The London School of Economics and Political Science

Cost allocation in connection and conflict problems on networks: A cooperative game theoretic approach

Nayat Horozoglu

A thesis submitted to the Department of Management of
the London School of Economics and Political Science
for the degree of Doctor of Philosophy in Operational Research

London, 2012

Declaration

I certify that the thesis I have presented for examination for the PhD degree of the London School of Economics and Political Science is solely my own work other than where I have clearly indicated that it is the work of others (in which case the extent of any work carried out jointly by me and any other person is clearly identified in it).

I warrant that this authorisation does not, to the best of my belief, infringe the rights of any third party.

Abstract

This thesis examines settings where multiple decision makers with conflicting interests benefit from cooperation in joint combinatorial optimisation problems. It draws on cooperative game theory, polyhedral theory and graph theory to address cost sharing in joint single-source shortest path problems and joint weighted minimum colouring problems.

The primary focus of the thesis are problems where each agent corresponds to a vertex of an undirected complete graph, in which a special vertex represents the common supplier. The joint combinatorial optimisation problem consists of determining the shortest paths from the supplier to all other vertices in the graph. The optimal solution is a shortest path tree of the graph and the aim is to allocate the cost of this shortest path tree amongst the agents. The thesis defines shortest path tree problems, proposes allocation rules and analyses the properties of these allocation rules. It furthermore introduces shortest path tree games and studies the properties of these games. Various core allocations for shortest path tree games are introduced and polyhedral properties of the core are studied. Moreover, computational results on finding the core and the nucleolus of shortest path tree games for the application of cost allocation in Wireless Multihop Networks are presented.

The secondary focus of the thesis are problems where each agent is interested in having access to a number of facilities but can be in conflict with other agents. If two agents are in conflict, then they should have access to disjoint sets of facilities. The aim is to allocate the cost of the minimum number of facilities required by the agents amongst them. The thesis models these cost allocation problems as a class of cooperative games called weighted minimum colouring games, and characterises total balancedness and submodularity of this class of games using the properties of the underlying graph.

Money,
It’s a crime.
Share it fairly,
But don’t take a slice of my pie.

Money, Pink Floyd (*Dark Side of the Moon*, 1973)

Acknowledgements

I would like to thank a number of people who have helped me along my journey towards obtaining my PhD. First of all, I am very grateful to Prof. Appa for encouraging me to undertake a PhD. His enthusiasm for research has always been an inspiration. I would also like to thank Katerina for supervising my thesis. She was very supportive and helpful throughout my doctoral work. Over the years she has become a friend as well as a supervisor, and I hope that she will always remember me as her first PhD student. I am furthermore very grateful to my examiners Prof. Williams and Vangelis for their constructive comments. With their help, my efforts and work during my PhD are much better reflected in my thesis.

In 2009, I attended a conference in Chicago with little hope of generating interest in my research. However, Herbert approached me after my presentation and invited me to come to Tilburg University. He told me that, for cooperative game theory, it was “the place to be”. He was right. I had some of the most fruitful days of my PhD in this Dutch town with the invaluable help and support of Herbert. Later on, I was lucky enough to meet Henk, who, together with Herbert, formed my very own “dynamic duo” of cooperative games.

I would also thank my “LSE OR family”, starting with mother Brenda who always took care of me and made my life as a PhD student much easier. Thanks also go to my older brothers Kai, Kostas and Srini who shared their experience and wisdom with me, and to my younger siblings Anastasia and Dimitris who cheered me up when I needed it the most.

My family and friends have also been of great support during my PhD. In particular, I am deeply indebted to my parents for their boundless love and belief, to my grandmother who raised me to be the person I am today and to my aunt who taught me to stay positive no matter what life throws at me.

Finally, I would like to thank Florian for motivating me to go on, for entertaining me when I was down and for being by my side every step along the way. I really would not have made it without you.

Contents

Chapter 1

Introduction

This thesis addresses cost allocation in situations where there are multiple decision makers who are motivated to cooperate upon solving a joint combinatorial optimisation problem in order to realise cost savings. The primary focus of the thesis is a connection problem, namely the shortest path tree problem, and the corresponding games. The secondary focus of the thesis are games arising from a conflict problem, namely the weighted minimum colouring games.

Combinatorial optimisation deals with problems in which a single decision maker searches for an optimal solution in a discrete set of potential solutions where optimality is defined with respect to the desired objective of the decision maker. There are a variety of practical applications such as ship routing, machine sequencing and airline crew scheduling that can be modelled as combinatorial optimisation problems. Although the optimal solution of these problems is a subset of a finite set and thus in theory can be identified by complete enumeration (Wolsey, 1998), the large number of possible solutions for such practical applications makes complete enumeration intractable. Therefore, a considerable amount of research on combinatorial optimisation concentrates on developing fast and efficient algorithms for solving different classes of combinatorial optimisation problems.

For combinatorial optimisation problems involving several decision makers who are motivated to work together in order to reduce their overall costs (to increase their overall profits), the theory of cooperative games provides valuable insights for cost (profit) allocation. Cooperative game theory concentrates on games with multiple decision makers (players) who are assumed to be able to form subsets (coalitions) and addresses the allocation of the total cost (profit) of a joint project if the players involved work together to reduce costs (increase profits). The players use different coalitions as a basis of negotiation for their share of the cost (profit) of the joint project. In cases where multiple decision makers with conflicting interests benefit from cooperation upon solving their joint combinatorial optimisation problem, the solution process can be seen as consisting of two sequential subprocesses. Firstly, combinatorial optimisation techniques enable the decision makers to solve their common problem to optimality. Next, making use of cooperative game theoretical concepts, a “mutually satisfactory” or “fair” allocation can potentially be reached.

In this thesis we employ two approaches from the cooperative game theory literature to address cost allocation arising from combinatorial optimisation problems. Firstly, we consider existing, well-known game theoretical allocation concepts and introduce new allocation rules that arise from the particular setting that is of interest to us. Secondly, we investigate the properties of the cooperative games that we define.

1.1 Literature Review

The following paragraphs will introduce four games arising from combinatorial optimisation problems to help position this thesis within the existing literature on the interplay between cooperative game theory and combinatorial optimisation. Firstly, we concentrate on two games defined on trees, namely minimum cost spanning tree and maintenance games, and in the context of these games, we define and illustrate cooperative game theoretical properties and solution concepts that this thesis will draw upon. We furthermore present Chinese postman games and highway games where cooperative game theoretical properties are studied in relation to the properties of the underlying graph representing the combinatorial optimisation problem. Various other examples of cooperative games arising from combinatorial optimisation problems that are not discussed below (e.g. travelling salesman games, sequencing games, assignment games) can be found in the comprehensive survey on games induced by operational research problems by Borm et al. (2001). Furthermore, cooperative games arising from combinatorial optimisation problems constitute the subject of the book by Curiel (1997). A wider class of cost allocation problems and “fairness” in cost allocation in general are discussed in Young (1985).

Example 1.1. ***Minimum Cost Spanning Tree Games***. Consider the following situation arising from the minimum cost spanning tree problem displayed in Figure 1.1. A *spanning tree* of a graph G is a connected and acyclic subgraph of G containing all vertices of G. A *minimum cost spanning tree* (MCST) of G is a spanning tree with the property that the sum of the costs of the edges of the spanning tree is minimum.

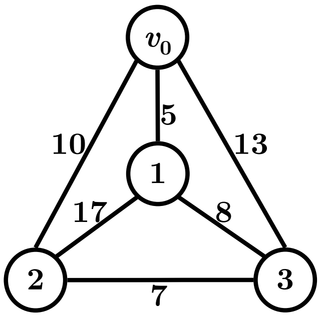


Figure 1.1: An MCST problem.

Let v_0 represent a water supplier and let the vertices 1, 2 and 3 represent three villages that need to be connected to the water supplier. The edge costs represent (in £millions) the cost of building a pipeline between two villages or between a village and the water supplier. The villages are to cover the cost of the construction of the water distribution system.

If the three villages choose to act on their own, they would have to pay £5m, £10m and £13m, respectively, for connecting to the water supplier. If, however, they all decide to cooperate to build a water distribution system, the pipelines go from the water supplier to village 1, from village 1 to village 3, and finally from village 3 to village 2 resulting in a total cost of £20m. In fact, this solution is an MCST of the complete graph induced by vertices v_0, 1, 2 and 3. The combinatorial optimisation problem of determining an MCST of a graph can be solved efficiently by the algorithms proposed by Kruskal (1956) and Prim (1957). Observe that when the villages cooperate, they can save (£5m + £10m + £13m)-£20m = £8m. Now, the question is how to allocate the total cost of £20m fairly amongst the three villages. Would dividing the total cost by 3 and allocating £6.66m to each village be considered a good allocation? Or is it better to allocate to each village the cost of the pipeline that connects them to the water distribution system and thus to assign £5m, £7m and £8m to villages 1, 2 and 3, respectively? Naturally, there are numerous possible ways of defining such allocations. As pointed out by Tijs and Driessen (1986), the choice of the "best" allocation is a contextual issue depending on the setting of the joint problem or on the perspectives of the decision makers as to what is deemed "fair" or "satisfactory". Therefore, a central aspect is to identify desirable properties that can then be used to compare and contrast different allocations. The literature on such properties is reviewed in Thomson (2007).

The cost allocation problem associated with the MCSTs was introduced by Claus and Kleitman (1973). From the numerous studies on cost allocation rules, the properties as well as the axiomatic characterisations of these rules for MCST problems, we mention Branzei et al. (2004), Moretti et al. (2005), Tijs et al. (2006), Bergantiños and Vidal-Puga (2007) and Bergantiños and Kar (2010), as well as Feltkamp et al. (1994a) and Feltkamp et al. (1994b) for generalisations of MCST problems.

The game theoretic approach to the allocation of cost in MCST problems was initiated by Bird (1976) and further analysed by Granot and Huberman (1981) and Granot and Huberman (1984).

A *cooperative (cost) game* is a pair (N, c) where $N = \{1, 2, \dots, n\}$ is the finite set of players and $c : 2^N \to \mathbb{R}$ is the *characteristic function* such that $c(\emptyset) = 0$. Here 2^N is the collection of all subsets of N (also referred to as coalitions). The MCST game (N, c), arising from the MCST problem in Figure 1.1, consists of the player set $N = \{1, 2, 3\}$ and the characteristic function c displayed in Table 1.1.

S	$\{1\}$	$\{2\}$	$\{3\}$	$\{1,2\}$	$\{1,3\}$	$\{2,3\}$	$\{1,2,3\}$
$c(S)$	5	10	13	15	13	17	20

Table 1.1: Coalitional costs of the MCST game (N, c) in £millions.

Consider the allocation of £6.66m to each village. Village 1 would not be satisfied with this allocation since if village 1 were to build a pipeline on its own, it would only have to pay £5m. Therefore, the allocation (6.66, 6.66, 6.66) does not provide incentive to village 1 to cooperate with the rest of the villages. In fact, every subset of villages can employ this perspective and use the cost of building a pipeline amongst themselves as a benchmark. Then an allocation vector $x = (x_1, x_2, x_3)$ where x_i denotes the cost allocated to player i by x, that guarantees that the exact cost of the water distribution system is paid by the three villages such that no subset of villages pay more than what they would have paid if they acted on their own must satisfy the following constraints:

$$x_1 + x_2 + x_3 = 20$$

$$
\begin{aligned}
x_1 + x_2 \quad\quad &\leq 15 \\
x_1 \quad\quad + x_3 &\leq 13 \\
x_2 + x_3 &\leq 17 \\
x_1 \quad\quad\quad\quad &\leq 5 \\
x_2 \quad\quad &\leq 10 \\
x_3 &\leq 13.
\end{aligned}
$$

The set of all solutions satisfying these constraints is called the *core* of (N, c) (Gillies, 1953) and is defined as

$$Core(c) = \left\{ x \in \mathbb{R}^N | \sum_{i \in N} x_i = c(N) \text{ and } \sum_{i \in S} x_i \leq c(S) \text{ for all } S \subset N \right\}.$$

Note that if an allocation $x \in \mathbb{R}^N$ satisfies $\sum_{i \in N} x_i = c(N)$, then it is called *efficient*. The collection $\{S_1, S_2, \ldots, S_k\}$ of coalitions of N is called a *balanced collection* if there exist positive numbers $\lambda_1, \lambda_2, \ldots, \lambda_k$ such that for every $i \in N$, $\sum_{j: S_j \ni i} \lambda_j = 1$. The numbers $\lambda_1, \lambda_2, \ldots, \lambda_k$ are called *balancing weights* (Kannai, 1992). The core of a game has been characterised by Bondareva (1963) and Shapley (1967) as follows. The core of a game (N, c) is non-empty if and only if for every balanced collection $\{S_1, S_2, \ldots, S_k\}$ with balancing weights $\lambda_1, \lambda_2, \ldots, \lambda_k$, $\sum_{j=1}^{k} \lambda_k c(S_j) \geq c(N)$ holds. Since the non-emptiness of the core and balancedness are shown to be equivalent, the games with non-empty cores are also referred to as *balanced games* . For this example, the allocation (5, 7, 8) that can be read from the MCST, called the Bird rule (Bird, 1976), is always a core element for the MCST games. Therefore, the core of the MCST games is not empty. Thus, MCST games are said to be balanced. Now, consider a *subgame* of (N, c), which is a game that is restricted to a coalition S. Formally, a subgame of (N, c) is denoted by (S, c^S) where $S \subseteq N$, $S \neq \emptyset$ and $c^S(T) = c(T)$ for all $T \subseteq S$. If we take $S = \{1, 3\}$, we have $c^S(\{1\}) = 5$, $c^S(\{3\}) = 13$ and $c^S(\{1, 3\}) = 13$. Then the allocation of £5m and £8m to villages 1 and 3, respectively, by the Bird rule gives a core allocation for the subgame (S, c^S) induced by $S = \{1, 3\}$. If any of the subgames (S, c^S) are balanced, then (N, c) is called *totally balanced*. Since every subgame of an MCST game is also an MCST game and thus is balanced, the MCST games are totally balanced.

Total balancedness provides coalitional stability as we have discussed above since it guarantees that no subset of players pay more than what they would have paid if they acted on their own. Nonetheless, this may not be adequate if the situation is dynamic, that is, new players might join over time. Assume that initially the water distribution system only covered villages 2 and 3. In this case, if the cost was allocated according to the Bird rule, village 2 would pay £10m and village 3 would pay £7m. When village 1 offers to join the water distribution system and the cost allocation is $(5, 7, 8)$ according to the Bird rule, player 3 would veto the inclusion of village 1 since if village 1 is a part of the water distribution system village 3 has to pay £8m instead of £7m. *Population monotonic allocation schemes* (PMASs), defined by Sprumont (1990), require that the cost allocated to a player does not increase if new players join the coalition to which it belongs. Therefore, this notion introduces an incentive for cooperation to the existing players since they would not pay more in the case of new players joining in. Formally, the table $x = (x_{S,i})_{S \in 2^N \setminus \{\emptyset\}, i \in S}$ is a PMAS if the following two conditions hold:

i. For all $S \in 2^N \setminus \{\emptyset\}$, $\sum_{i \in S} x_{S,i} = c(S)$.

ii. For all $S, T \in 2^N \setminus \{\emptyset\}$ such that $S \subseteq T$ and for all $i \in S$, $x_{S,i} \geq x_{T,i}$.

The Bird rule, therefore, would generate a table that is not a PMAS since for $i = 3$, $S = \{2, 3\}$ and $T = \{1, 2, 3\}$ we get $x_{S,i} = 7 \leq 8 = x_{T,i}$ as we have discussed above. It has been shown by Norde et al. (2004) that MCST games always allow a PMAS. For the MCST game considered in this example, a PMAS is presented in Table 1.2. Note that cooperative games that allow PMASs are totally balanced.

S	1	2	3
$\{1, 2, 3\}$	5	8	7
$\{1, 2\}$	5	10	$*$
$\{1, 3\}$	5	$*$	8
$\{2, 3\}$	$*$	10	7
$\{1\}$	5	$*$	$*$
$\{2\}$	$*$	10	$*$
$\{3\}$	$*$	$*$	13

Table 1.2: A PMAS for the MCST game (N, c) (costs in £millions).

$\diamond$

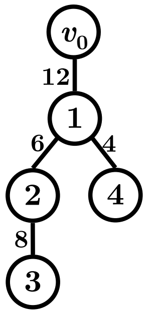


Figure 1.2: A maintenance problem.

Example 1.2. ***Maintenance Games***. We adapt the scenario of this example from Borm et al. (2001). Consider the following situation arising from the maintenance problem displayed in Figure 1.2. Let v_0 represent the town library and let the vertices 1, 2, 3 and 4 represent four houses that are connected to the town library via the fixed road network represented by the edges of the tree. Assume that the stripes on the entire road network need to be repainted and the cost of the repainting is to be paid by the four houses that use the road network. The maintenance costs are represented by the edge costs (in £100s). Therefore, the total cost of maintenance is £3000.

