq 0$, then the core of v^C is nonempty.*

Proof. Since $v^C(S) = v^1(S) + v^2(S) \geq v^1(S)$ for all $S \subset N$, and since $v^C(N) = C \leq v^1(N)$, we can find a core allocation x for v^C by scaling the (unique) core allocation for v^1 down in the following manner:

$$x_i := [\Pi_i(N) - \Pi_i(\emptyset)] \frac{C}{v^1(N)}$$

□

According to Lemma 7.3.2, if existing users of the network always welcome new users, i.e., if $v^2(S) \geq 0$ for every $S \subset N$, then the core of the game v^C is nonempty.

Lemma 7.3.3 *If, for $S = N$, an optimal solution to (7.6)-(7.9) is still optimal when we remove the capacity constraints, then $v^2 \geq 0$.*

Proof. The statement $v^2 \geq 0$, by using the definition of v^2, is equivalent to

$$\sum_{i \in S} \Pi_i(N) \geq \sum_{i \in S} \Pi_i(S) \qquad \text{for all } S \subseteq N. \tag{7.17}$$

In order to prove 7.17, define $\hat{\Pi}(S)$ as the value of (7.6) subject to (7.7) and (7.9), i.e., we remove all capacity constraints. Note that since $q_{ij} = -q_{ji}$ and hence $\sum_{i=1}^n q_{ij} = 0$, condition (7.7) implies

$$\sum_{i \in N} (q_i^s - q_i^d) = 0. \tag{7.18}$$

If q^s and q^d satisfy (7.18), we can always find a set of voltage angles that satisfies (7.7), hence in the absence of capacity constraints we may replace

(7.7) by (7.18). By substituting (7.9) in (7.18), we get, for any $R \subseteq N$,

$$\hat{\Pi}(R) = \max\left\{\sum_{i\in R} f_i(q_i^s, q_i^d) : \sum_{i\in R}(q_i^s - q_i^d) = 0\right\} + \sum_{i\in N\setminus R} \Pi_i(\emptyset), \quad (7.19)$$

where

$$f_i(q_i^d, q_i^s) := \int_0^{q_i^d} p_i^d(q)dq - \int_0^{q_i^s} p_i^s(q)dq$$

is the part of the objective function (7.6) corresponding to node i, and where $\Pi_i(\emptyset) = \max\{f_i(q_i^s, q_i^d) : q_i^s = q_i^d\}$ is the social surplus in node i when the local market has to be cleared separately. Since

$$\hat{\Pi}(R) = \sum_{i\in R} \hat{\Pi}_i(R) + \sum_{i\in N\setminus R} \hat{\Pi}_i(\emptyset) + \hat{\Pi}_{no}(R),$$

and since $\hat{\Pi}_{no}(R) = 0$, by the absence of capacity constraints, we can write

$$\begin{aligned}
\sum_{i\in R} \hat{\Pi}_i(R) &= \max\left\{\sum_{i\in R} f_i(q_i^s, q_i^d) : \sum_{i\in R}(q_i^s - q_i^d) = 0\right\} \\
&= \min_\lambda \max_{q^d, q^s}\left\{\sum_{i\in R} f_i(q_i^s, q_i^d) + \lambda \sum_{i\in R}(q_i^s - q_i^d)\right\} \\
&= \min_\lambda \sum_{i\in R} \max_{q_i^d, q_i^s}\left\{f_i(q_i^d, q_i^s) + \lambda(q_i^s - q_i^d)\right\} \\
&:= \min_\lambda \sum_{i\in R} g_i(\lambda)
\end{aligned} \quad (7.20)$$

The second equality of 7.20 follows from duality and, since demand (supply) functions are non-increasing (non-decreasing), the function f_i is concave. Let $\lambda(R)$ be the optimal multiplier in (7.20).

In order to prove (7.17), we shall show that, for any $S \subseteq N$, we have

$$\sum_{i\in S} \Pi_i(N) = \sum_{i\in S} \hat{\Pi}_i(N) \geq \sum_{i\in S} \hat{\Pi}_i(S) \geq \sum_{i\in S} \Pi_i(S). \quad (7.21)$$

The equality in (7.21) follows from the assumption in the lemma. The first inequality follows from (7.20), which states that $\sum_{i\in S} \hat{\Pi}_i(N) = \sum_{i\in S} g_i(\lambda(N))$ and $\sum_{i\in S} \hat{\Pi}_i(S) = \sum_{i\in S} g_i(\lambda(S))$. Note that

$$\sum_{i\in S} g_i(\lambda(N)) \geq \sum_{i\in S} g_i(\lambda(S))$$

follows from the fact that $\lambda(S)$ is the optimal multiplier with respect to the coalition S.

The second inequality in (7.21) will be shown by noting that

$$\Pi(S) = \sum_{i \in S} \Pi_i(S) + \sum_{i \in N \setminus S} \Pi_i(\emptyset) + \Pi_{no}(S) \tag{7.22}$$

$$\hat{\Pi}(S) = \sum_{i \in S} \hat{\Pi}_i(S) + \sum_{i \in N \setminus S} \hat{\Pi}_i(\emptyset) \tag{7.23}$$

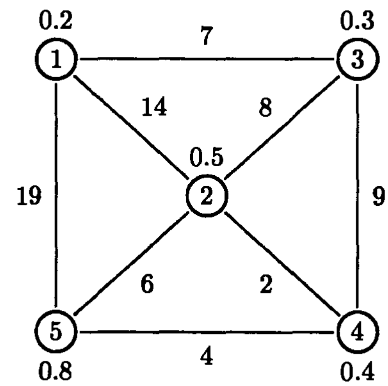
Since adding capacity constraints to (7.6)-(7.9) cannot increase the optimal objective function value, we must have $\Pi(S) \leq \hat{\Pi}(S)$. Also note that $\Pi_i(\emptyset) = \hat{\Pi}_i(\emptyset)$ for all $i \in N$, thus the second term in (7.22) equals the second term in (7.23). Then, since $\Pi_{no}(S) \geq 0$, and since $\Pi(S) \leq \hat{\Pi}(S)$, we must have $\sum_{i \in S} \hat{\Pi}_i(S) \geq \sum_{i \in S} \Pi_i(S)$. □

Note that it is possible to have $v^2(S) < 0$ for some $S \subset N$, and still have a nonempty core, as is illustrated by Example 7.2.2, for which the game is shown in Figure 7.4.

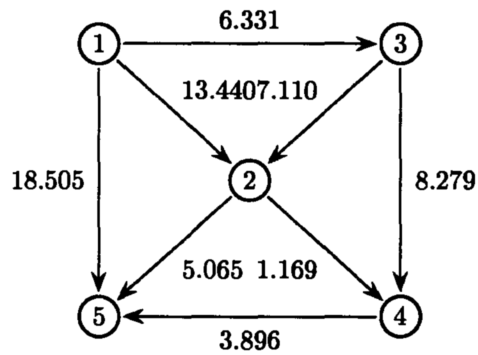
Even though the assumption in Theorem 7.3.1 holds, i.e., the grand coalition does not experience any capacity problems, this may not be the case for the sub-coalitions. This is illustrated by the next example. This example also illustrates the fact that, even though line capacities have been chosen so as to satisfy the needs of the grand coalition, changes in the production/consumption patterns may cause the core to *become* empty.

Example 7.3.4 [Figure 7.5] All the nodes have identical demand functions, with $a_i := 20$ and $b_i := 0.05$ for all $i \in N$. Cost parameters and line capacities are indicated in part (a) of Figure 7.5, and the line flows for the grand coalition in part (b). Line capacities have initially been set so as to accomodate the network flow corresponding to the grand coalition[14], and the cost of the network is $C := v^1(N) = 54.152$. Then, since there are no binding capacity constraints for the grand coalition, the core is nonempty, by Theorem 7.3.1. This does not, however, mean that capacity constraints are not binding for sub-coalitions, i.e., if the inhabitants of some nodes choose

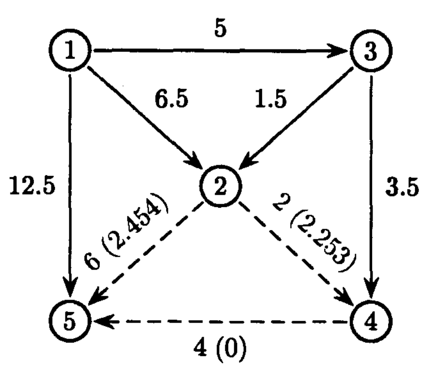
[14] Capacities have been set by rounding upwards (the absolute values of) the power flows corresponding to unconstrained dispatch for the grand coalition.