If house 3 wanted to pay for the repainting of the stripes on the roads that connect it to the town library without cooperating with the rest of the players, it would have to pay the cost of the entire path from v_0 to itself, which is £2600. However, if houses 2 and 3 wanted to pay for the repainting of the stripes on the roads that connect them to the town library without cooperating with the rest of the players, they would still pay £2600 since the path from v_0 to house 3 contains the connection of house 2 to v_0. The maintenance game arising from the maintenance problem in Figure 1.2 consists of the player set $N = \{1, 2, 3, 4\}$ and characteristic function c displayed in Table 1.3 where the cost of a coalition is the cost of the fixed tree network's subgraph that is also a tree preserving the connections of all the members of this coalition to the town library at minimum cost.

S	$\{1\}$	$\{2\}$	$\{3\}$	$\{4\}$	$\{1,2\}$	$\{1,3\}$	$\{1,4\}$	$\{2,3\}$	$\{2,4\}$
$c(S)$	12	18	26	16	18	26	16	26	22

S	$\{3,4\}$	$\{1,2,3\}$	$\{1,2,4\}$	$\{1,3,4\}$	$\{2,3,4\}$	$\{1,2,3,4\}$
$c(S)$	30	26	22	30	30	30

Table 1.3: Coalitional costs of the maintenance game (N, c) in £100s.

The maintenance games were discussed in Koster et al. (2001) and Bjøndal et al. (1999). Observe that the underlying combinatorial structure is an MCST for both MCST games and the maintenance games. Nonetheless, the characteristic function of these games, and thus the games are different since the connection network is fixed in maintenance games and is not fixed in MCST games.

For this maintenance game, we will present three well-known one point allocation concepts from the cooperative game theory literature, namely the Shapley value (Shapley, 1953), the nucleolus (Schmeidler, 1969), and the τ value. (Tijs, 1981).

Firstly, let us discuss the Shapley value. Assume that the houses enter this maintenance problem one by one in a given order, say house 1, house 3, house 2 and house 4. House 1 then has to pay the cost of the repainting of the road that connects it to the town library, which is £1200. When house 3 enters, the total cost of houses $\{1, 3\}$ is £2600, but since £1200 has already been paid by house 1, house 3 only pays £1400. Similarly, when house 2 enters it pays the difference between the cost of houses $\{1, 2, 3\}$, and $\{1, 3\}$, which is £0. Finally, when house 4 enters it pays the difference between the cost of houses $\{1, 2, 3, 4\}$, and $\{1, 2, 3\}$, which is £400. Thus, this particular order of entering the maintenance problem gives rise to the allocation $(1200, 0, 1400, 400)$.

Let $\Pi(N)$ denote the set of all permutations of N. Let π denote a permutation of N, and let $\pi(i)$ denote the order of player i in permutation π. Let $\pi^i = \{j : \pi(i) > \pi(j)\}$ denote the set of players preceding i in permutation π. Then the marginal vector $m^\pi(c)$ is defined as

$$m_i^\pi(c) = c(\pi^i \cup \{i\}) - c(\pi^i) \text{ for all } i \in N.$$

Therefore, $(1200, 0, 1400, 400)$ is the marginal vector corresponding to permutation $\pi = [1, 3, 2, 4]$. If we let $\pi = [2, 4, 3, 1]$, we get $m^\pi(c) = (c(\{1, 2, 3, 4\}) - c(\{2, 3, 4\}), c(\{2\}), c(\{2, 3, 4\}) - c(\{2, 4\}), c(\{2, 4\}) - c(\{2\})) = (0, 1800, 800, 400)$.

The Shapley value of a cost game (N, c), denoted by $\phi(c)$, is equal to the average of the marginal vectors over all permutations of N. For the maintenance game (N, c), the Shapley value $\phi(c) = (300, 600, 1400, 700)$. Formally,

$$\phi_i(c) = \sum_{\pi \in \Pi(N)} \frac{m_i^\pi(c)}{n!} \text{ for all } i \in N$$

where $n = |N|$.

Secondly, we discuss the nucleolus. Assume that the total cost of maintenance was allocated using the Shapley value as $(300, 600, 1400, 700)$ and consider the coalition $\{2, 3\}$. If houses 2 and 3 only cooperated with each other, they would have paid £2600, whereas according to the Shapley value they pay £2000. The difference between these two values, which equals £600, can be seen as a measure of satisfaction of this coalition from the Shapley value. In general, the *excess of a coalition S with respect to an allocation $x \in \mathbb{R}^N$*, denoted by $e(S, x)$, is defined as $e(S, x) = c(S) - \sum_{i \in S} x_i$. Furthermore, let $I(c)$ denote the set of *imputations* of a cost game (N, c) such that $I(c) = \{x \in \mathbb{R}^N : \sum_{i \in N} x_i = c(N)$ and $x_i \leq c(\{i\})$ for all $i \in N\}$. We adapt the following interpretation of the nucleolus by Maschler et al. (1979) due to its intuitive appeal. Assume that there exists an arbitrator who interprets $e(S, x)$ as the satisfaction of coalition S from allocation x. The arbitrator would like to choose an imputation that maximises the satisfaction of the least satisfied coalition from allocation x. There can possibly be more than one imputation that satisfy this criterion. Amongst these imputations, the arbitrator then would like to choose the one that maximises the satisfaction of the second least satisfied coalition from allocation x. If the arbitrator continues in this manner, the unique imputation that he reaches is the nucleolus. Thus, the nucleolus of a cooperative cost game is the unique imputation that lexicographically maximises the excesses of all coalitions. Formally, let $I(c) \neq \emptyset$ and let $x \in I(c)$ be an imputation. Let $\Theta(x) \in \mathbb{R}^{2^N}$ be a vector whose elements are the excesses $e(S, x)$ of all possible coalitions in 2^N in a nondecreasing order. Let $y, z \in \mathbb{R}^k$ be two vectors for which either there exists an index $j \leq k$ such that $y_i = z_i$ for $i < j$ and $y_j > z_j$ or $y = z$. Then y is said to be *lexicographically greater than or equal to* z, denoted by $y \succeq_{lex} z$. The *nucleolus* $\eta(c) \in I(c)$ is the unique imputation such that $\Theta(\eta(c)) \succeq_{lex} \Theta(x)$ for all $x \in I(c)$. For a balanced game, the nucleolus is always a core element. The nucleolus of the maintenance game (N, c) is equal to $\eta(c) = (400, 500, 1300, 800)$.

For a general cooperative cost game, the main drawback of the nucleolus and the Shapley value is the complexity of computing these allocations. We mention the work of Elkind and Pasechnik (2009) on the complexity of computing the nucleolus of weighted voting games, which may be applied to a wider class of games, as well as the work of Bachrach et al. (2010) on the complexity of computing the Shapley value of simple coalitional games.

Due to the special structure of the maintenance games, the nucleolus and the Shapley value can be computed easily with the following two "painting stories". We start with the computation of the nucleolus, which is due to Maschler et al. (1995). Assume that four painters, each hired by one of the four houses, are to carry out the repainting work. Assume furthermore that the edge costs now represent the number of minutes it would take one painter to repaint the stripes on the corresponding road. The repainting work is carried out according to the following rules (Borm et al., 2001):

(a) Every painter paints with speed 1.

(b) Every painter has to keep on working until the stripes on all the roads from the town library to the house that employed him have been repainted.

(c) Every painter only works on a road that is between the town library and the house that employed him.

(d) If the stripes on a road between a house's predecessor on the fixed tree and the town library have not been fully repainted, the painter employed by this house carries out the painting work on this road.

(e) As long as rules (a)-(d) permit, every painter works as close to the house that employed him as possible.

Table 1.4 presents the time (t) period in minutes and the roads (edges) that each painter is working on where e_{ij} denotes the edge between vertices i and j. Note that all the painters start working at $t = 0$ and a ✓ indicates that the painter has completed his work.

	1	2	3	4
$0 \leq t \leq 400$	$e_{v_0 1}$	$e_{v_0 1}$	e_{12}	$e_{v_0 1}$
$400 < t \leq 500$	✓	e_{12}	e_{12}	e_{14}
$500 < t \leq 800$	✓	✓	e_{23}	e_{14}
$800 < t \leq 1300$	✓	✓	e_{23}	✓
$1300 < t$	✓	✓	✓	✓

Table 1.4: The time (t) period in minutes and the roads whose stripes are being repainted by the four painters according to rules (a)-(e).

Therefore, painter 1 works for 400 minutes, painter 2 works for 500 minutes, painter 3 works for 1300 minutes and painter 4 works for 800 minutes to complete the repainting. This gives rise to the allocation $(400, 500, 1300, 800)$, which is the nucleolus of this game.

The Shapley value is computed with a slightly modified painting story (Bjøndal et al., 1999). Assume that rule (d) no longer applies and that rule (e) is replaced by (Borm et al., 2001):

(f) As long as rules (a)-(c) permit, every painter works as close to the town library as possible.

In this case, we get the time periods and the roads that each painter is working on presented in Table 1.5.

	1	2	3	4
$0 \leq t \leq 300$	$e_{v_0 1}$	$e_{v_0 1}$	$e_{v_0 1}$	$e_{v_0 1}$
$300 < t \leq 600$	✓	e_{12}	e_{12}	e_{14}
$600 < t \leq 700$	✓	✓	e_{23}	e_{14}
$700 < t \leq 1400$	✓	✓	e_{23}	✓
$1400 < t$	✓	✓	✓	✓

Table 1.5: The time (t) period in minutes and the roads whose stripes are being repainted by the four painters according to rules (a)-(c) and (f).

Therefore, painter 1 works for 300 minutes, painter 2 works for 600 minutes, painter 3 works for 1400 minutes and painter 4 works for 700 minutes to complete the repainting. This gives rise to the allocation $(300, 600, 1400, 700)$, which is the Shapley value of this game.

Thirdly, we illustrate the τ value. Let $i \in N$. Let us first assume that house i wants to pay the minimum amount it can hope to pay upon cooperation with the rest of the houses. This amount equals $c(N) - c(N \backslash \{i\})$ because this cost is only incurred when house i enters the game and is referred to as the *marginal contribution* of house i. For the maintenance game (N, c) of this example, this principle gives us the allocation $m(c) = (0, 0, 800, 400)$. The allocation of the marginal contribution $m_i(c)$ to house i can be seen as a lower bound on its share of the total cost. Now, let us establish an upper bound as follows. Consider the coalition $\{1, 2, 3\}$ and assume that all the houses except for house 1 pay their marginal contribution. Then house 1 would have to pay the remaining cost of $c(\{1, 2, 3\}) - m_2(c) - m_3(c) =$ 2600 - 0 - 800 $= 1800$. In fact, with respect to this coalition house 1 would never agree to pay more than £1800 since all the other houses in the coalition are already paying the lowest possible amount that they can hope to pay. For every coalition that house 1 is a member of it would never pay more than the remaining amount if the rest of the houses only pay their marginal contribution. Therefore, house 1

would only agree to pay the minimum of such remaining costs amongst all the coalitions that it is a member of. The minimum remaining cost of house 1 is with respect to coalition $\{1,4\}$, and calculated as $c(\{1,4\})-m_4(c) = 1600$ - $400 = 1200$. If every house employs this principle, we get the allocation $M(c) = (1200, 1800, 2600, 1600)$. Since, the total cost of repainting is £3000 and this is exactly the cost the houses would cover, we take the convex combination of the two aforementioned vectors such that the sum of the shares of the four houses in the resulting allocation equals £3000. In order words, we take the convex combination of vectors $m(c)$ and $M(c)$ such that the resulting allocation is efficient. This gives us the τ value of (N, c), $\tau(c) = (360, 540, 1340, 760)$. The τ value is a compromise between an allocation vector that favours each player and an allocation vector that disfavours each player. Formally, the *τ value* is defined as follows. Let $m_i(c) = c(N) - c(N\backslash\{i\})$ and let $M_i(c) = \min_{S\subseteq N:\, i\in S}\left\{c(S) - \sum_{j\in S\backslash\{i\}} m_j(c)\right\}$ for all $i \in N$. Then $\tau(c) = \mu m(c) + (1-\mu)M(c)$ where μ is such that $\sum_{i\in N}\tau_i(c) = c(N)$.

Finally, we consider two properties of the maintenance games. The cooperative game (N, c) is *submodular* if for all $i \in N$ and for all $S \subset T \subseteq N\backslash\{i\}$, its characteristic function satisfies $c(S\cup\{i\}) - c(S) \geq c(T\cup\{i\}) - c(T)$. Therefore, the incentive for joining a coalition increases as the coalition grows for a submodular game. Submodular games allow PMASs and consequently are totally balanced. The core of the submodular games has been characterised by Shapley (1971) and Ichiishi (1981) as the convex hull of the marginal vectors. Therefore, the Shapley value is always stable and the centre of mass of the marginal vectors for submodular games. The cooperative game (N, c) is *monotone* if its characteristic function satisfies $c(S) \leq c(T)$ for any $S \subseteq T \subseteq N$. Thus, this property implies that the cost of a given coalition is greater than or equal to the cost of each of its subsets, that is, larger coalitions pay more. The maintenance games are submodular and monotone.

Various generalisations of maintenance games are studied in Megiddo (1978), Granot and Granot (1992), Maschler et al. (1995), Granot et al. (1996), Granot and Maschler (1998), van Gellekom and Potters (1999), Miquel et al. (2006). $\diamond$

Example 1.3. ***Chinese Postman Games***. Consider the following situation arising from the Chinese postman problem displayed in Figure 1.3.

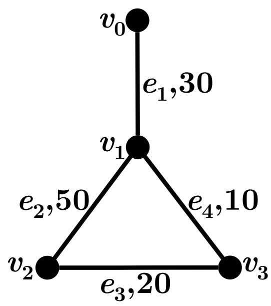


Figure 1.3: A Chinese postman problem.

Let v_0 represent the post office. A postman, who has to start and finish his journey at the post office, has to deliver mail to four streets of a neighbourhood represented by the edges e_1 to e_4. In order to deliver the mail, the postman has to visit each street at least once. The cost of the postman to travel through a street is represented by the weight of the corresponding edge. The combinatorial optimisation problem is to determine the tour that starts and ends at v_0, and visits each edge at least once at minimum cost. This problem is the Chinese postman problem (Kwan, 1963). An optimal tour for the problem in Figure 1.3 is $v_0, e_1, v_1, e_2, v_2, e_3, v_3, e_4, v_1, e_1, v_0$.

A cooperative cost game arising from this combinatorial optimisation problem is introduced by Hamers et al. (1999). Firstly, let $N = \{1, 2, 3, 4\}$ be the set of players such that player $i \in N$ is associated with edge (street) e_i. The computation of the cost of a coalition in the Chinese postman game of Hamers et al. (1999) is as follows. There are two types of costs involved, namely the travel and delivery costs. The cost of travelling through a street is the weight of the associated edge and is incurred every time the street is visited. On the other hand, the cost of delivery is only incurred when the postman delivers mail to a street. The delivery cost is the same for all streets and equals £10. Since the postman needs to start and finish his journey at the post office, he would have to travel through streets even when he is not delivering mail and in fact the cost of such trips constitute the characteristic function of the Chinese postman games. Let us explain this by calculating $c(\{1, 2\})$. The associated minimum cost tour is $v_0, e_1, v_1, e_2, v_2, e_3, v_3, e_4, v_1, e_1, v_0$. The postman starts by travelling through and delivering to streets e_1 and e_2, thus incurring the corresponding travel and delivery costs of £30+£10=£40 and £50+£10=£60, respectively. The costs of £40 and £60 are referred to as fixed costs and are to be paid by the associated players 1 and 2, respectively. Therefore, these costs can be attributed to the members of the coalition $\{1, 2\}$ individually and are not significant to the calculation of the cost of this coalition within the context of the Chinese postman game. Now, the postman needs to travel from v_2 to v_0 through streets e_3, e_4 and e_1 incurring £20+£10+£30 $=$ £60 of travelling cost when he is not delivering mail. This amount is to be covered by the members of the coalition $\{1, 2\}$ collectively. Thus, $c(\{1, 2\}) = 60$. Let us also calculate $c(\{3\})$. In this case, the minimum cost tour of the postman is $v_0, e_1, v_1, e_4, v_3, e_3, v_2, e_3, v_3, e_4, v_1, e_1, v_0$. Since the postman delivers mail during his first visit to e_3, we ignore the travelling cost of his first visit to e_3 upon calculating $c(\{3\})$. The cost of this coalition therefore covers travelling through streets e_1, e_4, e_3 (on the return trip the postman is not delivering mail to this street), e_4 and e_1 and adds up to £30+£10+£20+£10+£30=£100 of travelling cost when he is not delivering mail. Thus, $c(\{3\}) = 100$. The characteristic function c is displayed in Table 1.6.

S	$\{1\}$	$\{2\}$	$\{3\}$	$\{4\}$	$\{1,2\}$	$\{1,3\}$	$\{1,4\}$	$\{2,3\}$	$\{2,4\}$
$c(S)$	30	90	100	70	60	70	40	70	80

S	$\{3,4\}$	$\{1,2,3\}$	$\{1,2,4\}$	$\{1,3,4\}$	$\{2,3,4\}$	$\{1,2,3,4\}$
$c(S)$	90	40	50	60	60	30

Table 1.6: Coalitional costs of the Chinese postman game (N, c).

We conclude this example by summarising the relationships that have been established between the properties of the underlying undirected graph and the properties of the associated Chinese postman game by Hamers et al. (1999) and Granot et al. (1999). Note that Granot et al. (1999) also study the properties of the directed and mixed graphs. A graph is said to be *Chinese Postman (CP) balanced*, *CP totally balanced* and *CP submodular* if the corresponding Chinese postman game is balanced, totally balanced and submodular, respectively, for all edge costs and all locations of the post office. A connected undirected graph G is called *Eulerian* if G can be traversed such that each edge is visited only once. A connected undirected graph G is called *weakly Eulerian* if it consists of a number of Eulerian graphs such that shrinking each of these Eulerian graphs into a vertex gives a tree. A connected undirected graph G is called *weakly cyclic* if every edge of G is contained in at most one cycle. The first result is that a connected undirected graph G is weakly Eulerian if and only if it is CP balanced. The 'only-if'-part and the 'if'-part of this statement are due to Hamers et al. (1999) and Granot et al. (1999), respectively. For a connected undirected graph G, being weakly cyclic, CP totally balanced and CP submodular are shown to be equivalent by Granot et al. (1999). We furthermore mention the work of Granot and Hamers (2004) for a similar approach to studying game theoretical properties in relation to graph theoretical properties for CP and travelling salesman games. ◇

Example 1.4. ***Highway Games***. Consider the following situation arising from a highway problem. Let $N = \{1, 2, 3\}$ denote the set of agents. Each agent $i \in N$ is responsible for constructing a highway between two locations s_i and t_i. Locations of s_i and t_i, and the costs of the edges, which represent feasible connections, are displayed in Figure 1.4.

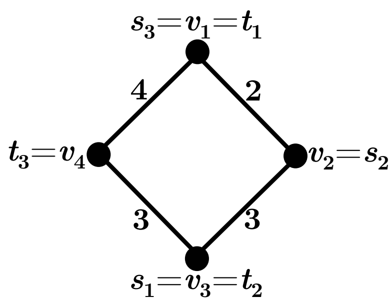


Figure 1.4: A highway problem.

The optimal solution of this problem is to construct a highway between v_1 and v_2, between v_2 and v_3, and between v_3 and v_4 at a total cost of £8bn.

The highway games were introduced by Mosquera (2007) for the case where the underlying graph is a tree. Nonetheless, here we present the cooperative cost game associated with a highway problem on a weakly cyclic graph, which is studied by Çiftçi et al. (2010). Next, we illustrate the computation of the cost of a coalition in a highway game. Let us start by calculating $c(\{1\})$. If agent 1 were to pay for the construction of a highway between s_1 and t_1, that is, between v_1 and v_3, the cheapest connection would be established through v_2 at a cost of $c(\{1\}) = 5$. Let us now calculate $c(\{1, 3\})$ where agents 1 and 3 cooperate to build a highway that connects s_1 to t_1, and s_3 to t_3 at minimum cost. Therefore, agents 1 and 3 construct a highway between v_1 and v_3 through v_4 at a cost of $c(\{1, 3\}) = 7$. The characteristic function c is displayed in Table 1.7.

S	$\{1\}$	$\{2\}$	$\{3\}$	$\{1,2\}$	$\{1,3\}$	$\{2,3\}$	$\{1,2,3\}$
$c(S)$	5	3	4	5	7	7	8

Table 1.7: Coalitional costs of the highway game (N, c) in £billions.

Finally, we summarise the characterisation of balancedness and submodularity of highway games on weakly cyclic graphs by Çiftçi et al. (2010). Similar to the case of the Chinese postman games, game theoretical properties are studied in relation to the properties of the underlying graph. A graph is said to be *highway game (HG) balanced* and *HG submodular* if all highway games induced by G are balanced and submodular, respectively. A weakly cyclic graph G is called *weakly triangular* if every cycle in G has 3 edges. Firstly, a graph G is weakly cyclic if and only if G is HG balanced. Secondly, a graph G is weakly triangular if and only if G is HG submodular. $\diamond$

1.2 Overview

We present a brief summary of the thesis in this section.

The first part of the thesis consisting of Chapter 2, Chapter 3 and Chapter 4 is concerned with situations arising from a shortest path tree problem and the associated games. Each agent in a shortest path tree problem corresponds to a vertex of an undirected complete graph, in which a special vertex represents the common supplier. The joint combinatorial optimisation problem of the agents involved consists of determining a tree that spans all the agents such that the path between an agent and the supplier is a shortest path. The cost of this shortest path tree is the sum of the cost of the shortest paths between the supplier and all the agents. In a shortest path tree game, the set of players is equal to the set of agents of the shortest path tree problem and the cost of a coalition is the cost of a shortest path tree formed by its members.

We illustrate a shortest path tree problem and the associated cooperative game in the following example.

Example 1.5. ***Shortest Path Tree Games***. Consider the following situation arising from the shortest path tree problem displayed in Figure 1.5.

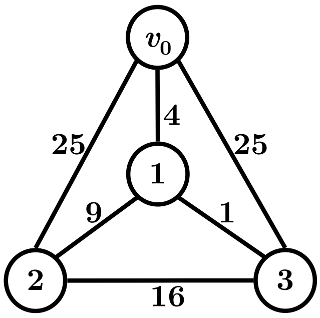


Figure 1.5: A shortest path tree problem.

Let v_0 represent a base station that provides internet to the users 1, 2 and 3 represented by the vertices. The edge costs represent (in £) the cost of establishing a wireless connection between two users or between a user and the base station. The users can connect to the base station directly or via other users who are willing to cooperate. If the three users decide to act on their own, they need to pay £4, £25 and £25, respectively, to connect to the base station. However, if they all decide to cooperate the minimum cost connection of the three users to the base station is a shortest path tree rooted at v_0 where user 1 connects directly to the base station at a cost of £4 and users 2 and 3 connect to the base station via user 1 at a cost of £13 and £5, respectively. Therefore, the total cost of the shortest path tree equals £4+£13+£5=£22 and this is the cost to be shared amongst the three users. The shortest path tree game (N, c) arising from the shortest path tree problem in Figure 1.5 consists of the player set $N = \{1, 2, 3\}$ and the characteristic function c displayed in Table 1.8 where the cost of a coalition $S \subseteq N$ is equal to the cost of the shortest path tree formed by its members to connect to v_0.

S	$\{1\}$	$\{2\}$	$\{3\}$	$\{1,2\}$	$\{1,3\}$	$\{2,3\}$	$\{1,2,3\}$
$c(S)$	4	25	25	17	9	50	22

Table 1.8: Coalitional costs of the shortest path tree game (N, c) in £.

◇

Chapter 2 defines and studies the shortest path tree problems. We address the allocation of the cost of a shortest path tree amongst the agents by proposing three different allocation rules specific to our context of shortest path tree problems. Firstly, the *tree solution* θ assigns to each agent the cost of its own shortest path on a shortest path tree formed by all the agents. Secondly, the δ^i rule is defined with respect to an agent $i \in N$ by considering a shortest path tree formed by all the agents excluding i. According to this rule all the agents excluding i pay the cost of their own shortest path on a shortest path tree formed by all the agents excluding i, and agent i pays the difference between the cost of a shortest path tree formed by all the agents and the cost of a shortest path tree formed by all the agents excluding i, that is, agent i pays its marginal contribution. Thirdly, the γ rule is defined as the average of δ^i for all agents i. Note that the δ^i and the γ rules are novel allocation rules proposed by us in this thesis. We furthermore analyse a number of properties of these three allocation rules in order to compare and contrast them. Finally, we axiomatically characterise the tree solution θ.

In Chapter 3, we introduce the shortest path tree games arising from the shortest path tree problems. We start by analysing the properties of the shortest path tree games to help us position these games amongst other classes of cooperative games. We show that the shortest path tree games are totally balanced, allow a PMAS but are not submodular and not monotone. Next, we apply a number of existing solution concepts from cooperative game theory literature to shortest path tree games. In particular, we consider the core (Gillies, 1953), the Shapley value (Shapley, 1953), the τ value (Tijs, 1981) and the nucleolus (Schmeidler, 1969). Initially, we concentrate on core allocations since the core of the shortest path tree games is not empty. Firstly, we demonstrate that the rules θ, δ^i and γ all generate core allocations. Secondly, we show that the Shapley and the τ values are not necessarily core elements for the shortest path tree games. Our work on core allocations is concluded by presenting two methods of generating core allocations for the shortest path tree games. These methods are based on determining a shortest path tree in the absence of a special player and thus are efficient since the shortest path tree problem can be solved efficiently using Dijkstra's algorithm (Dijkstra, 1959). Next, a polyhedral analysis of the core of the shortest path tree games is presented by identifying a class of extreme points and determining the dimension of the core of shortest path tree games, and by finding a class of facets of the core of the shortest path tree games that correspond to shortest path tree problems with a unique optimal tree. Furthermore, a collection of coalitions are identified for which the corresponding inequalities are redundant in the description of the core of the shortest path tree games, which leads to a result on the reduced description of the core of the shortest path tree games. In this chapter, we also discuss some aspects of the computation of the nucleolus of shortest path tree games from a theoretical point of view. Finally, we discuss the implications of the special case where the triangle inequality holds for the shortest path tree problem.

Chapter 4 presents our computational results on the core and the nucleolus of the SPT games. We consider a practical application of the SPT games, namely the cost allocation problem in Wireless Multihop Networks (WMNs). We start by defining WMNs and the associated cost allocation problem, and discussing their properties. Next, we present the results of our simulations on the reduction in the definition of the core of the SPT games. We then present three approaches to computing the nucleolus of the SPT games for this application. Firstly, we propose to compute the nucleolus of the SPT games for the WMN application using the linear programming based algorithm of Kopelowitz. Secondly, our reduction in the description of the core result is incorporated into this algorithm. Finally, a constraint generation approach for the computation of the nucleolus is employed in order to generate the cost of the non-redundant coalitions on the fly. This section concludes by a comparison of the performance of these three approaches.

The second part of the thesis consisting of Chapter 5 focuses on conflict situations arising from a weighted minimum colouring problem. Each agent represented by a vertex of an undirected graph is interested in having access to a number of facilities but can be in conflict with other agents. The number of facilities that an agent would like to have access to is represented by the weight of the corresponding vertex and the conflict relations between the agents are represented by edges between the corresponding vertices. The joint combinatorial optimisation problem of the agents is to determine the minimum number of facilities subject to the conflict relations, that is, to determine the weighted chromatic number of the graph representing the conflict relations. In order to address the allocation of the cost of the minimum number of facilities amongst the agents, we introduce a class of cooperative cost games, namely the weighted minimum colouring games. The following example demonstrates a weighted minimum colouring game.

Example 1.6. ***Weighted Minimum Colouring Games***. Consider the following situation corresponding to the weighted minimum colouring problem displayed in Figure 1.6.

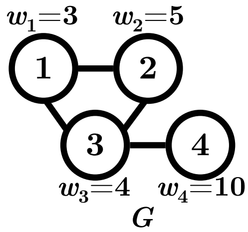


Figure 1.6: A weighted minimum colouring problem induced by graph G and weight vector w.

Let the vertices 1, 2, 3 and 4 represent four transmitters each of which needs to be assigned a certain number of frequency bands. The number of frequency bands required by each transmitter i is represented by the weight w_i of the corresponding vector. If two transmitters are connected by an edge, then unacceptable interference might occur between them and hence they must be assigned disjoint sets of frequency bands. Assume that each transmitter is owned by a cellular telephone network company. If companies 1, 2 and 4 all decide to act on their own then they require $3 + 5 + 10 = 18$ frequency bands. Transmitters 1 and 2 are connected and thus together they need 8 frequency bands. However, if companies 1 and 2 cooperated with company 4, the three companies would only require 10 frequency bands in total since transmitters 1 and 2 can use 8 of the 10 frequency bands of transmitter 4. Assume that the cost of a frequency band is fixed and equals £1. The weighted minimum colouring game arising from this problem consists of the player set $N = \{1, 2, 3, 4\}$ and the characteristic function $c^{G,w}$ displayed in Table 1.9 where the cost of a coalition $S \subseteq N$ is computed as the cost of the minimum number of frequency bands required by the players in S.

S	$\{1\}$	$\{2\}$	$\{3\}$	$\{4\}$	$\{1,2\}$	$\{1,3\}$	$\{1,4\}$	$\{2,3\}$	$\{2,4\}$
$c^{G,w}(S)$	3	5	4	10	8	7	10	9	10

S	$\{3,4\}$	$\{1,2,3\}$	$\{1,2,4\}$	$\{1,3,4\}$	$\{2,3,4\}$	$\{1,2,3,4\}$
$c^{G,w}(S)$	14	12	10	14	14	14

Table 1.9: Coalitional costs of the weighted minimum colouring game $(N, c^{G,w})$.

◇

Observe that the minimum number of frequency bands required by the players in S is equal to the weighted chromatic number of the subgraph of G induced by S with respect to w.

Note that the minimum colouring games arising from the minimum colouring problem, which is a special case of the weighted colouring problem when all the vertex weights are equal to 1, are introduced by Deng et al. (1999). The characterisation of total balancedness, submodularity and existence of a PMAS of minimum colouring games are due to Deng et al. (2000), Okamoto (2003) and Hamers et al. (2011), respectively. Chapter 5 characterises total balancedness and submodularity of the weighted minimum colouring games as follows. We show that a graph G induces a totally balanced WMC game for all positive integer weight vectors if and only if it is perfect and that any graph G induces a totally balanced WMC game for at least one positive integer weight vector. Furthermore, we show that a graph G induces a submodular WMC game for all positive integer weight vectors if and only if it is complete r-partite and that a graph G induces a submodular WMC game for at least one positive integer weight vector if and only if it is $(2K_2, P_4)$-free.

We conclude the thesis with some remarks on further research we have carried out on SPT problems and games as well as future research directions for both SPT and WMC games.

Chapter 2

Shortest Path Tree Problems

The single-source shortest path problem consists of finding a tree that spans all the vertices in a given graph with the property that the path between a preidentified source vertex to every other vertex in the graph is a shortest path. Thus, alternatively, one can refer to this problem as the shortest path tree problem. This problem often arises in designing wireless communication networks. Depending on the particular application, the shortest path tree problem may be dealing with time, cost, path loss or other aspects of these networks that accumulate additively along a path and is to be minimised. The application that initiated the study in this chapter is an example to such a wireless communication network problem (Makki et al., 2008). In a Wireless Multihop Network (WMN), there are geographically spread users who need to connect to the internet that is provided by the base station. In these networks, the connection cost of a user depends on the entire path between itself and the base station. Each wireless link in the network has a cost associated with the power needed to transmit, which is proportional to the distance between the users to the power of a small integer in the range $2-4$. Users can relay other users' signals in order to reduce the total power used in the network. The optimal (minimum cost) solution, which is calculated by the base station, is a shortest path tree formed by all the users. We provide further details, examples and analysis of WMNs in Section 4.1.

Graphs consist of discrete elements such as vertices and edges, therefore they provide a natural setting for situations where decision makers or agents are either located at or can control different parts of the network underlying their shared problem. This chapter is concerned with the allocation of joint costs in the context of a single-source shortest path problem on a graph and introduces the shortest path tree problems. We would like to emphasise that this problem, as well as the corresponding games studied in the next chapter, have been defined and analysed for the first time in this thesis. First, we associate an agent with each vertex, excluding the single special vertex that corresponds to the source, on an undirected complete graph with nonnegative edge costs. The optimal solution to the agents' joint single-source shortest path problem is a shortest path tree that connects them to the source. This shortest path tree has a cost that is the sum of the cost of the shortest paths between the source and all the agents. Having introduced this cooperative situation, we turn our attention to introducing allocation rules for sharing the cost of a shortest path tree amongst the agents. Firstly, we define the *tree solution* θ where each agent is assigned the cost of its shortest path on the shortest path tree formed by all the agents. Secondly, we define an allocation rule, which we call the δ^i rule with respect to an agent i. For this allocation, we consider the shortest path tree formed by the agents excluding i, each of which pays the cost of its shortest path on this tree, and agent i pays its marginal contribution. Thirdly, we consider an allocation rule γ that is the average of δ^i for all agents i. Note that δ^i and γ are novel allocation rules suggested by us in this chapter. Next, we consider a number of properties of cost allocation rules, and identify the properties satisfied by the aforementioned allocation rules. Finally, we present a characterisation of the tree solution θ.

This chapter is organised as follows. In Section 2.1, we define the shortest path tree problems. In Section 2.2, we present a number of properties of shortest path trees that will be used throughout this thesis. We then propose three different allocation rules for sharing cost amongst the agents in a shortest path tree problem in Section 2.3.1. Finally, in Section 2.3.2, we define properties of cost allocation rules in shortest path tree problems, analyse the allocation rules using these properties and we provide a characterisation of one of the allocation rules that we propose.