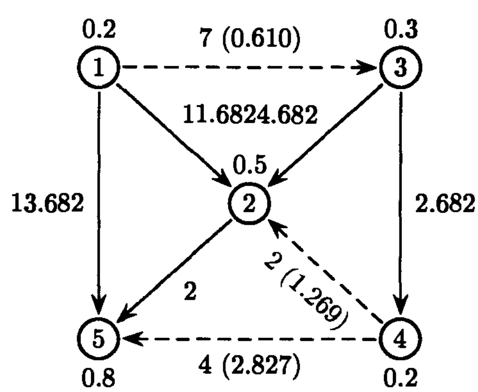
(a) Initial line capacities and production costs.



(b) Line flows for the grand coalition with the initial parameter values



(c) Line flows for the coalition $\{1, 4, 5\}$ with the initial parameter values.



(d) Optimal power flows for the grand coalition when the cost of production in node 4 is reduced.

Figure 7.5: Example 7.3.4

		S			$v^1(S)$	$v^2(S)$	$v^C(S)$	$e(S, nucl)$
1					20.949	12.266	33.215	13.594
	2				6.086	5.616	11.702	4.497
		3			0.003	0.879	0.882	-3.059
			4		0.220	-9.896	-9.676	**-4.670**
				5	9.115	11.020	20.135	9.858
1	2				27.034	6.889	33.924	7.099
1		3			20.951	12.972	33.924	10.363
	2	3			6.089	6.807	12.896	1.750
1			4		21.169	6.022	27.191	12.577
	2		4		6.306	2.060	8.366	6.168
		3	4		0.223	1.381	1.604	2.669
1				5	30.063	5.452	35.516	5.619
	2			5	15.201	10.782	25.983	8.502
		3		5	9.118	8.258	17.376	3.158
			4	5	9.335	10.379	19.714	14.444
1	2	3			27.037	6.887	33.924	3.158
1	2		4		27.255	2.644	29.898	8.080
1		3	4		21.172	12.975	34.147	15.592
	2	3	4		6.309	12.652	18.961	12.821
1	2			5	36.149	-2.107	34.042	-3.059
1		3		5	30.066	5.521	35.588	1.750
	2	3		5	15.204	21.169	36.372	14.951
1			4	5	30.284	1.673	31.957	7.066
	2		4	5	15.421	13.815	29.236	16.761
		3	4	5	9.338	17.421	26.759	17.547
1	2	3	4		27.258	9.115	36.372	10.613
1	2	3		5	36.152	0.220	36.372	**-4.670**
1	2		4	5	36.370	0.003	36.372	4.277
1		3	4	5	30.286	6.086	36.372	7.541
	2	3	4	5	15.424	20.949	36.372	19.956
1	2	3	4	5	36.372	0.000	36.036	0.000

Figure 7.6: The game, and excess values corresponding to the nucleolus, of Example 7.3.4 when the cost of production in node 4 is reduced

not to use the network, as is illustrated by part Figure 7.5(c), where the line flows corresponding to the coaliton $\{1, 4, 5\}$ are shown, and where we see that the flows of the lines (2,4), (2,5), and (4,5) are at their upper limit, as indicated by the dashed lines. Using the dual prices shown in parentheses, we can calculate the merchandizing surplus, using (7.12), as

$$\Pi_{no}(1, 4, 5) = 2 \cdot 2.253 + 6 \cdot 2.454 = 19.23.$$

After line capacities have been chosen, changes in the production and/or consumption pattern may take place. Suppose that c_4 drops from 0.4 to 0.2, i.e., production in node 4 becomes cheaper. After the change, the line flows are as shown in Figure 7.5(c), where the binding line capacities are indicated by dashed lines, and dual prices are shown in parentheses. If the grand coalition forms, the network owner will be able to collect

$$\Pi_{no}(N) = 7 \cdot 0.610 + 2 \cdot 1.269 + 4 \cdot 2.827 = 18.116$$

in short-term capacity charges. Assuming that C is the residual amount that needs to be covered after the short-term charges have been collected, his claim will now be reduced to 36.036 (= 54.152 - 18.116). Note that the reduced claim is smaller than $v^1(N) = 36.372$, i.e., the value for the grand coalition of using the network is larger than the network owner's claim. The resulting game v^C is shown in Figure 7.6. The nucleolus is given by the allocation vector

$$(19.620, 7.204, 3.941, -5.006, 10.276),$$

and the corresponding excess values are shown in Figure 7.6. Since the nucleolus is always a core element if the core is nonempty, and since some of the excess values are negative, the core must be empty. ◁

7.4 Possible responses to the problem of an empty core

When the core of v^C is empty, the network owner may find it difficult to allocate the entire cost C in such a way that no subset of network users will find it profitable to discontinue using the network. In this section we will discuss briefly how the network owner could handle such a situation.

One option for the network owner is to reduce his claim C in order to make the core of v^C nonempty. In order to find the maximal value of C that guarantees a nonempty core, he needs to solve the LP-problem:

$$\max x(N) \tag{7.24}$$

$$\text{subject to } x(S) \leq v^C(S) \qquad \text{for all } S \subset N \tag{7.25}$$

In Example 7.3.4, the solution of (7.24)-(7.25) is given by the allocation vector

$$x = (33.042, 0.000, 0.882, -9.676, 1.000),$$

which corresponds to a total claim of 25.249.

Another approach would be to punish those players that leave. The punishment can e.g. be in the form of a tax that those that choose not to join the network have to pay. Suppose, in Example 7.3.4, that such a tax is based on the amount of power produced/consumed in nodes that do not connect themselves to the network. If all the node markets are cleared separately, the following quantities will be observed:

$$\hat{q} = (86.473, 34.876, 57.195, 80.664, 22.372)$$

If a tax of ϵ is charged per unit of power, each node that chooses not to use the network will have to pay an amount of $\epsilon\hat{q}_i$. Thus the increased payment from the node inhabitants to the network owner when node i joins the network is

$$x_i - \epsilon\hat{q}_i,$$

where x is the allocation of the network cost. For the node inhabitants as a collective, it will be profitable to include the coalition S in the network if

$$x(S) - \epsilon\hat{q}(S) \leq v^C(S).$$

What is the lowest tax rate that makes it profitable for all node inhabitants to use the network, and at the same time allows the network owner to cover the network cost C? This is the solution to the LP-problem:

$$\min \epsilon \tag{7.26}$$

$$\text{subject to } x(S) - \epsilon\hat{q}(S) \leq v^C(S) \qquad \text{for all } S \subset N \tag{7.27}$$

$$x(N) = v^C(N) \tag{7.28}$$

The solution to (7.26)-(7.28) is given by the allocation vector

$$x = (36.355, 1.336, 3.073, -6.585, 1.857)$$

and the tax rate $\epsilon = 0.038311$. The approach followed here is similar to that of Tijs and Driessen (1986).

7.5 Other cost allocation methods and the core

The practical implementation of a tariff system must satisfy additional requirements such as:

(i) The cost allocation must be based on information that is available to *all* participants, i.e., both the network users and the network owner.

(ii) The procedure used to compute the cost allocation must not involve excessive computational costs.

(iii) The participants may have mental barriers excluding certain types of tariff systems from consideration. Negative contributions from any participants may e.g. be unacceptable.

An example of information that is private, i.e., not satisfying requirement (i), are the cost and demand parameters of producers and consumers. Here we will assume that the only information that satisfies requirement (i) are

- production and consumption quantities
- line flows
- physical characteristics of the network

Denote the observed production and consumption quantities of node i by $\tilde{q}_i^s$ and $\tilde{q}_i^d$, respectively, and the power flow over line (i,j) by $\tilde{q}_{ij}$.