Related Work

One of the well-known problems involving a network of decision makers and an underlying optimisation problem is the minimum cost spanning tree problems. In these problems, each agent is assumed to be located at a vertex of a graph, excluding a given source vertex, and there are costs associated with the edges of this graph. The graph theoretical question is to connect all the agents to the source vertex at minimum cost. Similar to a shortest path tree, a minimum cost spanning tree is a tree that spans all the agents. However, unlike a shortest path tree, the cost of a minimum cost spanning tree is equal to the sum of the costs of the edges used for connection. Such problems arise in applications such as allocating the cost of a water pipeline system amongst villages that connect to a water supplier or the cost of a cable television network amongst its users. The common aspect in these examples is that the cost of every link, a pipe or a cable connection, between two agents involved is incurred only once upon construction of the network. The cost allocation problem in minimum cost spanning trees was introduced by Claus and Kleitman (1973) and starting with the work of Bird (1976), has been extensively studied. There are numerous cost allocation rules that have been proposed for the minimum cost spanning tree problems. Some of these allocation rules, as well as the analysis of their properties, can be found in Branzei et al. (2004), Moretti et al. (2005), Tijs et al. (2006), Bergantiños and Vidal-Puga (2007) and Bergantiños and Kar (2010).

2.1 Definition of the Shortest Path Tree Problems

In this section, we define a shortest path tree (SPT) problem and introduce some notation related to the SPT problems that will be used throughout this chapter.

Let $N = \{1, 2, \ldots, n\}$ denote the set of agents. Let $G = (N \cup \{v_0\}, E)$ be an undirected complete graph with finite vertex set $N \cup \{v_0\}$ and edge set $E = \{\{i, j\} : i, j \in N \cup \{v_0\} \text{ and } i \neq j\}$ where v_0 is a special vertex that represents the source vertex. Let $t : E \to \mathbb{R}_0^+$ be a cost function that assigns a nonnegative cost t_{ij} to every edge $\{i, j\} \in E$ where $t_{ij} = t_{ji}$ since the graph is undirected. We denote an SPT problem by $((N \cup \{v_0\}, E), t)$. Observe that the source vertex v_0 is not associated with an agent, and every vertex in G, excluding the source vertex, is associated with exactly one agent.

The SPT problem $((N \cup \{v_0\}, E), t)$ consists of finding a tree $\mathcal{T}_{v_0}^N$ rooted at v_0 such that the unique path from v_0 to each $i \in N$ is a minimum cost path from v_0 to i in G. We refer to $\mathcal{T}_{v_0}^N$ as a shortest path tree. For simplicity of notation, we omit the subscript and the superscript and denote a shortest path tree of the SPT problem $((N \cup \{v_0\}, E), t)$ by $\mathcal{T}$. Henceforth, the shortest path of i on $\mathcal{T}$ will refer to the shortest path from v_0 to i on $\mathcal{T}$. Let $E(i, \mathcal{T})$ denote the set of edges on the shortest path of i on $\mathcal{T}$ and let $V(i, \mathcal{T})$ denote the set of vertices on the shortest path of i on $\mathcal{T}$, excluding v_0 but including i. The cost of $E(i, \mathcal{T})$ is denoted by $t^{E(i,\mathcal{T})}$ and is equal to the sum of the costs of the edges in $E(i,\mathcal{T})$. Formally,

$$t^{E(i,\mathcal{T})} = \sum_{\{k,l\}\in E(i,\mathcal{T})} t_{kl}.$$

The cost of a shortest path tree $\mathcal{T}$ is denoted by $t(\mathcal{T})$ and is equal to the sum of the costs of the shortest paths of all $i \in N$. We have

$$t(\mathcal{T}) = \sum_{i\in N} t^{E(i,\mathcal{T})}.$$

Note that there can be multiple shortest path trees, however the cost of the shortest path of an agent is the same on all shortest path trees of a given SPT problem. Consecutively, all the shortest path trees of a given SPT problem have the same cost. In other words, given an SPT problem, the value of $t^{E(i,\mathcal{T})}$ for all $i \in N$ and $t(\mathcal{T})$ do not depend on the choice of a specific shortest path tree.

Next, we give an example to the SPT problem and illustrate the concepts defined above.

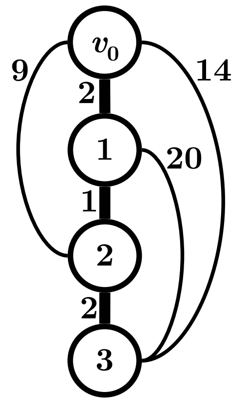


Figure 2.1: An SPT problem.

Example 2.1. Consider the SPT problem displayed in Figure 2.1. $\mathcal{T}$ is indicated by the bold lines. For agent 3, $E(3,\mathcal{T}) = \{\{v_0,1\},\{1,2\},\{2,3\}\}$ and $V(3,\mathcal{T}) = \{1,2,3\}$. We have $t(\mathcal{T}) = t^{E(1,\mathcal{T})} + t^{E(2,\mathcal{T})} + t^{E(3,\mathcal{T})} = 2+3+5 = 10$. $\diamond$

We would like to highlight that in the case of the triangle inequality holding, the shortest path tree will always be a star graph. Further discussions on this special case along with its implications for our results on SPT games are presented in Section 3.8.

Now, let us introduce notation relating to the SPT problem defined on a subgraph of G. Let $S \subseteq N$. Then the SPT problem $((S \cup \{v_0\}, E^S), t^S)$ is defined on $G^S = (S \cup \{v_0\}, E^S)$ such that $E^S = \{\{i, j\} : i, j \in S \cup \{v_0\}$ and $i \neq j\}$ where v_0 represents, also in the subgraph, the source vertex and t^S the restriction of the cost function t to S such that $t^S_{ij} = t_{ij}$ for all $i, j \in S \cup \{v_0\}$. We denote an optimal solution to $((S \cup \{v_0\}, E^S), t^S)$ by $\mathcal{T}^S$. For an agent $i \in S$, $E(i, \mathcal{T}^S)$ denotes the set of edges on the shortest path of i on $\mathcal{T}^S$ and $V(i, \mathcal{T}^S)$ denotes the set of vertices on the shortest path of i on $\mathcal{T}^S$, excluding v_0 but including i. The cost of $E(i, \mathcal{T}^S)$ is denoted by $t^{E(i,\mathcal{T}^S)}$ such that

$$t^{E(i,\mathcal{T}^S)} = \sum_{\{k,l\} \in E(i,\mathcal{T}^S)} t^S_{kl}.$$

The cost of a shortest path tree $\mathcal{T}^S$ is denoted by $t(\mathcal{T}^S)$ such that

$$t(\mathcal{T}^S) = \sum_{i \in S} t^{E(i,\mathcal{T}^S)}.$$

Example 2.2. Consider the SPT problem displayed in Figure 2.1. Let $S = \{2, 3\}$. The shortest path tree $\mathcal{T}^S$ formed by agents 2 and 3 to connect to v_0 is unique and its edges are $\{v_0, 2\}$ and $\{2, 3\}$. For agent 2, $E(2, \mathcal{T}^S) = \{\{v_0, 2\}\}$ and $V(2, \mathcal{T}^S) = \{2\}$. For agent 3, $E(3, \mathcal{T}^S) = \{\{v_0, 2\}, \{2, 3\}\}$ and $V(3, \mathcal{T}^S) = \{2, 3\}$. We have $t(\mathcal{T}^S) = t^{E(2,\mathcal{T}^S)} + t^{E(3,\mathcal{T}^S)} = 9 + 11 = 20$. ◇

Differences between the Shortest Path Tree Problems and Games and the Minimum Cost Spanning Tree Problems and Games

We start by discussing the differences between the SPT problem the minimum cost spanning tree (MCST) problem. Firstly, an SPT of a graph is not necessarily an MCST of the graph. The MCST is a spanning tree that connects the vertices in N to v_0 such that the sum of the costs of the edges in the tree is minimum. Therefore, given a graph G, an SPT and an MCST are not necessarily the same spanning tree, as we illustrate in Example 2.3.

Example 2.3. Let us consider the graph G, and the two subgraphs $\mathcal{T}$ and $\hat{\mathcal{T}}$ of G in Figure 2.2. $\mathcal{T}$ is the SPT. We have $t(\mathcal{T}) = t^{E(1,\mathcal{T})} + t^{E(2,\mathcal{T})} + t^{E(3,\mathcal{T})} = 2 + 2 + 3 = 7$. $\hat{\mathcal{T}}$ is an MCST with a total cost of $t_{v_0 1} + t_{12} + t_{13} = 2 + 1 + 1 = 4$. ◇

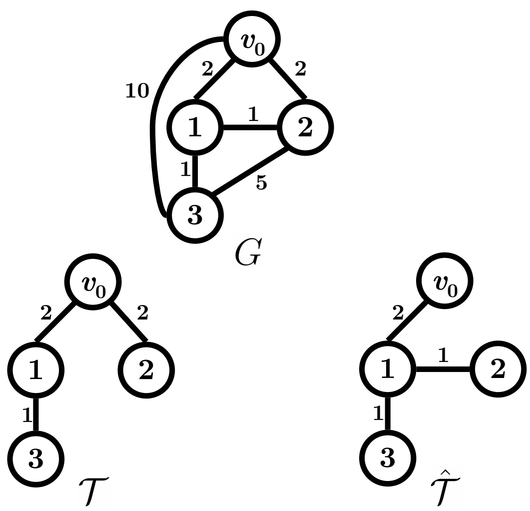


Figure 2.2: $\mathcal{T}$ and $\hat{\mathcal{T}}$ of G form a shortest path tree and a minimum cost spanning tree respectively.

This implies that, as we have discussed in Example 1.1, the characteristic function of an MCST game is different than that of an SPT game, which we define and study in the next chapter.

Secondly, we would like to mention the updating of an SPT and an MCST. Updating refers to adding a vertex, increasing the cost of a tree edge or decreasing the cost of a non-tree edge. Spira and Pan (1973) have shown that solving an updated SPT problem is as hard as solving the original problem. This is shown not to be the case for the updating of the MCSTs. Consider the following example of increasing the cost of a tree edge.

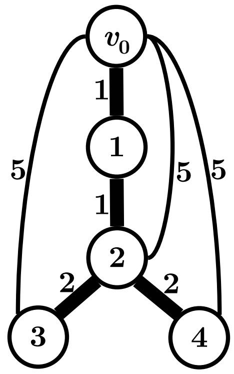
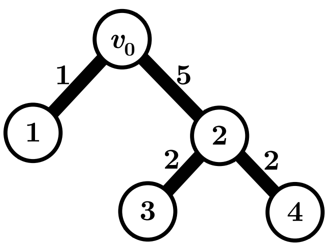
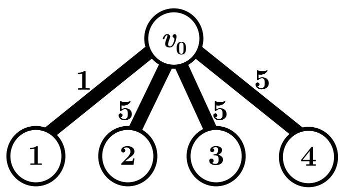


Figure 2.3: Graph G.

Example 2.4. Consider the graph G displayed in Figure 2.3. Assume that all the edges that are not shown have cost 100. The tree displayed in bold is the MCST and the SPT of this graph. The MCST has a total cost $1 + 1 + 2 + 2 = 6$ and the SPT has a cost $1+2+4+4 = 11$. Now, assume that we increase the cost of the tree edge $\{1, 2\}$ to 100. Spira and Pan (1973) argue that to construct an updated MCST, we consider the two subtrees that are formed if this edge is excluded. Then, we search for the minimum cost link between those two subtrees. For this example, an MCST of the updated problem can be seen in Figure 2.4(a) with a total cost of $1 + 5 + 2 + 2 = 10$. On the other hand, the authors have shown that in general constructing an updated SPT in the case of a tree edge cost increase is as hard as resolving a new SPT problem. For this example, the updated SPT tree can be seen in Figure 2.4(b) with a total cost of $1 + 5 + 5 + 5 = 16$. $\diamond$



(a) An updated MCST



(b) The updated SPT

Figure 2.4: An updated MCST and the updated SPT after the tree edge cost increase

These differences imply that known results for MCST problems and games do not directly yield any properties for SPT problems and games.

2.2 Properties of the Shortest Path Trees

This section introduces some definition and notation, and presents a number of properties of shortest path trees that will be used throughout this thesis.

Let $((N \cup \{v_0\}, E), t)$ be an SPT problem. Let $F(i, \mathcal{T})$ denote *the set of followers of an agent* i on $\mathcal{T}$ such that $F(i, \mathcal{T}) = \{j : i \in V(j, \mathcal{T})\}$ where $V(j, \mathcal{T})$ is the set of agents on the shortest path of j on $\mathcal{T}$, excluding v_0 but including j. Note that $i \in F(i, \mathcal{T})$. An agent i is called a *hub* of $\mathcal{T}$ if $V(i, \mathcal{T}) = \{i\}$. Thus, hubs are agents that are directly connected to v_0 on $\mathcal{T}$. Let $H(\mathcal{T}) = \{i : V(i, \mathcal{T}) = \{i\}\}$ denote the set of hubs of $\mathcal{T}$. Let $h \in H(\mathcal{T})$, then $B_h(\mathcal{T}) = F(h, \mathcal{T})$ is called *the branch of* N *induced by hub* h *of* $\mathcal{T}$. Observe that $\{B_h(\mathcal{T}) : h \in H(\mathcal{T})\}$ is a partition of N. We refer to this partition as *the branch partition of* N *with respect to* $\mathcal{T}$. The *hub of an agent* i on $\mathcal{T}$, denoted by $Hub(i, \mathcal{T})$, is defined as $Hub(i, \mathcal{T}) = h$ if $i \in B_h(\mathcal{T})$. An agent i is called a *leaf* of $\mathcal{T}$ if $F(i, \mathcal{T}) = \{i\}$. Let $L(\mathcal{T}) = \{i : F(i, \mathcal{T}) = \{i\}\}$ denote the set of leaves of $\mathcal{T}$.

Example 2.5. Let us consider the shortest path tree $\mathcal{T}$ illustrated in Figure 2.5.

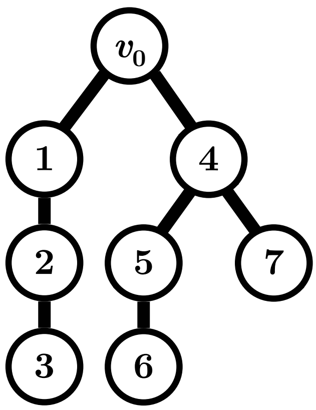


Figure 2.5: A shortest path tree.

We have $H(\mathcal{T}) = \{1, 4\}$ and $L(\mathcal{T}) = \{3, 6, 7\}$. Moreover, $F(5, \mathcal{T}) = \{5, 6\}$ and $F(1, \mathcal{T}) = \{1, 2, 3\}$. The two branches of N with respect to $\mathcal{T}$, which form a partition of N, are $B_1(\mathcal{T}) = \{1, 2, 3\}$ and $B_4(\mathcal{T}) = \{4, 5, 6, 7\}$. We have $Hub(3, \mathcal{T}) = 1$ and $Hub(5, \mathcal{T}) = 4$. ◇

We further introduce some notation, which will be used for the rest of this chapter. Consider the SPT problem $(((N\backslash\{i\}) \cup \{v_0\}, E^{N\backslash\{i\}}), t^{N\backslash\{i\}})$. We denote an optimal solution to this problem by $\mathcal{T}^{-i}$. Moreover, we let $\mathcal{T}^{-S}$ denote a shortest path tree of the SPT problem $(((N\backslash S) \cup \{v_0\}, E^{N\backslash S}), t^{N\backslash S})$. Therefore, $\mathcal{T}^{-i}$ and $\mathcal{T}^{-S}$ replace $\mathcal{T}^{N\backslash\{i\}}$ and $\mathcal{T}^{N\backslash S}$ for convenience of notation.

Next, we present a number of properties of shortest path trees. First, consider an SPT problem and a shortest path tree $\mathcal{T}$. Furthermore, consider a modified SPT problem that arises when an agent and all its followers on $\mathcal{T}$ are removed from the original SPT problem. The following lemma states that there always exists a shortest path tree of the modified SPT problem such that the set of edges on the shortest path of all the agents in the modified SPT problem is the same set of edges on their shortest path on $\mathcal{T}$.

Lemma 2.1. *Let* $((N \cup \{v_0\}, E), t)$ *be an SPT problem and let* $\mathcal{T}$ *be a shortest path tree. Let* $i \in N$. *Let* $(((N\backslash F(i,\mathcal{T})) \cup \{v_0\}, E^{N\backslash F(i,\mathcal{T})}), t^{N\backslash F(i,\mathcal{T})})$ *be an SPT problem. Then there exists a shortest path tree* $\mathcal{T}^{-F(i,\mathcal{T})}$ *such that* $E(j, \mathcal{T}^{-F(i,\mathcal{T})}) = E(j, \mathcal{T})$ *for all* $j \in N\backslash F(i,\mathcal{T})$.

Proof: Let $j \in N\backslash F(i,\mathcal{T})$. Then $V(j,\mathcal{T}) \subseteq N\backslash F(i,\mathcal{T})$. Therefore, there exists a $\mathcal{T}^{-F(i,\mathcal{T})}$ such that $E(j, \mathcal{T}^{-F(i,\mathcal{T})}) = E(j,\mathcal{T})$ for all $j \in N\backslash F(i,\mathcal{T})$. □

We further consider the SPT problem $((B_h(\mathcal{T}) \cup \{v_0\}, E^{B_h(\mathcal{T})}), t^{B_h(\mathcal{T})})$ where $B_h(\mathcal{T})$ is the branch of N induced by hub h of $\mathcal{T}$. We have the next corollary stating that there exists a shortest path tree formed by the agents in $B_h(\mathcal{T})$ on which the set of edges on the shortest path to an agent i is the set of edges on the shortest path to i on the shortest path tree formed by all the agents in N. The corollary follows from Lemma 2.1 by removing the followers of the hubs in $H(\mathcal{T})\backslash\{h\}$, in other words, by removing all the branches of N induced by $H(\mathcal{T})\backslash\{h\}$ one by one.