So-called *postage-stamp rate* methods allocates the fixed cost in proportion to nodal (production and consumption) quantities. If the cost is allocated in proportion to *gross production* in the nodes, we obtain the allocation vector x^s, where

$$x_i^s := \tilde{q}_i^s \frac{C}{\sum_{i \in N} \tilde{q}_i^s}.$$

If *gross consumption* is used, we obtain x^d, where

$$x_i^d := \tilde{q}_i^d \frac{C}{\sum_{i \in N} \tilde{q}_i^d}.$$

Another approach would be to allocate the cost on basis of net quantities. Let, for node i, net injection/withdrawal of energy be denoted by

$$\tilde{q}_i^+ := \max(\tilde{q}_i^s - \tilde{q}_i^d, 0) \quad \text{and} \quad \tilde{q}_i^- := \max(\tilde{q}_i^d - \tilde{q}_i^s, 0).$$

Allocating costs in proportion to net injection/withdrawal gives the allocations x^+ and x^-, respectively, where

$$x_i^+ := \tilde{q}_i^+ \frac{C}{\sum_{i \in N} \tilde{q}_i^+} \quad \text{and} \quad x_i^- := \tilde{q}_i^- \frac{C}{\sum_{i \in N} \tilde{q}_i^-}.$$

In e.g. the tariff system of the Norwegian central network, a combination of x^s, x^d, x^+, and x^- is used.[15]

Alternatively, a cost allocation method may use information on how the power *flows* over the lines. This is the approach suggested by Rudnick et al. (1995). From the linear DC approximation we can find the (traditional) *distribution factors*, where α_{ij}^{kr} is the flow over line (i,j) caused by injecting one unit of power in node k and withdrawing one unit of power in (the reference) node r. These distribution factors depend on the choice of reference node, i.e., we have to specify the withdrawal as well as the injection node. Rudnick et al. (1995) suggest a cost allocation method that is independent of the

[15] The production quantities used are actually *available production capacities*, and are in the hydropower-based Norwegian system based on average inflow of water registered over a number of years. The consumption quantities used are based on registered *maximal consumption*, usually corresponding to the coldest day of the year. See Statnett (2000) and Statnett (2002).

choice of reference node, using *Generalized Generation (Load) Distribution Factors*, first defined by Ng (1981).

For the case where network costs are to be allocated among producers (generators), let β_{ij}^k denote the amount of flow over line (i,j) caused by injecting one unit of power in node k. These distribution factors can be determined by solving the linear equation system[16]

$$\tilde{q}_{ij} = \sum_{k \in N} \beta_{ij}^k \tilde{q}_k^s \qquad 1 \le i < j \le n, \tag{7.29}$$

$$\beta_{ij}^k = \alpha_{ij}^{kr} + \beta_{ij}^r \qquad 1 \le i < j \le n \qquad k \in N \setminus \{r\}. \tag{7.30}$$

While (7.29) requires that the distribution factors should explain the actual flow over the lines, (7.30) expresses the following: Injecting one unit of power in node k, which causes the flow β_{ij}^k over line (i,j), is equivalent to

(i) injecting one unit of power in node k, to be withdrawn in node r (α_{ij}^{kr}),

(ii) *and* injecting one unit of power in node r (β_{ij}^r).

When using these factors to allocate costs, Rudnick et al. (1995) propose to charge a producer for the use of a line only if that producer contributes to the positive flow over the line, i.e., using the modified distribution factors

$$\hat{\beta}_{ij}^k := \begin{cases} \beta_{ij}^k & \beta_{ij}^k \tilde{q}_{ij} > 0, \\ 0 & \text{otherwise.} \end{cases}$$

If the cost of line (i,j) is c_{ij}, then the amount to be paid by the producers in node k is

$$c_{ij} \frac{\hat{\beta}_{ij}^k \tilde{q}_k^s}{\sum_{\ell \in N} \hat{\beta}_{ij}^\ell \tilde{q}_\ell^s}.$$

[16]The system (7.29)-(7.30) implies that if the production in node r changes by $\Delta\tilde{q}_r^s$, then the change $\Delta\tilde{q}_j^d$ in the consumption of any node $j \in N$ will satisfy

$$\frac{\Delta\tilde{q}_r^s}{\tilde{q}^s} = \frac{\Delta\tilde{q}_j^d}{\tilde{q}_j^d},$$

where $\tilde{q}^s := \sum_{k \in N} \tilde{q}_k^s$, i.e., the relative change in consumption will be the same in all nodes. A proof can be found in Ng (1981). A similar result holds for the load distribution factors defined by (7.31)-(7.32), i.e., if consumption in node r changes by $\Delta\tilde{q}_r^d$, then the change $\Delta\tilde{q}_j^s$ in the production in node $j \in N$ will satisfy

$$\frac{\Delta\tilde{q}_r^d}{\tilde{q}^d} = \frac{\Delta\tilde{q}_j^s}{\tilde{q}_j^s},$$

where $\tilde{q}^d := \sum_{k \in N} \tilde{q}_k^d$.

The total amount to be paid by the producers in node k is then

$$x_k^{GDF+} := \sum_{i<j} c_{ij} \frac{\hat{\beta}_{ij}^k \tilde{q}_k^s}{\sum_{\ell \in N} \hat{\beta}_{ij}^\ell \tilde{q}_\ell^s}.$$

If producers that creates counterflows are credited accordingly, we get an allocation where producers in node k are charged the amount

$$x_k^{GDF} := \sum_{i<j} c_{ij} \frac{\beta_{ij}^k \tilde{q}_k^s}{\sum_{\ell \in N} \beta_{ij}^\ell \tilde{q}_\ell^s}.$$

Distribution factors for consumers (load) are defined in a similar manner, i.e., using the equation system[17]

$$\tilde{q}_{ij} = \sum_{k \in N} \gamma_{ij}^k \tilde{q}_k^d \qquad 1 \leq i \leq j \leq n, \tag{7.31}$$

$$\gamma_{ij}^k = \gamma_{ij}^r + \alpha_{ij}^{rk} \qquad 1 \leq i \leq j \leq n \qquad k \in N \setminus \{r\}. \tag{7.32}$$

The equation system (7.32) says that withdrawing one unit of power in node k, thereby creating the flow γ_{ij}^k over line (i, j), is equivalent to

(i) withdrawing one unit of power in node r (γ_{ij}^r),

(ii) *and* injecting one unit of power in node r, to be withdrawn in node k (α_{ij}^{rk}).

In the case where consumers are not given credit for flows going in the opposite direction of the observed line flow, we use the modified distribution factors

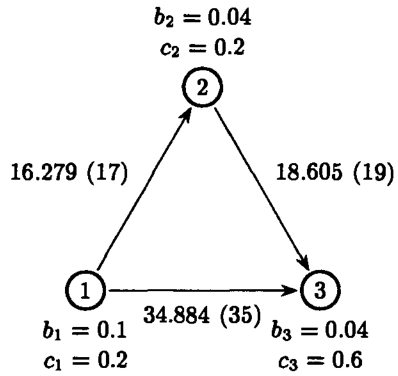
$$\hat{\gamma}_{ij}^k := \begin{cases} \gamma_{ij}^k & \gamma_{ij}^k \tilde{q}_{ij} > 0, \\ 0 & \text{otherwise.} \end{cases}$$

The total amount to be paid by consumers in node k will then be

$$x_k^{LDF+} := \sum_{i<j} c_{ij} \frac{\hat{\gamma}_{ij}^k \tilde{q}_k^d}{\sum_{\ell \in N} \hat{\gamma}_{ij}^\ell \tilde{q}_\ell^d},$$

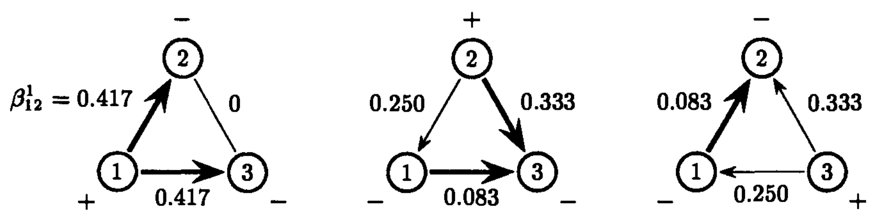
or, if credit for counterflows is given,

$$x_k^{LDF} := \sum_{i<j} c_{ij} \frac{\gamma_{ij}^k \tilde{q}_k^d}{\sum_{\ell \in N} \gamma_{ij}^\ell \tilde{q}_\ell^d}.$$

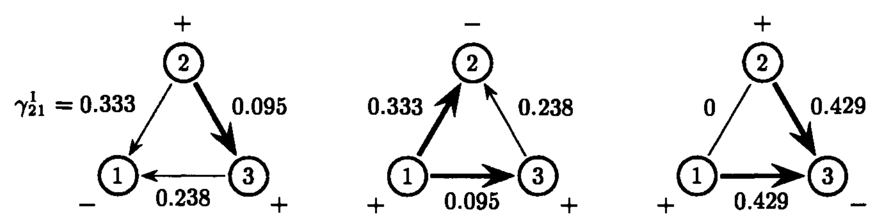


i	q_i^s	q_i^d	p_i
1	83.721	32.558	16.744
2	83.721	81.395	16.744
3	27.907	81.395	16.744

(a) Market equilibrium when all nodes use the network



(b) Generalized Generation Distribution Factors



(c) Generalized Load Distribution Factors

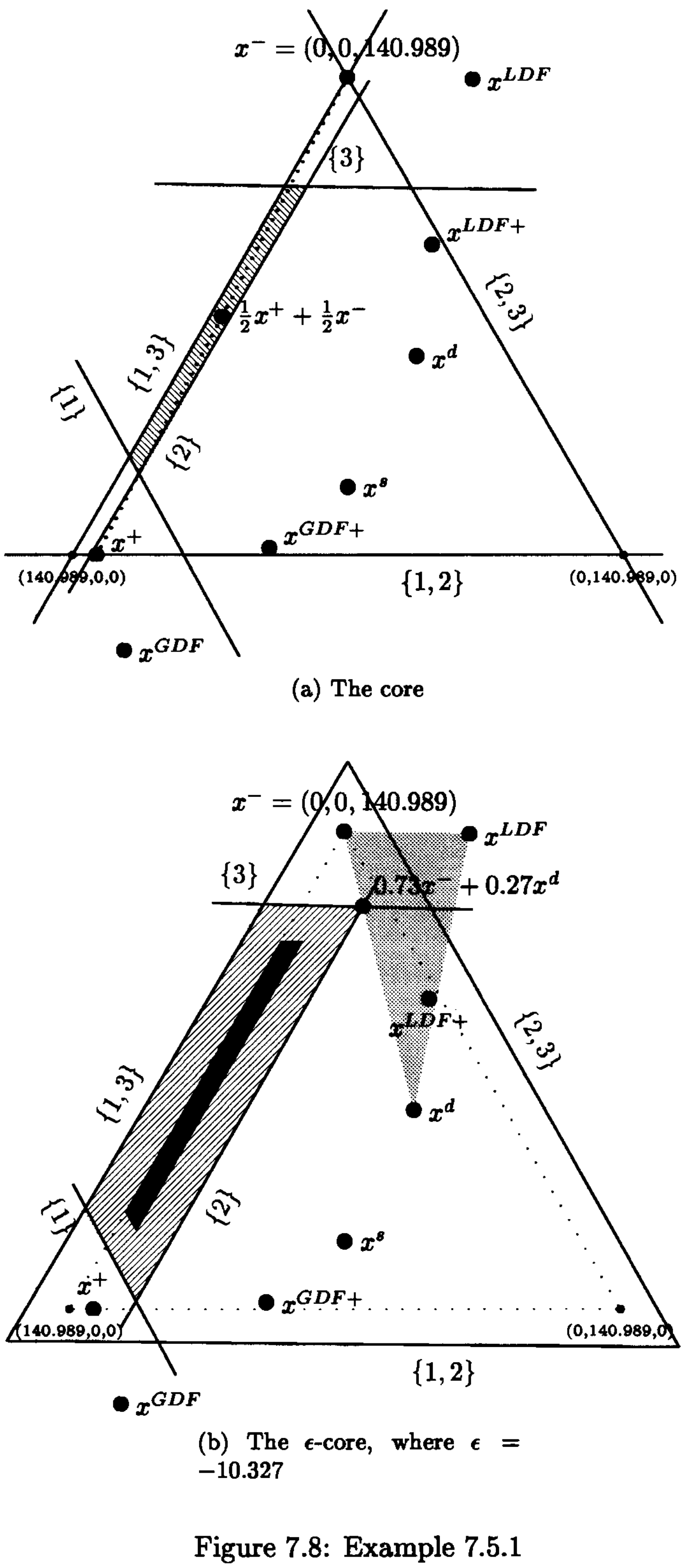
i	x_i^+	x_i^-	x_i^s	x_i^d	x_i^{GDF+}	x_i^{GDF}	x_i^{LDF+}	x_i^{LDF}
1	134.859	0.000	60.424	23.498	89.566	141.840	3.080	−31.662
2	6.130	0.000	60.424	58.745	49.313	27.091	46.394	32.406
3	0.000	140.988	20.141	58.745	2.110	−27.942	91.514	140.244

(d) Cost allocations

S			$v^C(S)$
1			112.221
	2		5.520
		3	108.476
1	2		140.989
1		3	140.989
	2	3	140.989
1	2	3	140.989

(e) The game

Figure 7.7: Example 7.5.1



(a) The core

(b) The ϵ-core, where $\epsilon = -10.327$

Figure 7.8: Example 7.5.1

Example 7.5.1 [Figures 7.7 and 7.8.] In this three-node example (see Figure 7.7(a)), production is cheapest in node 1 and 2. Different slopes (b_i) of the demand functions can be interpreted as markets of different size, where a steep demand curve corresponds to a relatively small market, hence the largest markets are in node 2 and 3. Line capacities have been set by rounding upwards the line flows corresponding to unconstrained dispatch for the grand coalition. In the cost allocation problem, we assume that $C = v^1(N)$. The line costs are assumed to be proportional to line capacities, i.e.,

$$c_{ij} := C \frac{CAP_{ij}}{\sum_{k,\ell \in N: k<\ell} CAP_{k\ell}}.$$

In a situation where all the three markets are connected via the network, node 1 (3) will have a considerable net export (import) of power. If net injection is used as a basis for cost allocation, i.e., the allocation x^+ in Figure 7.7(d), that node 1 will be paying a large share of the total cost. In net withdrawals (x^-) is used, node 3 will be paying the entire cost, since this node is the only one that has a net import of power. The use of gross production/consumption as basis for cost allocation give allocations that are less extreme.

The distribution factors β_{ij}^k, satisfying (7.29)-(7.30), are shown in Figure 7.7(b), where +'s indicate node k, i.e., the node for which production is increased. The heavy lines indicate whether β_{ij}^k have the same sign as $\tilde{q}_{ij}$. For line $(1,2)$, the contributions to the total line flow $\tilde{q}_{12} = 16.279$ from producers 1, 2, and 3, respectively, are

$$\begin{aligned}
\tilde{q}_{12}^1 &= \beta_{12}^1 \tilde{q}_1^s = 0.417 \cdot 83.721 \approx 34.884, \\
\tilde{q}_{12}^2 &= \beta_{12}^2 \tilde{q}_2^s = -0.250 \cdot 83.721 \approx -20.930, \text{ and} \\
\tilde{q}_{12}^3 &= \beta_{12}^3 \tilde{q}_3^s = 0.083 \cdot 27.907 \approx 2.326.
\end{aligned}$$

The cost of line (1,2) is

$$c_{12} = C \cdot \frac{CAP_{12}}{CAP_{12} + CAP_{13} + CAP_{23}} = 140.989 \cdot \frac{17}{17+19+35} = 33.758.$$

If no credit for counterflows is given, the producers in 1 and 3 will pay for this line alone, i.e., they will pay

$$33.758 \cdot \frac{34.884}{34.884 + 2.326} = 31.648 \text{ and } 33.758 \cdot \frac{2.326}{34.884 + 2.326} = 2.110,$$

[17] Note that $\alpha_{ij}^{rk} = -\alpha_{ij}^{kr}$.

respectively. According to x^{GDF+}, node 1 pays most of the total cost, since it is contributing heavily to the flow on line (1,2) and (1,3). Node 3 is only contributing to the flow along line (1,2), so it pays a small share of the total cost. If credit for counterflows is given, as in the allocation x^{GDF}, node 3 will in fact be paid to use the network, since it contributes negatively to flows along some lines! For allocations x^{LDF+} and x^{LDF}, corresponding to the distribution factors determined by (7.31)-(7.32) and shown in Figure 7.7(c), the situation is reversed. The consumers in node 3, which has the largest net import of power, pays most of the total cost.