Corollary 2.1. *Let* $((N \cup \{v_0\}, E), t)$ *be an SPT problem and let* $\mathcal{T}$ *be a shortest path tree. Let* $h \in H(\mathcal{T})$. *Let* $((B_h(\mathcal{T}) \cup \{v_0\}, E^{B_h(\mathcal{T})}), t^{B_h(\mathcal{T})})$ *be an SPT problem. Then there exists a shortest path tree* $\mathcal{T}^{B_h(\mathcal{T})}$ *such that*

$$E(i, \mathcal{T}^{B_h(\mathcal{T})}) = E(i, \mathcal{T}) \text{ for all } i \in B_h(\mathcal{T}).$$

We call the tree $\mathcal{T}^{B_h(\mathcal{T})}$ that satisfies $E(i, \mathcal{T}^{B_h(\mathcal{T})}) = E(i,\mathcal{T})$ for all $i \in B_h(\mathcal{T})$ *the tree branch of* $\mathcal{T}$ *induced by hub* $h \in H(\mathcal{T})$. Next, we have a lemma stating that the cost of a shortest path tree $\mathcal{T}$ is equal to the sum of the costs of its tree branches induced by the branch partition of N with respect to $\mathcal{T}$.

Lemma 2.2. *Let $((N \cup \{v_0\}, E), t)$ be an SPT problem and let $\mathcal{T}$ be a shortest path tree. Let $\mathcal{T}^{B_h(\mathcal{T})}$ be the tree branch of $\mathcal{T}$ induced by hub $h \in H(\mathcal{T})$. Then we have*

$$t(\mathcal{T}) = \sum_{h \in H(\mathcal{T})} t(\mathcal{T}^{B_h(\mathcal{T})}).$$

Proof: We get

$$t(\mathcal{T}) = \sum_{i \in N} t^{E(i,\mathcal{T})} = \sum_{h \in H(\mathcal{T})} \sum_{i \in B_h(\mathcal{T})} t^{E(i,\mathcal{T})} = \sum_{h \in H(\mathcal{T})} t(\mathcal{T}^{B_h(\mathcal{T})})$$

where the last equality follows from Corollary 2.1. □

The following lemma states that given an SPT problem, the shortest path cost of an agent i never increases if new agents enter the problem.

Lemma 2.3. *Let $S \subseteq T \subseteq N$. Let $((S \cup \{v_0\}, E^S), t^S)$ and $((T \cup \{v_0\}, E^T), t^T)$ be SPT problems, and let $\mathcal{T}^S$ and $\mathcal{T}^T$ be shortest path trees. Let $i \in S$. Then*

$$t^{E(i,\mathcal{T}^T)} \leq t^{E(i,\mathcal{T}^S)}.$$

Proof: Let $E(i, \mathcal{T}^S) = \{\{v_0, i_1\}, \{i_1, i_2\}, \ldots, \{i_{k-1}, i_k\}, \{i_k, i\}\}$ be the set of edges on the shortest path to i on $\mathcal{T}^S$. Since $\{i_1, i_2, \ldots, i_k, i\} \subseteq S \subseteq T$, we have $E(i, \mathcal{T}^S) \subseteq E^T$. Therefore, the edges in $E(i, \mathcal{T}^S)$ are on a path to i whereas the edges in $E(i, \mathcal{T}^T)$ are on a shortest path to i in the SPT problem $((T \cup \{v_0\}, E^T), t^T)$. Hence, $t^{E(i,\mathcal{T}^T)} \leq t^{E(i,\mathcal{T}^S)}$. □

2.3 Cost Allocation in the Shortest Path Tree Problems

In this section, we address the issue of allocating the cost of a shortest path tree amongst the agents. In Section 2.3.1, we propose three different cost allocation methods. Firstly, we consider a shortest path tree formed by all the agents and define the *tree solution* θ by allocating to every agent the cost of its shortest path. Secondly, we consider a shortest path tree formed by all the agents excluding an agent i and define the δ^i rule by allocating to each of these agents the cost of its shortest path on this tree and by allocating to agent i its marginal contribution. Thirdly, we consider an allocation rule γ that is the average of δ^i for all agents i. In Section 2.3.2, we consider a number of properties of cost allocation rules and identify the properties satisfied by the aforementioned allocation rules. Finally, we present a characterisation of the tree solution θ.

2.3.1 Cost Allocation Rules

This section proposes three cost allocation rules for sharing the cost of a shortest path tree amongst the agents.

Let $\mathcal{SPT}(N)$ denote the class of all SPT problems corresponding to the agent set N. A cost allocation rule is a function $\psi : \mathcal{SPT}(N) \to \mathbb{R}^N$. The cost allocated to an agent i by cost allocation rule ψ is denoted by ψ_i.

We start with the allocation rule where each agent i is assigned the cost of its shortest path on $\mathcal{T}$. This allocation is similar to the Bird rule for the minimum cost spanning tree problems (Bird, 1976). We call this the *tree solution* since it is derived from a shortest path tree $\mathcal{T}$ of a given SPT problem.

Definition 2.1. *Let* $((N \cup \{v_0\}, E), t)$ *be an SPT problem and let* $\mathcal{T}$ *be a shortest path tree. Then the* tree solution θ *is defined as*

$$\theta_i\big(((N \cup \{v_0\}, E), t)\big) = t^{E(i,\mathcal{T})} \text{ for all } i \in N.$$

Note that for an SPT problem, the tree solution is unique even when $\mathcal{T}$ is not unique since the cost of the shortest path of an agent i is the same on all shortest path trees.

Example 2.6. Consider the SPT problem displayed in Figure 2.6. The tree solution is $\theta\big(((N \cup \{v_0\}, E), t)\big) = (2, 3, 5)$. $\diamond$

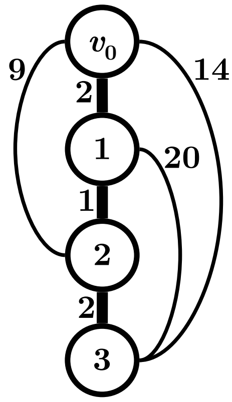


Figure 2.6: An SPT problem.

We next introduce for each $i \in N$ an allocation rule, δ^i. For the allocation $\delta^i\big(((N \cup \{v_0\}, E), t)\big)$ each agent $j \in N\backslash\{i\}$ pays its shortest path cost on the tree formed by all the agents excluding agent i and agent i pays its marginal contribution.

Definition 2.2. *Let $((N \cup \{v_0\}, E), t)$ be an SPT problem and let $\mathcal{T}$ be a shortest path tree. Let $i \in N$. Then the δ^i rule is defined as*

$$\delta^i_j\big(((N \cup \{v_0\}, E), t)\big) = \begin{cases} t^{E(j,\mathcal{T}^{-i})} & \textit{for } j \in N\backslash\{i\}, \\ t(\mathcal{T}) - t(\mathcal{T}^{-i}) & \textit{for } j = i. \end{cases}$$

The δ^i rule is defined for each agent $i \in N$. We next propose a cost allocation rule that is the average of δ^i for all $i \in N$.

Definition 2.3. *Let $((N \cup \{v_0\}, E), t)$ be an SPT problem. Then the γ rule is defined as*

$$\gamma\big(((N \cup \{v_0\}, E), t)\big) = \frac{1}{n} \sum_{i \in N} \delta^i\big(((N \cup \{v_0\}, E), t)\big).$$

Example 2.7. Consider the SPT problem in Figure 2.6, we have $t(\mathcal{T}) = 10$. For the cost allocation $\delta^1\big(((N \cup \{v_0\}, E), t)\big)$, agents 2 and 3 are allocated the cost of their shortest paths on $\mathcal{T}^{-1}$, which equal to $t^{E(2,\mathcal{T}^{-1})} = 9$ and $t^{E(3,\mathcal{T}^{-1})} = 11$ respectively. We have $t(\mathcal{T}^{-1}) = 20$. The surplus of $t(\mathcal{T}) - t(\mathcal{T}^{-1}) = 10 - 20 = -10$ is allocated to agent 1 giving us $\delta^1\big(((N \cup \{v_0\}, E), t)\big) = (-10, 9, 11)$. Furthermore, we have $\delta^2\big(((N \cup \{v_0\}, E), t)\big) = (2, -6, 14)$ and $\delta^3\big(((N \cup \{v_0\}, E), t)\big) = (2, 3, 5)$. Since the γ rule is the average of these three allocations, $\gamma\big(((N \cup \{v_0\}, E), t)\big) = (-2, 2, 10)$. $\diamond$

The next proposition states that the tree solution is equal to the δ^i rule when i is a leaf agent.

Proposition 2.1. *Let $((N \cup \{v_0\}, E), t)$ be an SPT problem and let $\mathcal{T}$ to be a shortest path tree. Let $L(\mathcal{T})$ denote the set of leaves of $\mathcal{T}$. Then*

$$\theta((N \cup \{v_0\}, E), t) = \delta^l((N \cup \{v_0\}, E), t) \text{ for all } l \in L(\mathcal{T}).$$

Proof: We have $L(\mathcal{T}) \neq \emptyset$. Consider the allocation $\delta^l((N \cup \{v_0\}, E), t)$. Since l is a leaf, none of the agents in $N\backslash\{l\}$ have l on their shortest path on $\mathcal{T}$. For all $j \in N\backslash\{l\}$, we get

$$\delta^l_j((N \cup \{v_0\}, E), t) = t^{E(j,\mathcal{T}^{-l})} = t^{E(j,\mathcal{T})} = \theta_j((N \cup \{v_0\}, E), t).$$

Since $\sum_{j\in N} \theta_j((N\cup\{v_0\}, E), t) = t(\mathcal{T})$ and $\sum_{j\in N} \delta^l_j((N\cup\{v_0\}, E), t) = t(\mathcal{T})$ by the definitions of these allocation rules, $\delta^l_l((N \cup \{v_0\}, E), t) = \theta_l((N \cup \{v_0\}, E), t)$. Thus, we have $\theta((N \cup \{v_0\}, E), t) = \delta^l((N \cup \{v_0\}, E), t)$. $\square$

Example 2.8. Consider the SPT problem in Figure 2.6 where agent 3 is a leaf and we have $\delta^3\big(((N \cup \{v_0\}, E), t)\big) = \theta\big(((N \cup \{v_0\}, E), t)\big) = (2, 3, 5)$. $\diamond$

Alternative Definitions of the γ Rule

In this section, we present two alternative definitions of the γ rule, which we have defined as the average of δ^i for all $i \in N$.

For the first definition, we define the *maximum claims of i from j*, denoted by m_{ij}, to be the difference between the shortest path cost of j on $\mathcal{T}$ and the shortest path cost of j on $\mathcal{T}^{-i}$. That is, $m_{ij} = t^{E(j,\mathcal{T})} - t^{E(j,\mathcal{T}^{-i})}$ for $i \neq j$, $i, j \in N$ and $m_{ii} = 0$. Note that, from Lemma 2.3, which states that the shortest path cost of an agent never increases if new agents enter the problem, we have $m_{ij} \leq 0$ for all $i, j \in N$. Starting from the definition of the γ rule, for $i \in N$, we derive

$$
\begin{aligned}
\gamma_i\big(((N\cup\{v_0\},E),t)\big) &= \frac{1}{n}\sum_{j\in N}\delta_i^j\big(((N\cup\{v_0\},E),t)\big)\\
&= \frac{1}{n}\left[\delta_i^i\big(((N\cup\{v_0\},E),t)\big) + \sum_{j\in N\setminus\{i\}}\delta_i^j\big(((N\cup\{v_0\},E),t)\big)\right]\\
&= \frac{1}{n}\left[t(\mathcal{T}) - t(\mathcal{T}^{-i}) + \sum_{j\in N\setminus\{i\}} t^{E(i,\mathcal{T}^{-j})}\right]\\
&= \frac{1}{n}\left[\sum_{j\in N} t^{E(j,\mathcal{T})} - \sum_{j\in N\setminus\{i\}} t^{E(j,\mathcal{T}^{-i})} + \sum_{j\in N\setminus\{i\}} t^{E(i,\mathcal{T}^{-j})}\right]\\
&= \frac{1}{n}\left[\sum_{j\in N} t^{E(j,\mathcal{T})} - \sum_{j\in N\setminus\{i\}} t^{E(j,\mathcal{T}^{-i})} + \sum_{j\in N\setminus\{i\}} t^{E(i,\mathcal{T}^{-j})}\right]\\
&\quad + \frac{1}{n}\left[t^{E(i,\mathcal{T}^{-i})} - t^{E(i,\mathcal{T}^{-i})}\right]\\
&= \frac{1}{n}\sum_{j\in N}\left[t^{E(j,\mathcal{T})} - t^{E(j,\mathcal{T}^{-i})} + t^{E(i,\mathcal{T}^{-j})}\right]\\
&= t^{E(i,\mathcal{T})} - t^{E(i,\mathcal{T})} + \frac{1}{n}\sum_{j\in N}\left[t^{E(j,\mathcal{T})} - t^{E(j,\mathcal{T}^{-i})} + t^{E(i,\mathcal{T}^{-j})}\right]\\
&= t^{E(i,\mathcal{T})} + \frac{1}{n}\sum_{j\in N}\left[-t^{E(i,\mathcal{T})} + t^{E(j,\mathcal{T})} - t^{E(j,\mathcal{T}^{-i})} + t^{E(i,\mathcal{T}^{-j})}\right]\\
&= t^{E(i,\mathcal{T})} + \frac{1}{n}\sum_{j\in N}\left[t^{E(j,\mathcal{T})} - t^{E(j,\mathcal{T}^{-i})}\right] - \frac{1}{n}\sum j\in N\left[t^{E(i,\mathcal{T})} - t^{E(i,\mathcal{T}^{-j})}\right]\\
&= t^{E(i,\mathcal{T})} + \frac{1}{n}\sum_{j\in N} m_{ij} - \frac{1}{n}\sum_{j\in N} m_{ji}.
\end{aligned}
$$

We get the following definition of the γ rule:

$$
\gamma_i\big(((N\cup\{v_0\},E),t)\big) = t^{E(i,\mathcal{T})} + \frac{1}{n}\sum_{j\in N} m_{ij} - \frac{1}{n}\sum_{j\in N} m_{ji} \quad \text{for all } i\in N.
$$

Therefore, the γ rule is defined as an adjustment on the shortest path cost of an agent. First, the γ rule allocates their shortest path cost to all of the agents. Then each agent receives their average maximum claims since $m_{ij} \leq 0$ and pays their average maximum claims to all other agents.

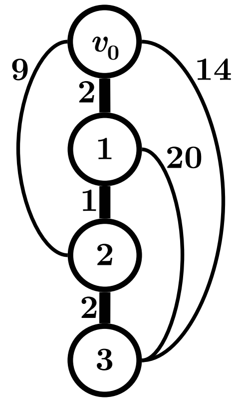


Figure 2.7: An SPT problem.

Example 2.9. Consider the SPT problem displayed in Figure 2.7. We have

$$\begin{aligned}
m_{11} &= 0 \text{ (by definition)} \\
m_{12} &= t^{E(2,\mathcal{T})} - t^{E(2,\mathcal{T}^{-1})} = 3 - 9 = -6 \\
m_{13} &= t^{E(3,\mathcal{T})} - t^{E(3,\mathcal{T}^{-1})} = 5 - 11 = 0 \\
m_{21} &= t^{E(1,\mathcal{T})} - t^{E(1,\mathcal{T}^{-2})} = 2 - 2 = 0 \\
m_{22} &= 0 \text{ (by definition)} \\
m_{23} &= t^{E(3,\mathcal{T})} - t^{E(3,\mathcal{T}^{-2})} = 5 - 14 = -9 \\
m_{31} &= t^{E(1,\mathcal{T})} - t^{E(1,\mathcal{T}^{-3})} = 2 - 2 = 0 \\
m_{32} &= t^{E(2,\mathcal{T})} - t^{E(2,\mathcal{T}^{-3})} = 3 - 3 = 0 \\
m_{33} &= 0 \text{ (by definition).}
\end{aligned}$$

Since $t^{E(1,\mathcal{T})} = 2$, $t^{E(2,\mathcal{T})} = 3$ and $t^{E(3,\mathcal{T})} = 5$, we get

$$\begin{aligned}
\gamma_1 &= 2 + \frac{1}{3}[0 - 6 - 6] - \frac{1}{3}[0 + 0 + 0] = -2 \\
\gamma_2 &= 3 + \frac{1}{3}[0 + 0 - 9] - \frac{1}{3}[-6 + 0 + 0] = 2 \\
\gamma_3 &= 5 + \frac{1}{3}[-6 - 9 + 0] - \frac{1}{3}[0 + 0 + 0] = 10
\end{aligned}$$

which was shown to be the γ allocation for this SPT problem in Example 2.7. $\diamond$

For the second definition, we define the *power of* i *over* j, denoted by p_{ij}, to be the difference of the cost of a shortest path to i when j is not present and the cost of a shortest path to j when i is not present. That is, $p_{ij} = t^{E(i,\mathcal{T}^{-j})} - t^{E(j,\mathcal{T}^{-i})}$ for $i, j \in N$. Observe that $p_{ij} = -p_{ji}$. Note furthermore that the lower the value of p_{ij} the more powerful agent i is over j. Starting from the definition of the γ rule, for $i \in N$, we derive

$$\begin{aligned}
\gamma_i\big(((N \cup \{v_0\}, E), t)\big) &= \frac{1}{n}\sum_{j\in N} \delta_i^j\big(((N \cup \{v_0\}, E), t)\big) \\
&= \frac{1}{n}\left[\delta_i^i\big(((N \cup \{v_0\}, E), t)\big) + \sum_{j\in N\setminus\{i\}} \delta_i^j\big(((N \cup \{v_0\}, E), t)\big)\right] \\
&= \frac{1}{n}\left[t(\mathcal{T}) - t(\mathcal{T}^{-i}) + \sum_{j\in N\setminus\{i\}} t^{E(i,\mathcal{T}^{-j})}\right] \\
&= \frac{1}{n}\left[t(\mathcal{T}) - \sum_{j\in N\setminus\{i\}} t^{E(j,\mathcal{T}^{-i})} + \sum_{j\in N\setminus\{i\}} t^{E(i,\mathcal{T}^{-j})}\right] \\
&= \frac{t(\mathcal{T})}{n} + \frac{1}{n}\sum_{j\in N\setminus\{i\}} \left[t^{E(i,\mathcal{T}^{-j})} - t^{E(j,\mathcal{T}^{-i})}\right] \\
&= \frac{t(\mathcal{T})}{n} + \frac{1}{n}\sum_{j\in N\setminus\{i\}} \left[t^{E(i,\mathcal{T}^{-j})} - t^{E(j,\mathcal{T}^{-i})}\right] + \frac{1}{n}\left[t^{E(i,\mathcal{T}^{-i})} - t^{E(i,\mathcal{T}^{-i})}\right] \\
&= \frac{t(\mathcal{T})}{n} + \frac{1}{n}\sum_{j\in N} \left[t^{E(i,\mathcal{T}^{-j})} - t^{E(j,\mathcal{T}^{-i})}\right] \\
&= \frac{t(\mathcal{T})}{n} + \frac{1}{n}\sum_{j\in N} p_{ij}.
\end{aligned}$$

We get the following definition of the γ rule:

$$\gamma_i\big(((N \cup \{v_0\}, E), t)\big) = \frac{t(\mathcal{T})}{n} + \frac{1}{n}\sum_{j\in N} p_{ij} \quad \text{for all } i \in N.$$

Therefore, the γ rule is defined as an adjustment on the egalitarian allocation where each agent is assigned an equal share of the shortest path tree cost. First, the γ rule allocates an equal share of the cost to all of the agents. Then the cost of each agent is adjusted based on its *average power over* N, which is defined as $\mathcal{P}_i = \frac{1}{n}\sum_{j\in N} p_{ij}$. That is, if $\mathcal{P}_i > 0$ and the agent does not have much power over others it pays more than the egalitarian allocation and if $\mathcal{P}_i < 0$ and the agent has large amounts of power over others it pays less than the egalitarian allocation.

Example 2.10. Consider the SPT problem displayed in Figure 2.7. We have

$$\begin{aligned}
p_{11} &= t^{E(1,\mathcal{T}^{-1})} - t^{E(1,\mathcal{T}^{-1})} = 0\\
p_{12} &= t^{E(1,\mathcal{T}^{-2})} - t^{E(2,\mathcal{T}^{-1})} = 2 - 9 = -7 = -p_{21}\\
p_{13} &= t^{E(1,\mathcal{T}^{-3})} - t^{E(3,\mathcal{T}^{-1})} = 2 - 11 = -9 = -p_{31}\\
p_{22} &= t^{E(2,\mathcal{T}^{-2})} - t^{E(2,\mathcal{T}^{-2})} = 0\\
p_{23} &= t^{E(2,\mathcal{T}^{-3})} - t^{E(3,\mathcal{T}^{-2})} = 3 - 14 = -11 = -p_{32}\\
p_{33} &= t^{E(3,\mathcal{T}^{-3})} - t^{E(3,\mathcal{T}^{-3})} = 0.
\end{aligned}$$

Since $t(\mathcal{T}) = 10$, we get

$$\begin{aligned}
\gamma_1 &= \frac{10}{3} + \frac{1}{3}[0 - 7 - 9] = -2\\
\gamma_2 &= \frac{10}{3} + \frac{1}{3}[7 + 0 - 11] = 2\\
\gamma_3 &= \frac{10}{3} + \frac{1}{3}[9 + 11 + 0] = 10
\end{aligned}$$

which was shown to be the γ allocation for this SPT problem in Example 2.7. $\diamond$

2.3.2 Properties of the Cost Allocation Rules

We start this section by presenting a set of properties of a cost allocation rule $\psi : \mathcal{SPT}(N) \to \mathbb{R}^N$ where ψ_i denotes the cost allocated to agent i. We then check which of these properties are satisfied by the cost allocation rules defined in the previous section. Finally, we present a characterisation of the tree solution θ.

Definition of the Properties of the Cost Allocation Rules and Related Results

This section introduces a number of properties of cost allocation rules and presents a number of results for shortest path trees related to these properties.

Efficiency

A cost allocation is *efficient* if the total cost to be shared amongst all the agents is exactly the cost of a shortest path tree $\mathcal{T}$. In other words, no excess over total cost is charged and no outside resource is required to cover any part of the costs.

Definition 2.4. *Let* $((N \cup \{v_0\}, E), t)$ *be an SPT problem and let* $\mathcal{T}$ *be a shortest path tree. Then a cost allocation rule* ψ *is* efficient *if*

$$\sum_{i \in N} \psi_i\big(((N \cup \{v_0\}, E), t)\big) = t(\mathcal{T}).$$

Branch Efficiency

For the SPT problem, we introduce an efficiency property defined on the branch partition of N with respect to $\mathcal{T}$. We call it the *branch efficiency* property. A cost allocation rule is branch efficient with respect to $\mathcal{T}$ if the sum of allocations to the agents of the branch $B_h(\mathcal{T})$ is equal to the cost of the tree branch $\mathcal{T}^{B_h(\mathcal{T})}$ of $\mathcal{T}$ induced by hub $h \in H(\mathcal{T})$.

Definition 2.5. *Let* $((N \cup \{v_0\}, E), t)$ *be an SPT problem and let* $\mathcal{T}$ *be a shortest path tree. Let* $\mathcal{T}^{B_h(\mathcal{T})}$ *be the tree branch of* $\mathcal{T}$ *induced by hub* $h \in H(\mathcal{T})$*. Then a cost allocation rule* ψ *is* branch efficient with respect to $\mathcal{T}$ *if*

$$\sum_{i \in B_h(\mathcal{T})} \psi_i\big(((N \cup \{v_0\}, E), t)\big) = t(\mathcal{T}^{B_h(\mathcal{T})}) \textit{ for all } h \in H(\mathcal{T}).$$

Branch efficiency has appeared in Megiddo (1978) as a decomposition property for fixed trees.

Next, we show that a cost allocation rule that satisfies branch efficiency also satisfies efficiency.

Proposition 2.2. *Let* $((N \cup \{v_0\}, E), t)$ *be an SPT problem and let* $\mathcal{T}$ *be a shortest path tree. Then if a cost allocation rule* ψ *is branch efficient with respect to* $\mathcal{T}$ *then it is efficient.*

Proof: Let ψ be branch efficient with respect to $\mathcal{T}$. Then we have

$$\sum_{i \in N} \psi_i\big(((N \cup \{v_0\}, E), t)\big) = \sum_{h \in H(\mathcal{T})} \sum_{i \in B_h(\mathcal{T})} \psi_i\big(((N \cup \{v_0\}, E), t)\big)$$

$$= \sum_{h \in H(\mathcal{T})} t(\mathcal{T}^{B_h(\mathcal{T})})$$
$$= t(\mathcal{T})$$

where the last equality follows from Lemma 2.2. Thus, ψ is efficient. □

Symmetry

Now, we turn our attention to a different property. The *symmetry* property states that a symmetric cost allocation rule assigns the same cost to two symmetric agents i and j. The notion of symmetric agents and the symmetry property that are introduced next are inspired by Bergantiños and Vidal-Puga (2007) for the minimum cost spanning tree problems. For an SPT problem $((N \cup \{v_0\}, E), t)$, agents i and j are said to be *symmetric* if for all $k \in N \cup \{v_0\}$, $t_{ik} = t_{jk}$.

Definition 2.6. *Let $((N \cup \{v_0\}, E), t)$ be an SPT problem. Let $i, j \in N$ be symmetric agents. Then a cost allocation rule ψ is* symmetric *if*

$$\psi_i\big(((N \cup \{v_0\}, E), t)\big) = \psi_j\big(((N \cup \{v_0\}, E), t)\big).$$

We have the following lemma stating that on a shortest path tree formed by a subset of the agents, such that symmetric agents i and j are both in this subset, the cost of the shortest path of i and of j are the same.

Lemma 2.4. *Let $S \subseteq N$. Let $((S \cup \{v_0\}, E^S), t^S)$ be an SPT problem and let $\mathcal{T}^S$ be a shortest path tree. Let $i, j \in S$ be symmetric agents. Then $t^{E(i,\mathcal{T}^S)} = t^{E(j,\mathcal{T}^S)}$.*

Proof: Let $E(j, \mathcal{T}^S) = \{\{v_0, j_1\}, \{j_1, j_2\}, \ldots, \{j_{k-1}, j_k\}, \{j_k, j\}\}$ be the set of edges on the shortest path to j on $\mathcal{T}^S$. If $i \in \{j_1, j_2, \ldots, j_k\}$, then $t^{E(j,\mathcal{T}^S)} \geq t^{E(i,\mathcal{T}^S)}$. If $i \notin \{j_1, j_2, \ldots, j_k\}$, then there exists a path to i consisting of edges $E(i) = \{\{v_0, j_1\}, \{j_1, j_2\}, \ldots, \{j_{k-1}, j_k\}, \{j_k, i\}\}$ since i and j are symmetric, which satisfies $t^{E(i)} = t^{E(j,\mathcal{T}^S)}$. Since $E(i)$ is a path and $E(i, \mathcal{T}^S)$ is a shortest path, we get $t^{E(i)} \geq t^{E(i,\mathcal{T}^S)}$. Thus, $t^{E(j,\mathcal{T}^S)} \geq t^{E(i,\mathcal{T}^S)}$. Since i and j are symmetric, interchanging them in the above argument gives us $t^{E(i,\mathcal{T}^S)} \geq t^{E(j,\mathcal{T}^S)}$. Hence, $t^{E(j,\mathcal{T}^S)} = t^{E(i,\mathcal{T}^S)}$. □

Strong Cost Monotonicity

Next, we consider a monotonicity property called *strong cost monotonicity* inspired by Bergantiños and Vidal-Puga (2004). This property implies that if a subset of edge costs decrease, and the cost of any remaining edges stay the same, then no agent should be allocated a higher cost.

Definition 2.7. *Let* $((N \cup \{v_0\}, E), t)$ *and* $((N \cup \{v_0\}, E), \bar{t})$ *be SPT problems where* t *and* $\bar{t}$ *are two cost functions such that* $t > \bar{t}$. *Then a cost allocation rule* ψ *is* strong cost monotone *if*

$$\psi_i\big(((N \cup \{v_0\}, E), t)\big) \geq \psi_i\big(((N \cup \{v_0\}, E), \bar{t})\big) \text{ for all } i \in N.$$

The following lemma states that the cost of a shortest path of an agent never increases if the cost of an edge decreases. The proof of this lemma is omitted since the result follows trivially.

Lemma 2.5. *Let* $S \subseteq N$. *Let* $((S \cup \{v_0\}, E^S), t^S)$ *and* $((S \cup \{v_0\}, E^S), \bar{t}^S)$ *be SPT problems, and let* $\mathcal{T}^S$ *and* $\overline{\mathcal{T}}^S$ *be corresponding shortest path trees. Let* $i, j \in S \cup \{v_0\}$. *Let* t *and* $\bar{t}$ *be two cost functions such that* $t_{kl} = \bar{t}_{kl}$ *for all* $k, l \in S \cup \{v_0\}$, $\{k, l\} \neq \{i, j\}$ *and* $t_{ij} > \bar{t}_{ij}$. *Let* $t^{E(m,\mathcal{T}^S)} = \sum_{\{p,q\} \in E(m,\mathcal{T}^S)} t^S_{pq}$ *and* $\bar{t}^{E(m,\overline{\mathcal{T}}^S)} = \sum_{\{p,q\} \in E(m,\overline{\mathcal{T}}^S)} \bar{t}^S_{pq}$. *Then we have* $t^{E(m,\mathcal{T}^S)} \geq \bar{t}^{E(m,\overline{\mathcal{T}}^S)}$ *for all* $m \in S$.

Power Monotonicity

We define a monotonicity property for the SPT problems. Recall that $p_{ij} = t^{E(i,\mathcal{T}^{-j})} - t^{E(j,\mathcal{T}^{-i})}$ is the power of i over j for $i, j \in N$ and $\mathcal{P}_i = \frac{1}{n}\sum_{j \in N} p_{ij}$ is the average power of an agent i over N. An allocation rule is *power monotone* if the cost it allocates to the more powerful agent i is lower than the cost it allocates to the less powerful agent j such that $\mathcal{P}_i \leq \mathcal{P}_j$.

Definition 2.8. *Let* $((N \cup \{v_0\}, E), t)$ *be an SPT problem. Let* $i, j \in N$ *such that* $\mathcal{P}_i \leq \mathcal{P}_j$. *Then a cost allocation rule* ψ *is* power monotone *if*

$$\psi_i\big(((N \cup \{v_0\}, E), t)\big) \leq \psi_j\big(((N \cup \{v_0\}, E), \bar{t})\big).$$

Leaf Consistency

Finally, we define a consistency property for the SPT problems. An allocation rule is *leaf consistent* if the cost it allocates to an agent is the same upon the removal of a leaf from the original SPT problem.

Definition 2.9. *Let* $((N \cup \{v_0\}, E), t)$ *be an SPT problem. Let* $l \in L(\mathcal{T})$ *be a leaf and let* $((N\backslash\{l\} \cup \{v_0\}, E^{N\backslash\{l\}}), t^{N\backslash\{l\}})$ *be an SPT problem. Then a cost allocation rule* ψ *is* leaf consistent *if*

$$\psi_i\big(((N \cup \{v_0\}, E), t)\big) = \psi_i\big(((N\backslash\{l\} \cup \{v_0\}, E^{N\backslash\{l\}}), t^{N\backslash\{l\}})\big) \text{ for all } i \in N\backslash\{l\}.$$

Properties Satisfied by the Proposed Cost Allocation Rules

In this section, we consider each of the cost allocation rules defined in Section 2.3.1 and identify the properties that they satisfy. We start with the tree solution θ.

Theorem 2.1. *Let* $((N \cup \{v_0\}, E), t)$ *be an SPT problem and let* $\mathcal{T}$ *be a shortest path tree. Then tree solution* $\theta\big(((N \cup \{v_0\}, E), t)\big)$ *is branch efficient with respect to* $\mathcal{T}$*, efficient, symmetric, strong cost monotone and leaf consistent.*

Proof: It is sufficient to prove that θ is branch efficient with respect to $\mathcal{T}$, symmetric, strong cost monotone and leaf consistent. Recall that the tree solution θ assigns to every agent the cost of its shortest path on $\mathcal{T}$. We get

$$\sum_{i \in B_h(\mathcal{T})} \theta_i\big(((N \cup \{v_0\}, E), t)\big) = \sum_{i \in B_h(\mathcal{T})} t^{E(i,\mathcal{T})} = t(\mathcal{T}^{B_h(\mathcal{T})}) \text{ for } h \in H(\mathcal{T})$$

where the last equality follows from Corollary 2.1. Therefore, θ is branch efficient with respect to $\mathcal{T}$.

Next, we show that θ is symmetric. Let $i, j \in N$ be symmetric agents. We have

$$\theta_i\big(((N \cup \{v_0\}, E), t)\big) = t^{E(i,\mathcal{T})} = t^{E(j,\mathcal{T})} = \theta_j\big(((N \cup \{v_0\}, E), t)\big).$$

where the second equality follows from Lemma 2.4 by setting $S = N$.
Now, we show that θ satisfies strong cost monotonicity. Let us consider SPT problems $((N \cup \{v_0\}, E), t)$ and $((N \cup \{v_0\}, E), \bar{t})$, and the corresponding shortest path trees $\mathcal{T}$ and $\overline{\mathcal{T}}$ where t and $\bar{t}$ are two cost functions such that $t > \bar{t}$. We get

$$\theta_k\big(((N \cup \{v_0\}, E), t)\big) = t^{E(k,\mathcal{T})} \geq \bar{t}^{E(k,\overline{\mathcal{T}})} = \theta_k\big(((N \cup \{v_0\}, E), \bar{t})\big) \text{ for all } k \in N$$

where the inequality follows from Lemma 2.5 by setting $S = N$.