The game v^C is shown in Figure 7.7(c), and the core is illustrated in Figure 7.8(a) (the hatched area). We see that none of the eight cost allocations belong to the core. However, by combining some of them, core points may be obtained. E.g. the allocation

$$\frac{1}{2}x^+ + \frac{1}{2}x^-$$

belongs to the core. It is easily seen from Figure 7.8 that, in order to obtain core allocations, we need to combine x^- with x^+ and/or x^{GDF}. ◁

7.6 Composite methods

Even though, as in Example 7.6.1, allocation methods based on a single measure (e.g. net injections) does not give a core allocation, it may be possible to find allocations that are in, or at least closer to, the core by allowing combinations of measures. An interesting question is then, how close to the core can we get, given the cost measures that we are considering? In Example 7.5.1, if we want to cover network costs solely by charging consumers, we may e.g. consider the four measures corresponding to the allocations x^-, x^d, x^{LDF+}, and x^{LDF}. By allowing combinations, we can obtain all cost allocations that belong to the convex hull of these four allocations, shown as the grey area in Figure 7.8(b). Since the intersection of this area and the core (the black area) is empty, it is not possible to find a core allocation using only these four measures. However, we can find allocations in the *strong*

ϵ-*core* of v^C, given by

$$C_\epsilon(v^C) = \left\{ x \in I^*(v^C) : \sum_{i \in S} x_i + \epsilon \leq v^C(S) \ \forall S \subset N \right\},$$

by making ϵ small enough. If $\epsilon = 0$, then $C_\epsilon(v^C)$ is just the usual core of v^C. Also, if $C_\epsilon(v^C) \neq \emptyset$ for some $\epsilon \geq 0$, then the core is nonempty. What is the largest value of ϵ for which

$$C_\epsilon(v^C) \cap \operatorname{conv}(x^-, x^d, x^{LDF+}, x^{LDF}) \neq \emptyset?$$

For Example 7.5.1, by setting $\epsilon := -10.327$, the intersection consists of only one point, namely

$$0.73x^- + 0.27x^d$$

I.e., 73% of the cost should be covered through a charge on net withdrawals, and 27% through a charge on gross consumption.

In order to find an "optimal" combination of various cost measures in the general case, denote the set of available measures by A, and let y^j be the allocation vector corresponding to measure $j \in A$. We seek a convex combination of the vectors in A that is in (or as close as possible to) the core of v^C, i.e., we solve the following LP-problem, which is similar to the LP-problems of Algorithm 6.2.1.

$$\begin{aligned}
&\text{maximize } \epsilon && && (7.33)\\
&\text{subject to } \epsilon \leq v^C(S) - \sum_{i \in S} x_i && \emptyset \neq S \subset N && (7.34)\\
&\qquad x_i = \sum_{j \in A} w_j y_i^j && i \in N && (7.35)\\
&\qquad \sum_{j \in A} w_j = 1 && && (7.36)\\
&\qquad w_j \geq 0 && j \in A && (7.37)
\end{aligned}$$

In the same manner as when computing the pre-nucleolus, we maximize the smallest excess value, given by (7.33) and (7.34). The only difference between (7.33)-(7.37) and problem $P(1)$ of Algorithm 6.2.1 is that the restriction

$$\sum_{i \in N} x_i = v^C(N) \qquad (7.38)$$

has been replaced by (7.35)-(7.37). Note that (7.38) is implied by (7.35)-(7.37), since each of the allocation vectors y^j satisfies $\sum_{i \in N} y_i^j = v^C(N)$. We will therefore refer to allocation vectors resulting from (7.33)-(7.37) as *restricted nucleoli*. If (7.33)-(7.37) does not have a unique solution, we fix the value of the coalition(s) corresponding to the lowest excess value and solve the problem again. In the same manner as in Algorithm 6.2.1, we continue fixing the excess value of coalitions until the problem has a unique solution.

Note that the number of constraints in (7.34) would be $2^n - 2$, a very large number even for moderate values of n. In order to overcome the computational difficulties, a solution method could use constraint generation, as described in Section 6.5.[18]

Example 7.6.1 [Figures 7.9-7.11] For all five nodes, the constant terms of the demand functions are equal to 20. The largest market is that of node 5, while the smallest market is in node 1. Production is most expensive in node 5, and least expensive in node 1. Line capacities and costs have been set in the same manner as in Example 7.5.1. We assume that the grand coalition has formed, i.e., all the nodes are currently using the network, hence the line flows and nodal quantities shown in part (b) and (c) of Figure 7.9 will be used for computing the allocations presented earlier in this section. The values of these allocations are presented in part (d) of Figure 7.9. Since node 1 and 5 in some sense are extreme cases, where node 1 has a relatively large production and node 5 a large consumption, the various allocations differ especially in the way that these two nodes are treated. If cost allocation is based on net injection, gross production, or generation distribution factors, node 1 pays a relatively large share of the total cost, while if it is based on net withdrawal, gross consumption, or load distribution factors, node 5 pays a relatively large share.

The game v^C is shown in Figure 7.10, and since there are no binding capacity

[18]Replace (7.34) by

$$\epsilon \leq v^C(S) - \sum_{i \in S} x_i \qquad S \in \Omega, \tag{7.39}$$

where Ω is a subset of the set of possible coalitions. Suppose (ϵ^*, x^*, w^*) is an optimal solution of (7.33), (7.35)-(7.37) and (7.39). In order to check whether any of the constraints

constraints for the grand coalition, we know from Theorem 7.3.1 that the core is nonempty. To see whether the allocations presented in Figure 7.9(d) belong to the core, we compute the excess values shown in Figure 7.10. It turns out that none of the computed allocations belong to the core, since

---

corresponding to coalitions not in Ω violates (7.34), we need to solve

$$\min_{S\notin\Omega\cup\{N,\emptyset\}}\left[v^C(S)-\sum_{i\in S}x_i^*\right]<\epsilon^*.$$

By the definition of v^C in (7.16), this is equivalent to solving

$$\begin{aligned}&\min_{S\notin\Omega\cup\{N,\emptyset\}}\left[\sum_{i\in N}\Pi_i(N)-\sum_{i\in N}\Pi_i(N\setminus S)-\sum_{i\in S}x_i^*\right]\\&=\sum_{i\in N}\Pi_i(N)-\max_{S\notin\Omega\cup\{N,\emptyset\}}\left[\sum_{i\in N}\Pi_i(N\setminus S)+\sum_{i\in S}x_i^*\right]\end{aligned}\tag{7.40}$$

Since

$$\Pi(N\setminus S)=\sum_{i\in N}\Pi_i(N\setminus S)+\Pi_{no}(N\setminus S)$$

where $\Pi_{no}(N\setminus S)\geq 0$, an upper bound to the maximization problem in (7.40) is given by

$$\max_{S\notin\Omega\cup\{N,\emptyset\}}\left[\Pi(N\setminus S)+\sum_{i\in S}x_i^*\right],$$

which, by (7.6)-(7.9), can be written as

$$\max_{\substack{S\notin\Omega\cup\{N,\emptyset\}\\ q^s,q^d,\theta}}\left[\sum_{i=1}^{n}f_i(q_i^s,q_i^d)+\sum_{i\in S}x_i^*\right]\tag{7.41}$$

$$\text{subject to } q_i^s-q_i^d=\sum_{j=1}^{n}B_{ij}(\theta_i-\theta_j)\qquad 1\leq i\leq n\tag{7.42}$$

$$B_{ij}(\theta_i-\theta_j)\leq CAP_{ij}\qquad 1\leq i,j\leq n\tag{7.43}$$

$$q_i^s-q_i^d=0\qquad i\in S\tag{7.44}$$

By letting s_i, $i\in N$, be binary variables indicating whether i is included in S or not, we can formulate this as a nonlinear integer problem by replacing (7.41) and (7.44) by

$$\max_{q^s,q^d,\theta}\left[\sum_{i=1}^{n}f_i(q_i^s,q_i^d)+\sum_{i\in S}s_ix_i^*\right]\tag{7.45}$$

$$-M(1-s_i)\leq q_i^s-q_i^d\leq M(1-s_i)\qquad i\in N,\tag{7.46}$$

respectively, where M is a sufficiently large number. We also add the constraints

$$\sum_{i\in S}s_i+\sum_{i\in N\setminus S}(1-s_i)\geq 1\qquad S\in\Omega\cup\{N,\emptyset\}\tag{7.47}$$

$$s_i\in\{0,1\}\qquad i\in N,\tag{7.48}$$

where constraint (7.47) eliminates coalitions already in Ω from consideration.

there is at least one negative excess value for each of them.