Finally, we show that θ satisfies leaf consistency. We have

$$\begin{aligned}\theta_i\big(((N \cup \{v_0\}, E), t)\big) &= t^{E(i,\mathcal{T})}\\ &= t^{E(i,\mathcal{T}^{-l})}\\ &= \theta_i\big(((N\backslash\{l\} \cup \{v_0\}, E^{N\backslash\{l\}}), t^{N\backslash\{l\}})\big)\end{aligned}$$

for all $i \in N\backslash\{l\}$ where the second equality follows from Lemma 2.1 since for a leaf $l \in L(\mathcal{T})$ we have $F(l, \mathcal{T}) = \{l\}$. $\square$

We show that the tree solution for the SPT problems is not power monotone in the next example.

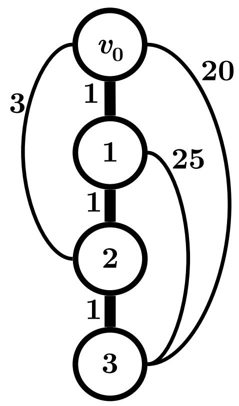


Figure 2.8: An SPT problem.

Example 2.11. Consider the SPT problem displayed in Figure 2.8. The tree solution $\theta\big(((N \cup \{v_0\}, E), t)\big) = (1, 2, 3)$. We have

$$\begin{aligned}p_{11} &= t^{E(1,\mathcal{T}^{-1})} - t^{E(1,\mathcal{T}^{-1})} = 0\\ p_{12} &= t^{E(1,\mathcal{T}^{-2})} - t^{E(2,\mathcal{T}^{-1})} = 2 - 3 = -1 = -p_{21}\\ p_{13} &= t^{E(1,\mathcal{T}^{-3})} - t^{E(3,\mathcal{T}^{-1})} = 2 - 4 = -2 = -p_{31}\\ p_{22} &= t^{E(2,\mathcal{T}^{-2})} - t^{E(2,\mathcal{T}^{-2})} = 0\\ p_{23} &= t^{E(2,\mathcal{T}^{-3})} - t^{E(3,\mathcal{T}^{-2})} = 3 - 20 = -17 = -p_{32}\\ p_{33} &= t^{E(3,\mathcal{T}^{-3})} - t^{E(3,\mathcal{T}^{-3})} = 0.\end{aligned}$$

Therefore, $\mathcal{P}_1 = \frac{0-1-2}{3} = -1$ and $\mathcal{P}_2 = \frac{1+0-17}{3} = -\frac{16}{3}$. We have $\theta_1\big(((N \cup \{v_0\}, E), t)\big) \leq \theta_2\big(((N \cup \{v_0\}, E), t)\big)$ where $\mathcal{P}_1 \geq \mathcal{P}_2$ and thus the tree solution is not power monotone. $\diamond$

Now, we present the properties of the δ^i rule.

Theorem 2.2. *Let $((N \cup \{v_0\}, E), t)$ be an SPT problem and let $\mathcal{T}$ be a shortest path tree. Let $i \in N$, then $\delta^i\big(((N \cup \{v_0\}, E), t)\big)$ is branch efficient with respect to $\mathcal{T}$, efficient and symmetric.*

Proof: Efficiency follows trivially from the definition of δ^i. Let us now show that this rule is branch efficient with respect to $\mathcal{T}$. Assume that $Hub(i, \mathcal{T}) = h^*$. Thus, we have $i, h^* \in B_{h^*}(\mathcal{T})$. The δ^i rule allocates the cost of the shortest path of j on $\mathcal{T}^{-i}$ to agents $j \in N \backslash \{i\}$. From Lemma 2.1, we know that the shortest path costs of the agents in branches $B_h(\mathcal{T})$ for all $h \in H(\mathcal{T})\backslash\{h^*\}$ do not change if i is removed, since they are not followers of i on $\mathcal{T}$. Hence,

$$\sum_{j \in B_h(\mathcal{T})} \delta^i_j\big(((N \cup \{v_0\}, E), t)\big) = \sum_{j \in B_h(\mathcal{T})} t^{E(j,\mathcal{T})} = t(\mathcal{T}^{B_h(\mathcal{T})}) \text{ for all } h \in H(\mathcal{T})\backslash\{h^*\}.$$

Since the δ^i rule is efficient, we have

$$\sum_{j \in B_{h^*}(\mathcal{T})} \delta^i_j\big(((N \cup \{v_0\}, E), t)\big) = t(\mathcal{T}) - \sum_{h \in H(\mathcal{T})\backslash\{h^*\}} t(\mathcal{T}^{B_h(\mathcal{T})}) = t(\mathcal{T}^{B_{h^*}(\mathcal{T})}).$$

Now, let us consider the symmetry property. Let $k, l \in N$ be symmetric agents. Let $i \in N$ be an agent and consider the corresponding δ^i rule. We discuss the following cases.

Case 1. $i \neq k, l$. We have

$$\delta_k^i\big(((N \cup \{v_0\}, E), t)\big) = t^{E(k,\mathcal{T}^{-i})} = t^{E(l,\mathcal{T}^{-i})} = \delta_l^i\big(((N \cup \{v_0\}, E), t)\big)$$

where the second equality follows from Lemma 2.4 by setting $S = N\backslash\{i\}$.

Case 2. $i = k$. First, we show that $t^{E(j,\mathcal{T})} = t^{E(j,\mathcal{T}^{-i})}$ for all the agents in $N\backslash\{i\}$. The agents that are not followers of i are not affected by the removal of agent i, and the agents that are followers of i, excluding itself, have a shortest path through agent l that is symmetric to i since $t_{ij} = t_{lj}$ for all $j \in N$. We have

$$\begin{aligned}
\delta_i^i\big(((N \cup \{v_0\}, E), t)\big) &= t(\mathcal{T}) - t(\mathcal{T}^{-i}) \\
&= \sum_{j \in N} t^{E(j,\mathcal{T})} - \sum_{j \in N\backslash\{i\}} t^{E(j,\mathcal{T}^{-i})} \\
&= t^{E(i,\mathcal{T})} - \sum_{j \in N\backslash\{i\}} \left[t^{E(j,\mathcal{T})} - t^{E(j,\mathcal{T}^{-i})}\right] \\
&= t^{E(i,\mathcal{T})} \\
&= t^{E(l,\mathcal{T})} \\
&= \delta_l^i\big(((N \cup \{v_0\}, E), t)\big)
\end{aligned}$$

where the fourth equality follows from $t^{E(j,\mathcal{T})} = t^{E(j,\mathcal{T}^{-i})}$ for all the agents in $N\backslash\{i\}$ and the fifth equality follows from Lemma 2.4 by setting $S = N$.

Based on the above discussion, we can conclude that δ^i is symmetric.

□

We show that the δ^i rule for the SPT problems is not leaf consistent and is not strong cost and power monotone in the next example.

Example 2.12. Consider the SPT problem in Figure 2.8 of Example 2.11. We have shown that $\mathcal{P}_1 \geq \mathcal{P}_2$. For this problem, $\delta^1\big(((N \cup \{v_0\}, E), t)\big) = (-1, 3, 4)$. We have $\delta_1^1\big(((N \cup \{v_0\}, E), t)\big) \leq \delta_2^1\big(((N \cup \{v_0\}, E), t)\big)$ where $\mathcal{P}_1 \geq \mathcal{P}_2$ and thus the δ^i rule is not power monotone.

Now, consider the SPT problem in Figure 2.9. For this problem, we have $\delta^1\big(((N \cup \{v_0\}, E), t)\big) = (-13, 9, 11, 12)$, $\delta^2\big(((N \cup \{v_0\}, E), t)\big) = (2, -9, 14, 12)$ and $\delta^3\big(((N \cup \{v_0\}, E), t)\big) = (2, 3, 2, 12)$ and $\delta^4\big(((N \cup \{v_0\}, E), t)\big) = (2, 3, 5, 9)$.

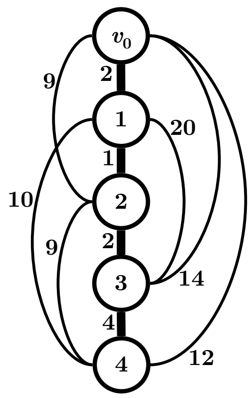


Figure 2.9: An SPT problem.

We start with leaf consistency. Consider removing agent $4 \in L(\mathcal{T})$. We get $\delta^1\big(((N\backslash\{4\}\cup\{v_0\}), t^{N\backslash\{4\}})\big) = (-10, 9, 11)$, $\delta^2\big(((N\backslash\{4\}\cup\{v_0\}), t^{N\backslash\{4\}})\big) = (2, -6, 14)$ and $\delta^3\big(((N\backslash\{4\} \cup \{v_0\}), t^{N\backslash\{4\}})\big) = (2, 3, 5)$. Therefore, $\delta^1_1\big(((N \cup \{v_0\}, E), t)\big) \neq \delta^1_1\big(((N\backslash\{4\} \cup \{v_0\}), t^{N\backslash\{4\}})\big)$, $\delta^2_2\big(((N \cup \{v_0\}, E), t)\big) \neq \delta^2_2\big(((N\backslash\{4\} \cup \{v_0\}), t^{N\backslash\{4\}})\big)$ and $\delta^3_3\big(((N \cup \{v_0\}, E), t)\big) \neq \delta^3_3\big(((N\backslash\{4\} \cup \{v_0\}), t^{N\backslash\{4\}})\big)$. Thus, δ^i is not leaf consistent.

Now, we consider strong cost monotonicity. Let us change the cost of the edge $\{v_0, 4\}$ from 12 to 11 and assume that all the other edge costs remain the same. The δ^1 allocation for the new SPT problem is $(-12, 9, 11, 11)$, the δ^2 allocation is $(2, -8, 14, 11)$ and the δ^3 allocation is $(2, 3, 3, 11)$ and the δ^4 allocation does not change. Since, in the new SPT problem δ^1_1, δ^2_2 and δ^3_3 increase as an edge cost decreases, δ^i is not strong cost monotone. $\diamond$

We finally study the properties of the γ rule.

Theorem 2.3. *Let* $((N \cup \{v_0\}, E), t)$ *be an SPT problem and let* $\mathcal{T}$ *be a shortest path tree. Then* $\gamma\big(((N\cup\{v_0\}, E), t)\big)$ *is branch efficient with respect to* $\mathcal{T}$*, efficient, symmetric and power monotone.*

Proof: We start with branch efficiency. From Theorem 2.2, for all $i \in N$, δ^i rule is branch efficient with respect to $\mathcal{T}$. Therefore, we have

$$\begin{aligned}\sum_{j \in B_h(\mathcal{T})} \gamma_j\big(((N \cup \{v_0\}, E), t)\big) &= \sum_{j \in B_h(\mathcal{T})} \frac{1}{n} \sum_{i \in N} \delta^i_j\big(((N \cup \{v_0\}, E), t)\big) \\ &= \frac{1}{n} \sum_{i \in N} \sum_{j \in B_h(\mathcal{T})} \delta^i_j\big(((N \cup \{v_0\}, E), t)\big) \\ &= \frac{1}{n} \sum_{i \in N} t(\mathcal{T}^{B_h(\mathcal{T})}) \\ &= t(\mathcal{T}^{B_h(\mathcal{T})})\end{aligned}$$

for all $h \in H(\mathcal{T})$.

Next, we show that the γ rule is symmetric. For symmetric agents $k, l \in N$ we have $\delta^i_k\big(((N \cup \{v_0\}, E), t)\big) = \delta^i_l\big(((N \cup \{v_0\}, E), t)\big)$ for all $i \in N$. Thus, we get

$$\begin{aligned}\gamma_k\big(((N \cup \{v_0\}, E), t)\big) &= \frac{1}{n} \sum_{i \in N} \delta^i_k\big(((N \cup \{v_0\}, E), t)\big) \\ &= \frac{1}{n} \sum_{i \in N} \delta^i_l\big(((N \cup \{v_0\}, E), t)\big) \\ &= \gamma_l\big(((N \cup \{v_0\}, E), t)\big).\end{aligned}$$

Finally, we show that the γ rule is power monotone. Recall the following alternative definition of the γ rule.

$$\gamma_i\big(((N \cup \{v_0\}, E), t)\big) = \frac{t(\mathcal{T})}{n} + \frac{1}{n} \sum_{j \in N} p_{ij} \text{ for all } i \in N.$$

Based on this definition, for any $i, j \in N$ such that $\mathcal{P}_i \leq \mathcal{P}_j$, we have $\gamma_i\big(((N \cup \{v_0\}, E), t)\big) \leq \gamma_j\big(((N \cup \{v_0\}, E), t)\big)$ and therefore this rule is power monotone. □

In the next example, we demonstrate that the γ rule is not leaf consistent and is not strong cost monotone.

Example 2.13. Consider the SPT problem in Figure 2.9. For this problem, we have $\gamma\big(((N \cup \{v_0\}, E), t)\big) = (-\frac{7}{4}, \frac{3}{2}, 8, \frac{45}{4})$.

We start with leaf consistency. Consider removing agent $4 \in L(\mathcal{T})$. We get $\gamma\big(((N \backslash \{4\} \cup \{v_0\}), t^{N \backslash \{4\}})\big) = \frac{1}{3}\left[(-2, 2, 10) + (2, -6, 14) + (2, 3, 5)\right] = (-2, 2, 10)$. Thus, $\gamma_i\big(((N \cup \{v_0\}, E), t)\big) \neq \gamma_i\big(((N \backslash \{4\} \cup \{v_0\}), t^{N \backslash \{4\}})\big)$ for $i \in N \backslash \{4\}$. Therefore,

γ is not leaf consistent.

Now, we consider strong cost monotonicity. Let us change the cost of the edge $\{v_0, 4\}$ from 12 to 11 and assume that all the other edge costs remain the same. The γ allocation for the new SPT problem is $(-\frac{6}{4}, \frac{7}{4}, \frac{33}{4}, \frac{42}{4})$. Since, the cost allocated to agents 1, 2 and 3 increase as an edge cost decreases, γ is not strong cost monotone. $\diamond$

We present a summary of the properties of the allocation rules in the Table 2.1.

	θ	δ^i	γ
Efficiency	✓	✓	✓
Branch Efficiency	✓	✓	✓
Symmetry	✓	✓	✓
Strong Cost Monotonicity	✓	×	×
Power Monotonicity	×	×	✓
Leaf Consistency	✓	×	×

Table 2.1: Properties of the cost allocation rules.

Characterisation of the Tree Solution θ

This section characterises the tree solution θ.

Theorem 2.4. *Let $((N \cup \{v_0\}, E), t)$ be an SPT problem. Then the cost allocation rule θ is the unique rule that satisfies efficiency and leaf consistency.*

Proof: From Theorem 2.1, θ is efficient and leaf consistent. Let ψ be a cost allocation rule that satisfies efficiency and leaf consistency. We show that $\psi\big(((N \cup \{v_0\}, E), t)\big) = \theta\big(((N \cup \{v_0\}, E), t)\big)$ by induction on the number of agents in $((N \cup \{v_0\}, E), t)$. First, let $|N| = 1$. We get $\psi\big(((N \cup \{v_0\}, E), t)\big) = \theta\big(((N \cup \{v_0\}, E), t)\big)$ from efficiency. Assume that if $|N| = n$ then $\psi\big(((N \cup \{v_0\}, E), t)\big) = \theta\big(((N \cup \{v_0\}, E), t)\big)$. Let $|N| = n + 1$. The shortest path tree $\mathcal{T}$ formed by the agents in N, has at least one leaf. Consider removing a leaf $l \in L(\mathcal{T})$. Since ψ is leaf consistent, we have

$$\psi_i\big(((N \cup \{v_0\}, E), t)\big) = \psi_i\big(((N\backslash\{l\} \cup \{v_0\}, E^{N\backslash\{l\}}), t^{N\backslash\{l\}})\big)$$

for all $i \in N\backslash\{l\}$. Then by induction,

$$\psi_i\big(((N\backslash\{l\} \cup \{v_0\}, E^{N\backslash\{l\}}), t^{N\backslash\{l\}})\big) = \theta_i\big(((N\backslash\{l\} \cup \{v_0\}, E^{N\backslash\{l\}}), t^{N\backslash\{l\}})\big)$$

for all $i \in N \backslash \{l\}$. Since θ is leaf consistent, we get

$$\theta_i\big(((N \cup \{v_0\}, E), t)\big) = \theta_i\big(((N \backslash \{l\} \cup \{v_0\}, E^{N \backslash \{l\}}), t^{N \backslash \{l\}})\big)$$

for all $i \in N \backslash \{l\}$. Therefore, we have

$$\psi_i\big(((N \cup \{v_0\}, E), t)\big) = \theta_i\big(((N \cup \{v_0\}, E), t)\big)$$

for all $i \in N \backslash \{l\}$. From efficiency,

$$\psi_l\big(((N \cup \{v_0\}, E), t)\big) = \theta_l\big(((N \cup \{v_0\}, E), t)\big).$$

This completes the proof. □

Chapter 3

Shortest Path Tree Games

We introduce the shortest path tree (SPT) games corresponding to the SPT problems defined in the previous chapter. The set of players of an SPT game is the set of agents in the corresponding SPT problem. The cost of a subset of players (coalition) in an SPT game is the cost of the shortest path tree formed by its members. This chapter studies several properties and solutions of SPT games.