Some results from the procedure based on (7.33)-(7.37) are shown in Figure 7.11, where we combine some or all of the eight cost allocation methods presented in Section 7.5. For example, as shown in (a), if we restrict the attention to combinations of the four postage-stamp rate methods (x^+, x^-, x^s, x^d), it is possible to obtain the core allocation x^1, where 35.1% of the cost is recovered through a charge based on net injections (x^+), and 64.9% through a charge based on net withdrawals (x^-). Of course, if we include more candidates for combination, we get better core allocations, i.e., higher excess values (lexicographically). The allocation x^5, where combinations of all 8 methods were allowed, has the same minimal excess value as x^1 (0.557), but the second lowest excess value (5.355) is higher than the second lowest value for x^1 (0.764). Note that it is not possible to obtain core allocations solely based on distribution factors (x^2), nor by considering only production-oriented (x^3) or consumption-oriented (x^4) measures. ◁

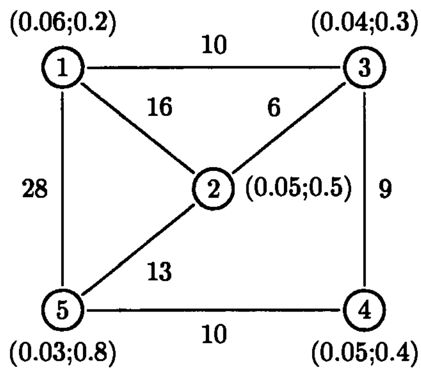
7.7 Conclusion

Our model takes into account both the behaviour of humans, i.e., the producers and consumers, and the electrons flowing through the electricity network, as described in Section 7.2. The cost allocation problem is modeled using a game-theoretic framework in Section 7.3, where the consumer and producer surpluses of the nodes are used to form an upper bound on the amount that the network owner can charge a coalition. Our game differs from other cost sharing games in that the characteristic function does not measure the stand-alone cost of coalitions. In forming the game, we also made the assumption that sidepayments can be made between *all players*, not only those forming part of a coalition. The main result of Section 7.3 is that the core is nonempty whenever the line capacities does not constrain the grand coalition. In Section 7.4 we show how the network owner may handle the problem of an empty core, either by lowering the total payments collected, or by taxing individuals that choose not to make use of the network. Section 7.5 illustrates, using a numerical example, how commonly used cost allocation methods, such as the postage-stamp rate methods, may fail to yield core elements. However, by combining several methods, core points may be obtained, as is shown in Section 7.6. An LP-based procedure for computing core points, given that the points must be a combination of existing cost allocation methods, is presented.

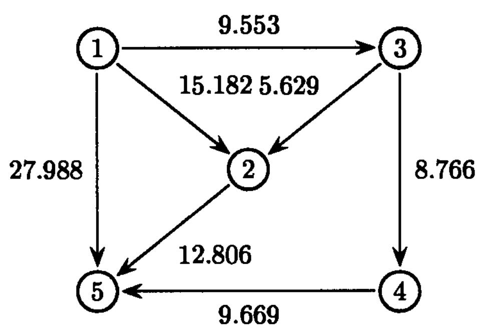
The model could be made more realistic by introducing uncertainty, e.g., with respect to demand variations or line failures. Also, the optimization problem (7.6)-(7.8) may not be a realistic description of the market for electricity. E.g., in the Norwegian spot market[19], a zonal pricing system is used, where a sub-optimal solution of (7.6)-(7.8) is chosen.

Problem instances of a more realistic size would be difficult to handle computationally, and the LP-based procedure described in Section 7.6 could be revised by introducing constraint generation (see footnote 18 on page 165).

[19]See Bjørndal and Jörnsten (2001).



(a) Line capacities and demand/cost parameters $(b_i; c_i)$.



(b) Line flows for the grand coalition.

i	q_i^s	q_i^d	p_i
1	89.090	36.368	17.818
2	35.636	43.641	17.818
3	59.393	54.551	17.818
4	44.545	43.641	17.818
5	22.272	72.735	17.818

(c) Nodal equilibrium quantities for the grand coalition.

i	x_i^+	x_i^-	x_i^s	x_i^d	x_i^{GDF+}	x_i^{GDF}	x_i^{LDF+}	x_i^{LDF}
1	92.743	0.000	36.515	14.906	57.716	92.602	2.146	−22.895
2	0.000	14.082	14.606	17.887	11.059	−4.847	18.503	23.823
3	8.517	0.000	24.343	22.359	22.910	39.010	11.049	−13.471
4	1.590	0.000	18.257	17.887	10.361	−4.550	21.695	22.345
5	0.000	88.768	9.129	29.812	0.804	−19.364	49.457	93.049

(d) Postage-stamp rate allocations and allocations based on distribution factors.

Figure 7.9: Example 7.6.1

S					v^C	$e(x^+)$	$e(x^-)$	$e(x^s)$	$e(x^d)$	$e(x^{GDF+})$	$e(x^{GDF})$	$e(x^{LDF+})$	$e(x^{LDF})$
1					86.103	-6.640	86.103	49.589	71.198	28.388	-6.498	83.958	108.999
	2				24.868	24.868	10.787	10.262	6.981	13.810	29.716	6.365	1.045
		3			10.711	2.193	10.711	-13.633	-11.648	-12.199	-28.299	-0.338	24.182
			4		1.115	-0.475	1.115	-17.142	-16.772	-9.246	5.665	-20.580	-21.229
				5	65.571	65.571	-23.198	56.442	35.759	64.767	84.935	16.114	-27.478
1	2				90.971	-1.772	76.890	39.851	58.179	22.197	3.217	70.322	90.043
1		3			90.768	-10.492	90.768	29.910	53.504	10.143	**-40.843**	77.574	127.135
	2	3			27.225	18.708	13.143	-11.724	-13.021	-6.744	-6.938	-2.327	16.872
1			4		81.664	**-12.669**	81.664	26.892	48.871	13.587	-6.388	57.823	82.215
	2		4		25.039	23.449	10.957	-7.824	-10.735	3.619	34.436	-15.159	-21.129
		3	4		21.628	11.520	21.628	-20.973	-18.618	-11.644	-12.832	-11.116	12.754
1				5	100.997	8.254	12.229	55.353	56.279	42.477	27.759	49.394	30.844
	2			5	84.041	84.041	-18.809	60.306	36.342	72.178	108.252	16.080	-32.831
		3		5	61.397	52.879	**-27.372**	27.925	9.226	37.682	41.751	0.890	-18.181
			4	5	66.003	64.413	-22.766	38.617	18.304	54.837	89.917	-5.150	-49.390
1	2	3			94.597	-6.663	80.515	19.133	39.446	2.913	-32.167	62.899	107.140
1	2		4		92.448	-1.885	78.366	23.070	41.768	13.312	9.244	50.104	69.176
1		3	4		94.785	-8.065	94.785	15.669	39.633	3.798	-32.277	59.895	108.806
	2	3	4		28.283	18.176	14.201	**-28.923**	**-29.850**	**-16.047**	-1.329	**-22.964**	-4.414
1	2			5	102.743	10.000	-0.107	42.493	40.139	33.164	34.353	32.637	8.766
1		3		5	101.942	0.682	13.174	31.955	34.866	20.512	-10.305	39.290	45.260
	2	3		5	88.173	79.656	-14.677	40.095	18.116	53.400	73.375	9.164	-15.228
1			4	5	101.079	6.746	12.311	37.178	38.475	32.198	32.392	27.781	8.582
	2		4	5	88.970	87.380	-13.880	46.978	23.385	66.746	117.732	-0.685	**-50.246**
		3	4	5	76.467	66.360	-12.301	24.738	6.410	42.391	61.371	-5.734	-25.455
1	2	3	4		102.850	0.000	88.768	9.129	29.812	0.804	-19.364	49.457	93.049
1	2	3		5	102.850	1.590	-0.000	18.257	17.887	10.361	-4.550	21.695	22.345
1	2		4	5	102.850	8.517	0.000	24.343	22.359	22.910	39.010	11.049	-13.471
1		3	4	5	102.850	0.000	14.082	14.606	17.887	11.059	-4.847	18.503	23.823
	2	3	4	5	102.850	92.743	0.000	36.515	14.906	57.716	92.602	2.146	-22.895
1	2	3	4	5	102.850	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Figure 7.10: Excess values for various allocation vectors in Example 7.6.1

Candidates	Solution
x^+, x^-, x^s, x^d	$x^1 = 0.351x^+ + 0.649x^-$
x^{GDF+}, x^{GDF}, x^{LDF+}, x^{LDF}	$x^2 = 0.572x^{GDF} + 0.428x^{LDF}$
x^+, x^s, x^{GDF+}, x^{GDF}	$x^3 = 0.686x^+ + 0.056x^s + 0.258x^{GDF+}$
x^-, x^d, x^{LDF+}, x^{LDF}	$x^4 = 0.485x^- + 0.515x^d$
all 8	$x^5 = 0.383x^+ + 0.557x^- + 0.015x^{GDF+} + 0.045x^{GDF}$

(a)

i	$nucl_i$	x_i^1	x_i^2	x_i^3	x_i^4	x_i^5
1	53.959	32.517	43.181	80.549	7.684	40.546
2	11.110	9.145	7.421	3.674	16.043	7.665
3	5.355	2.986	16.554	13.121	11.526	10.153
4	0.557	0.557	6.958	4.788	9.220	0.557
5	31.868	57.645	28.736	0.718	58.377	43.929

(b)

S					$e(nucl)$	$e(x^1)$	$e(x^2)$	$e(x^3)$	$e(x^4)$	$e(x^5)$
1					32.144	53.587	42.922	5.554	78.420	45.558
	2				13.758	15.724	17.448	21.194	8.825	17.077
		3			5.355	7.724	**-5.843**	-2.410	-0.815	5.355
			4		**0.557**	**0.557**	**-5.843**	**-3.673**	-8.106	**0.557**
				5	33.702	7.925	36.835	64.853	7.194	16.970
1	2				25.902	49.311	40.369	6.748	67.244	42.634
1		3			31.454	55.266	31.033	-2.902	71.559	44.867
	2	3			10.759	15.094	3.251	10.430	-0.344	14.078
1			4		27.147	48.590	31.524	**-3.673**	64.760	40.561
	2		4		13.372	15.337	10.661	16.577	-0.225	16.690
		3	4		15.715	18.084	-1.884	3.719	0.881	15.715
1				5	15.170	10.835	29.079	19.730	34.936	11.851
	2			5	41.063	17.250	47.884	79.649	9.621	27.649
		3		5	24.173	0.764	16.106	47.557	**-8.506**	7.441
			4	5	33.577	7.799	30.308	60.496	-1.595	16.845
1	2	3			24.173	49.950	27.441	-2.747	59.344	40.905
1	2		4		26.822	50.230	34.888	3.437	59.501	43.554
1		3	4		34.913	58.725	28.091	-3.673	66.355	48.326
	2	3	4		11.260	15.595	-2.649	6.700	**-8.506**	14.579
1	2			5	5.806	3.436	23.405	17.802	20.639	5.806
1		3		5	10.759	8.794	13.470	7.554	24.356	7.441
	2	3		5	39.839	18.396	35.463	70.660	2.227	26.426
1			4	5	14.695	10.360	22.204	15.024	25.798	11.376
	2		4	5	45.434	21.622	45.855	79.790	5.329	32.021
		3	4	5	38.686	15.277	24.219	57.840	-2.656	21.954
1	2	3	4		31.868	57.646	28.736	0.718	58.377	48.600
1	2	3		5	**0.557**	**0.557**	6.958	4.788	9.220	**0.557**
1	2		4	5	5.355	2.986	16.554	13.121	11.526	5.355
1		3	4	5	11.110	9.145	7.421	3.674	16.043	7.791
	2	3	4	5	53.959	32.516	43.182	80.549	7.684	40.546

(c)

Figure 7.11: The nucleolus and restricted nucleoli for Example 7.6.1

Bibliography

Aadland, D. and Kolpin, V. (1998). Shared irrigation cost: an empirical and axiomatical analysis. *Mathematical Social Sciences*, 849:203–218.

Adams, W. and Sherali, H. (1990). Linearization strategies for a class of zero-one mixed integer programming problems. *Operations Research*, 38(2):217–226.

Aumann, R. and Maschler, M. (1985). Game theoretic analysis of a bankruptcy problem from the Talmud. *Journal of Economic Theory*, 36:195–213.

Bergen, A. and Vittal, V. (1986). *Power Systems Analysis.* Prentice Hall, 2nd edition.

Bjørndal, E., Hamers, H., and Koster, M. (1999a). Cost allocation in a bank ATM network. Mimeo, Tilburg University, Tilburg, The Netherlands.

Bjørndal, E., Koster, M., and Tijs, S. H. (1999b). Weighted allocation rules for standard fixed tree games. CentER discussion paper 9979, Tilburg University, Tilburg, The Netherlands.

Bjørndal, M. and Jörnsten, K. (2001). Zonal pricing in a deregulated electricity market. *Energy Journal*, 22/1:51–73.

Bondareva, O. (1963). Some applications of linear programming methods to the theory of cooperative games (in Russian). *Problemy Kibernetici*, 10:119–139.

Brown, J. (1979). The sharing problem. *Operations Research*, 27(2):324–340.

Chardaire, P. (2001). The core and nucleolus of games: A note on a paper by Göthe-Lundgren et al. *Mathematical Programming*, A90:147–151.

Chun, Y. (1999). Equivalence of axioms for bankruptcy problems. *International Journal of Game Theory*, 28:511–520.

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

Curiel, I., Maschler, M., and Tijs, S. H. (1987). Bankruptcy games. *Mathematical Methods of Operations Research*, 31:143–159.

Dagan, N. (1996). New characterizations of old bankruptcy rules. *Social Choice and Welfare*, 13:51–59.

Dagan, N., Serrano, R., and Volij, O. (1997). A noncooperative view of consistent bankruptcy rules. *Games and Economic Behavior*, 18:55–72.

Dagan, N. and Volij, O. (1997). Bilateral comparisons and consistent fair division rules in the context of bankruptcy problems. *International Journal of Game Theory*, 26:11–25.

Davis, M. and Maschler, M. (1965). The kernel of a cooperative game. Working paper, Princeton University.

Driessen, T. (1991). A survey of consistency properties in cooperative game theory. *SIAM Review*, 33(1):43–59.

Dubey, P. (1982). The Shapley value as aircraft landing fees revisited. *Management Science*, 28(8):869–874.

Dutta, B. (1990). The egalitarian solution and reduced game properties in convex games. *International Journal of Game Theory*, 19:153–169.

Dutta, B. and Ray, D. (1989). A concept of egalitarianism under participation constraints. *Econometrica*, 57:615–635.

Ebert, U. (1999). Using equivalent income of equivalent adults to rank income distributions. *Social Choice and Welfare*, 16:233–258.

Fujishige, S. (1980). Lexicographically optimal base of polymatroid with respect to a weight vector. *Mathematics of Operations Research*, 5(2):186–196.

Gellekom, J., Potters, J., Reijnierse, J., Tijs, S., and Engel, M. (2000). Characterization of the Owen set of linear production processes. *Games and Economic Behavior*, 32/1:139–156.

Gilmore, P. and Gomory, R. (1962). A linear programming approach to the cutting-stock problem. *Operations Research*, 9:849–859.

Glover, F. (1975). Improved linear integer programming formulations of nonlinear integer problems. *Management Science*, 22(4):455–460.

Göthe-Lundgren, M., Jörnsten, K., and Värbrand, P. (1996). On the nucleolus of the basic vehicle routing game. *Mathematical Programming*, 72:83–100.

Gow, S. and Thomas, L. (1998). Interchange fees for bank ATM networks. *Naval Research Logistics*, 45:407–417.

Granot, D. (1986). A generalized linear production model: A unifying model. *Mathematical Programming*, 34:212–222.

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

Granot, D., Maschler, M., Owen, G., and Zhu, W. (1996). The kernel /nucleolus of a standard fixed tree game. *International Journal of Game Theory*, 25:219–244.

Herrero, C., Maschler, M., and Villar, A. (1999). Individual rights and collective responsibility: The rights-egalitarian solution. *Mathematical Social Sciences*, 37:59–77.

Hokari, T. (1998). Weighted Dutta-Ray solutions on convex games. *mimeo*, University of Rochester, Rochester.

Hokari, T. and Thomson, W. (2000). Bankruptcy and weighted generalizations of the talmud rule. Mimeo, University of Rochester, Rochester, U.S.

Huberman, G. (1980). The nucleolus and the essential coalitions. *Lecture notes in control and information sciences (Berling, Springer)*, 28:416–422.

Ichimori, T., Ishii, H., and Nishida, T. (1982). Optimal sharing. *Mathematical Programming*, 23(3):341–348.

Kalai, E. and Samet, D. (1987). On weighted Shapley values. *International Journal of Game Theory*, 16:205–222.

Kalai, E. and Samet, D. (1988). Weighted Shapley values. In Roth, A. E., editor, *The Shapley Value: Essays in Honour of Lloyd S. Shapley*. Cambridge University Press, New York.

Kohlberg, E. (1971). On the nucleolus of a characteristic function game. *SIAM Journal of Applied Mathematics*, 20:62–66.

Kopelowitz, A. (1967). Computation of the kernels of simple games and the nucleolus of n-person games. RM-131, Mathematics Department, The Hebrew University of Jerusalem, Israel.

Koster, M. (1999). Weighted constrained egalitarianism in TU-games. CentER Discussion Paper 99107, Tilburg University, Tilburg, The Netherlands.

Koster, M., Molina, E., Sprumont, Y., and Tijs, S. H. (1998). Sharing the cost of a network: core and core allocations. CentER Discussion Paper 9821, Tilburg University, Tilburg, The Netherlands.

Laporte, G. (1998). Recent advances in routing algorithms. In Labbé, M., Laporte, G., Tanczos, K., and Toint, P., editors, *Operations Research Decision Aid Methodologies in Traffic and Transportation Management*, pages 183–195. Springer.

Littlechild, S. (1974). A simple expression for the nucleolus in a special case. *International Journal of Game Theory*, 3:21–30.

Littlechild, S. and Owen, G. (1977). A further note on the nucleolus of the airport game. *International Journal of Game Theory*, 3:21–29.

Littlechild, S. and Thompson, G. (1977). Aircraft landing fees: a game theory approach. *The Bell Journal of Economics*, 8:186–204.

Marangon Lima, J. (1996). Allocation of transmission fixed charges: An overview. *IEEE Transactions on Power Systems*, 11(3):1409–1418.

Maschler, M., Peleg, B., and Shapley, L. (1979). Geometric properties of the kernel, nucleolus and related solution concepts. *Mathematics of Operations Research*, 4:303–338.

Maschler, M., Potters, J., and Reijnierse, H. (1995). Monotonicity properties of the nucleolus of standard tree games. Report no. 9556, Department of Mathematics, University of Nijmegen, Nijmegen, The Netherlands.

Megiddo, N. (1974). Optimal flows in networks with multiple sources and sinks. *Mathematical Programming*, 7:97–107.

Megiddo, N. (1977). A good algorithm for lexicographically optimal flows in multi-terminal networks. *Bulletin of the American Mathematical Society*, 83:407–409.

Megiddo, N. (1978). Computational complexity of the game theory approach to cost allocation for a tree. *Mathematics of Operations Research*, 3:189–196.

Monderer, D., Samet, D., and Shapley, L. S. (1992). Weighted values and the core. *International Journal of Game Theory*, 21:27–39.

Moulin, H. (1988). *Axioms of cooperative decision making*, volume 15 of *Econometric Society Monographs*. Cambridge University Press, Cambridge, New York, U. S.

Muto, S., Nakayama, M., Potters, J., and Tijs, S. (1988). On big boss games. *The Economic Studies Quarterly*, 39(4):303–321.

Muto, S., Potters, J., and Tijs, S. (1989). Information market games. *International Journal of Game Theory*, 18:209–226.

Ng, W. (1981). Generalized generation distribution factors for power system security evalutions. *IEEE Transactions on Power Apparatus and Systems*, PAS-100(3):1001–1005.

Nouweland, A., Borm, P., Golstein Brouwers, W., Groot Bruinderink, R., and Tijs, S. (1996). A game theoretic approach to problems in telecommunications. *Management Science*, 42(2):294–303.

O'Neill, B. (1982). A problem of rights arbitration from the Talmud. *Mathematical Social Sciences*, 2:345–371.

Owen, G. (1975). On the core of linear production games. *Mathematical Programming*, 9:358–370.

Pan, J., Teklu, Y., Rahman, S., and Jun, K. (2000). Review of usage-based transmission cost allocation methods under open access. *IEEE Transactions on Power Systems*, 15(4):1218–1224.

Peleg, B. (1986). On the reduced game property and its converse. *International Journal of Game Theory*, 15:187–200.

Pérez-Arriga, I., Rubio, F., Puerta, J., Arceluz, J., and Marín, J. (1995). Marginal pricing of transmission services: An analysis of costs recovery. *IEEE Transactions on Power Systems*, 10(1):546–553.

Petersen, C. (1971). A note on transforming the product of variables to linear form in linear programs. *Working paper*, Purdue University.

Potters, J. and Sudhölter, P. (1999). Airport problems and consistent allocation rules. *Mathematical Social Sciences*, 38:83–102.

Potters, J. and Tijs, S. (1989). Information market games with more than one informed player. *Methods of Operations Research*, 63:313–324.

Rudnick, H., Palma, R., and Fernandez, J. (1995). Marginal pricing and supplement cost allocation in transmission open access. *IEEE Transactions on Power Systems*, 10(2):1125–1142.

Schmeidler, D. (1969). The nucleolus of a characteristic function game. *SIAM Journal of Applied Mathematics*, 17:1163–1179.

Shapley, L. (1953). A value for n-person games. In Kuhn, W. and Tucker, A., editors, *Contributions to the Theory of Games II*, pages 307–317, Princeton. Princeton University Press.

Shapley, L. (1971). Cores of convex games. *International Journal of Game Theory*, 1:11–26.

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

Shirmohammadi, D., Gribik, P., Law, E., Malinowski, J., and O'Donnell, R. (1989). Evaluation of transmission network capacity use for wheeling transactions. *IEEE Transactions on Power Systems*, 4(4):1405–1413.

Shirmohammadi, D., Vieira Filho, X., Gorenstin, B., and Pereira, M. (1996). Some fundamental technical concepts about cost based transmission pricing. *IEEE Transactions on Power Systems*, 11(2):1002–1008.

Sprumont, Y. (1990). Population monotonic allocation schemes for cooperative games with transferable utility. *Games and Economic Behavior*, 2:378–394.

Statnett (2000). Tariffheftet 2000. Available at www.statnett.no.

Statnett (2002). Sentralnettariffen 2002. Available at www.statnett.no.

Stearns, R. (1968). Convergent transfer schemes for n-person games. *Trans. Amer. Math. Soc.*, 134:449–459.

Tijs, S. (1981). Bounds for the core of a game and the τ-value. In Moeschlin, O. and Pallaschke, D., editors, *Game Theory and Mathematical Economics*, pages 123–132. North-Holland Publ. Comp., Amsterdam.

Tijs, S. H. and Driessen, T. (1986). Game theory and cost allocation problems. *Management Science*, 32:1015–1028.

Wright, S. (1997). *Primal-Dual Interior-Point Methods.* SIAM.

Wu, F., Varaiya, P., Spiller, P., and Oren, S. (1996). Folk theorems on transmission access:proofs and counterexamples. *Journal of Regulatory Economics*, 10:5–23.

Yanovskaya, E. (1992). Set-valued analogues of the prenucleolus for cooperative TU-games. Mimeo, Russian Academy of Sciences, St. Petersburg, Russia.

Zipkin, P. (1980a). Bounds for row-aggregation in linear programming. *Operations Research*, 28(4):903–916.

Zipkin, P. (1980b). Bounds on the effect of aggregating variables in linear programs. *Operations Research*, 28(2):403–418.