Cooperative game theory is commonly used to model interactive settings where multiple decision makers benefit from cooperation upon solving a joint problem and it provides a natural framework to study cost allocation problems in such settings. The main focus of cooperative game theory is allocating the optimal cost value of this joint problem amongst the decision makers involved. In the cooperative game theory literature, there are various allocation concepts, which either define a one-point solution or define a set of solutions. For the SPT games, we consider the Shapley value (Shapley, 1953), the τ value (Tijs, 1981) and the nucleolus (Schmeidler, 1969), which define one-point solutions, and the core (Gillies, 1953), which defines a set of solutions. In a cooperative game, the players are able to form coalitions whose cost they use as a basis of comparison with the cost they are allocated for the overall game. The core is a fundamental notion in cooperative game theory. It is a convex polyhedron described by an exponential number of inequalities, each of which guarantees that the players in a coalition do not pay more than what they would have paid if they broke away from the rest of the players. Core allocations create no disincentive for cooperation and consequently are considered to be stable. Core allocations do not exist for all cooperative games. However, if they do exist, it is desirable to find allocations that belong to the core.

In this chapter, we study the core of the SPT games since it is not empty. We start by identifying core allocations for the SPT games. We first investigate the stability of the allocation rules defined in Chapter 2 for SPT problems. Although these rules have been defined independent of the game context, they all transpire to be stable for the SPT games. On the other hand, the Shapley value and the τ value are not necessarily core elements. We conclude our work on core allocations by proposing two methods for generating core allocations. The proposed methods are both based on solving a shortest path tree problem in the absence of a special player, and this problem can be solved using Dijkstra's algorithm (Dijkstra, 1959) with a complexity of $O(|N|^2)$ where N is the set of players. Next, we present a polyhedral analysis of the core of the SPT games. We determine the dimension and identify a class of facets of the core of SPT games that correspond to SPT problems with a unique optimal tree, and we present a class of extreme points of the core of the SPT games. We furthermore identify a collection of coalitions for which the corresponding inequalities can be omitted from the description of the core of the SPT games. Finally, we discuss some aspects of the computation of the nucleolus of an SPT game theoretically.

Another area of interest of cooperative game theory is properties that are satisfied by different classes of games. This chapter shows that the SPT games are totally balanced, allow a population monotonic allocation scheme but are not submodular and not monotone.

This chapter is organised as follows. In Sections 3.1 and 3.2, we define the SPT games and study their properties respectively. In Section 3.3, we focus on core allocations for the SPT games. Section 3.4 determines the dimension and identifies a class of facets of the core of SPT games that correspond to SPT problems with a unique optimal tree, and presents a class of extreme points of the core of the SPT games. In Section 3.5, we identify a collection of coalitions for which the corresponding inequalities are redundant in the description of the core of the SPT games. Finally, Section 3.7 reviews some aspects of the computation the nucleolus of the SPT games.

Related Work

In order to enable us to position our work on SPT games within the body of existing literature, we next mention a number of cooperative games dealing with situations where cost is to be allocated amongst cooperating players who are located at or can control different elements of a graph. The minimum cost spanning tree (MCST) games were introduced Bird (1976), and studied in Granot and Huberman (1981) and Granot and Huberman (1984). Similar to SPT games, there is one-to-one correspondence between the vertices, excluding the source vertex, and the players in MCST games. The players form a minimum cost spanning tree where the connection cost of a player is equal to its connection cost to the tree, not to the source vertex. The MCST games have nonempty cores since the allocation where each player is assigned their connection cost to a given minimum cost spanning tree, known as the Bird rule, is always a core element.

A different application of cooperative games to cost allocation in graph theoretical problems is the class of shortest path (SP) games studied in Fragnelli et al. (2000) and Voorneveld and Grahn (2002). The underlying network optimisation problem is that of finding the shortest path between a source and a sink. Fragnelli et al. (2000) assume there are various sources and sinks in the network. Each player is associated with a set of nodes including the sources and the sinks. The players can generate income when they transport a good from a source to the sink in the network, and incur the cost of the transportation given by the length of the path from a source to a sink. They study the allocation of the profits in such settings amongst the players involved and identify the conditions under which SP games are balanced. On the other hand, Voorneveld and Grahn (2002) assume that there is a single source and a single sink in the SP game. Each player in this version of the SP games is associated with a number of arcs rather than nodes. Moreover, a single arc can be associated with more than one player. A player can generate a reward if he can transfer his own good from the source of the network to the sink. The authors show that their version of SP games is totally balanced and allow a population monotonic allocation scheme. Furthermore, they present methods for obtaining core elements.

Finally, we consider congestion network problems and the corresponding games studied by Quant et al. (2006). In congestion network problems, the cost of an arc of the network depends on the number of agents using this arc. For example, if more vehicles use a particular road it would be more "costly" in terms of the congestions effects to use this particular road. It is shown that the congestion network problems with concave cost functions have an optimal solution that forms a tree, however the core of the corresponding games may be empty whereas congestion network games with convex cost functions are balanced. Note that an SPT problem is in fact a symmetric congestion network problem with linear cost functions. For the congestion network games corresponding to such problems, the authors show that assigning to each player the costs of its optimal path yields a unique core element. In this chapter, we restate this result for the SPT games where we show that the tree solution, which is unique, always generates a core element.

Other examples of the interplay between cooperative game theory and cost allocation problems arising from networks can be found in the survey by Borm et al. (2001).

3.1 Definition of the Shortest Path Tree Games

This section introduces the SPT games where the set of players is equal to the set of agents N in an SPT problem $((N \cup \{v_0\}, E), t)$. Recall that $\mathcal{T}^S$ denotes a shortest path tree of the SPT problem $((S \cup \{v_0\}, E^S), t^S)$, and the cost of $\mathcal{T}^S$ is denoted by $t(\mathcal{T}^S)$.

Definition 3.1. *Let* $((N \cup \{v_0\}, E), t)$ *be an SPT problem. Then the corresponding* shortest path tree (SPT) game (N, c) *is defined by*

$$c(S) = t(\mathcal{T}^S) \text{ for all } S \subseteq N.$$

Next, we illustrate the characteristic function c of the SPT game corresponding to the SPT problem from Example 2.1.

Example 3.1. Consider the SPT problem with $N = \{1, 2, 3\}$ in Figure 3.1. For the corresponding SPT game (N, c), we have $c(N) = t(\mathcal{T}) = 2+3+5 = 10$. For $S = \{3\}$, $c(S) = t(\mathcal{T}^S) = 14$ and for $S = \{2, 3\}$, $c(S) = t(\mathcal{T}^S) = 9 + 11 = 20$.

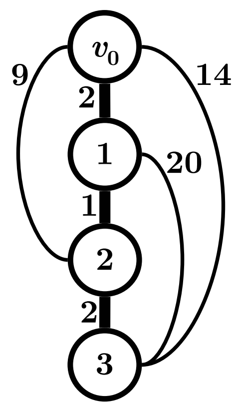


Figure 3.1: An SPT problem.

Table 3.1 gives the costs of all the coalitions of the SPT game (N, c).

S	$\{1\}$	$\{2\}$	$\{3\}$	$\{1,2\}$	$\{1,3\}$	$\{2,3\}$	$\{1,2,3\}$
$c(S)$	2	9	14	5	16	20	10

Table 3.1: Coalitional costs of the SPT game (N, c).

$\diamond$

3.2 Properties of the Shortest Path Tree Games

In this section, we study some properties of the SPT games to provide a more detailed understanding of these games as well as helping us to position the SPT games amongst other cooperative games. We show that the SPT games are totally balanced, they allow a population monotonic allocation scheme but they are not submodular. Finally, we show that SPT games are not monotone.

First, we show that the SPT games are totally balanced, that is, the core of any of the subgames of an SPT game is not empty. Similar to the Bird rule for the MCST games (Bird, 1976), a core allocation can be derived from a shortest path tree $\mathcal{T}$ of an SPT problem. Recall that the tree solution θ is defined as $\theta_i((N \cup \{v_0\}, E), t) = t^{E(i,\mathcal{T})}$ for all $i \in N$.

Theorem 3.1. *Let $((N \cup \{v_0\}, E), t)$ be an SPT problem. Let (N, c) be the corresponding SPT game. Then $\theta((N \cup \{v_0\}, E), t) \in Core(c)$.*

Proof: We know that $\theta((N \cup \{v_0\}, E), t)$ is efficient from Theorem 2.1. Therefore, we get $\sum_{i \in N} \theta_i((N \cup \{v_0\}, E), t) = t(\mathcal{T}) = c(N)$.

For $S \subset N$, we have

$$\sum_{i \in S} \theta_i((N \cup \{v_0\}, E), t) = \sum_{i \in S} t^{E(i,\mathcal{T})} \leq \sum_{i \in S} t^{E(i,\mathcal{T}^S)} = c(S)$$

where the inequality follows from Lemma 2.3. Thus, $\theta((N \cup \{v_0\}, E), t)$ is in $Core(c)$.

$\square$

Now, we show that SPT games are totally balanced.

Theorem 3.2. *SPT games are totally balanced.*

Proof: Let $((N \cup \{v_0\}, E), t)$ be an SPT problem. Let $S \subseteq N$. Let $((S \cup \{v_0\}, E^S), t^S)$ be an SPT problem and let $\mathcal{T}^S$ be a shortest path tree. Let (S, c^S) be the subgame where $c^S(T) = c(T)$ for all $T \subseteq S$. Therefore, every subgame (S, c^S) of an SPT game is also an SPT game and is balanced from Theorem 3.1 since the tree solution θ is always in the core of the SPT games. □

Next, we prove the existence of a population monotonic allocation scheme for SPT games.

Theorem 3.3. *Let $((N \cup \{v_0\}, E), t)$ be an SPT problem. Let (N, c) be the corresponding SPT game. Then the tree solution θ generates a population monotonic allocation scheme for (N, c).*

Proof: We show that both conditions of population monotonic allocation schemes hold for θ.

i. For all $S \in 2^N \backslash \{\emptyset\}$, we have

$$\sum_{i \in S} \theta_i((S \cup \{v_0\}, E^S), t^S) = \sum_{i \in S} t^{E(i, \mathcal{T}^S)} = c(\mathcal{T}^S) = c(S).$$

ii. For all $S, T \in 2^N \backslash \{\emptyset\}$ such that $S \subseteq T$ and for all $i \in S$, we have

$$\theta_i((S \cup \{v_0\}, E^S), t^S) = t^{E(i, \mathcal{T}^S)} \geq t^{(i, \mathcal{T}^T)} = \theta_i((T \cup \{v_0\}, E^T), t^T)$$

where the inequality follows from Lemma 2.3. □

We show that the SPT games are not submodular and not monotone in the following example.

Example 3.2. Consider the SPT problem and the corresponding SPT game displayed in Figure 3.2 (please see next page). Let $S = \{2\}$, $T = \{1, 2\}$ and $i = 3$. We have $S \subseteq T \subseteq N \backslash \{i\}$. We have $c(S \cup \{i\}) - c(S) = c(\{2, 3\}) - c(\{2\}) = 5 - 11 = -6$ and $c(T \cup \{i\}) - c(T) = c(\{1, 2, 3\}) - c(\{1, 2\}) = 4 - 3 = 1$. The SPT game is not submodular since $c(S \cup \{i\}) - c(S) < c(T \cup \{i\}) - c(T)$.

For monotonicity, consider $S = \{2\}$ and $T = \{1, 2\}$. We have $S \subseteq T$ but $c(S) > c(T)$.

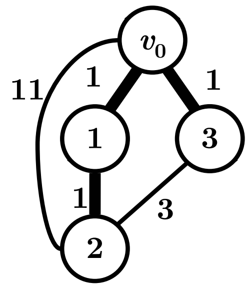


Figure 3.2: An SPT problem.

3.3 Core Allocations for the Shortest Path Tree Games

This section starts with showing that the cost allocation rules that have been introduced for the SPT problems in Section 2.3.1 are elements of the core of the corresponding SPT games. Then, the Shapley and the τ values of the SPT games are considered. It is shown that these values are not necessarily core elements for SPT games. Finally, two methods of generating core elements are presented. Both of these methods rely on finding a shortest path tree in the absence of a special player.

3.3.1 Core Allocations Generated by the Cost Allocation Rules for the Shortest Path Tree Problems

Firstly, we know that the tree solution θ generates a core element of the SPT games from Theorem 3.1.

Consider the SPT problem $((N\backslash\{i\}\cup\{v_0\}, E^{N\backslash\{i\}}), t^{N\backslash\{i\}})$ and recall that we denote an optimal solution to this problem by $\mathcal{T}^{-i}$. Next, we show that the δ^i rule generates a core element.

Theorem 3.4. *Let $((N\cup\{v_0\}, E), t)$ be an SPT problem. Let (N, c) be the corresponding SPT game. Let $i \in N$. Then $\delta^i((N \cup \{v_0\}, E), t) \in \ Core(c)$.*

Proof: We know that the δ^i rule is efficient from Theorem 2.2. Therefore, we get $\sum_{j\in N} \delta^i_j((N \cup \{v_0\}, E), t) = t(\mathcal{T}) = c(N)$. Next, considering the two cases below, we show that $\sum_{j\in S} \delta^i_j((N \cup \{v_0\}, E), t) \leq c(S)$ holds for all $S \subset N$.

Case 1. $i \notin S$. We have

$$\sum_{j\in S} \delta_j^i((N \cup \{v_0\}, E), t) = \sum_{j\in S} t^{E(j,\mathcal{T}^{-i})} \leq \sum_{j\in S} t^{E(j,\mathcal{T}^S)} = c(S)$$

where the inequality follows from Lemma 2.3.

Case 2. $i \in S$. We have

$$\begin{aligned}
\sum_{j\in S} \delta_j^i((N \cup \{v_0\}, E), t) &= \delta_i^i((N \cup \{v_0\}, E), t) + \sum_{j\in S\setminus\{i\}} \delta_j^i((N \cup \{v_0\}, E), t) \\
&= t(\mathcal{T}) - t(\mathcal{T}^{-i}) + \sum_{j\in S\setminus\{i\}} t^{E(j,\mathcal{T}^{-i})} \\
&= \sum_{j\in N} t^{E(j,\mathcal{T})} - \sum_{j\in N\setminus\{i\}} t^{E(j,\mathcal{T}^{-i})} + \sum_{j\in S\setminus\{i\}} t^{E(j,\mathcal{T}^{-i})} \\
&= \sum_{j\in N} t^{E(j,\mathcal{T})} - \sum_{j\in N\setminus S} t^{E(j,\mathcal{T}^{-i})} \\
&= \sum_{j\in S} t^{E(j,\mathcal{T})} + \sum_{j\in N\setminus S} \left[t^{E(j,\mathcal{T})} - t^{E(j,\mathcal{T}^{-i})}\right] \\
&\leq \sum_{j\in S} t^{E(j,\mathcal{T})} \\
&\leq \sum_{j\in S} t^{E(j,\mathcal{T}^S)} \\
&= t(\mathcal{T}^S) = c(S)
\end{aligned}$$

where the inequalities follow from Lemma 2.3. □

Theorem 3.4 implies the following corollary since the γ rule is a convex combination of δ^i for $i \in N$ and $Core(c)$ is a convex set.

Corollary 3.1. *Let $((N \cup \{v_0\}, E), t)$ be an SPT problem. Let (N, c) be the corresponding SPT game. Then $\gamma((N \cup \{v_0\}, E), t) \in Core(c)$.*

3.3.2 The Shapley and the τ Values for the Shortest Path Tree Games

We consider two well-known one point allocation concepts from the cooperative game theory literature, namely the Shapley and the τ values. In the next example, we show that the Shapley value of the SPT games is not necessarily a core element.

Example 3.3. Consider the SPT problem displayed in Figure 3.3 and the corresponding SPT game (N, c). The Shapley value of (N, c) is $\phi(c) = (\frac{1}{3}, -\frac{1}{6}, 6\frac{1}{2}, 3\frac{1}{3})$. Let $S = \{2, 4\}$, we have $c(S) = 3$. We get $\sum_{i \in S} \phi_i(c) = \phi_2(c) + \phi_4(c) = -\frac{1}{6} + 3\frac{1}{3} = 3\frac{1}{6} > 3$. Therefore, $\phi(c) \notin Core(c)$. $\diamond$

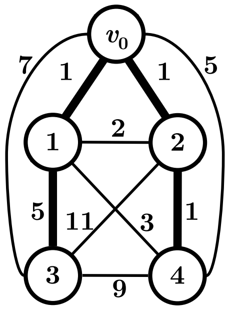


Figure 3.3: An SPT problem.

We show that the τ value is not necessarily in the core of an SPT game in the following example.

Example 3.4. Consider the SPT problem displayed in Figure 3.4 and the corresponding SPT game (N, c). Assume that all the edges that are not shown have cost 100. The τ value of (N, c) is $\tau(c) = (\frac{21}{41}, -\frac{19}{41}, 2\frac{21}{41}, \frac{4}{41}, 4\frac{12}{41}, 6\frac{2}{41})$. Let $S = \{1, 3\}$, we have $c(S) = 3$. We get $\sum_{i \in S} \tau_i(c) = \tau_1(c) + \tau_3(c) = \frac{21}{41} + 2\frac{21}{41} = 3\frac{1}{41} > 3$. Therefore, $\tau(c) \notin Core(c)$. $\diamond$

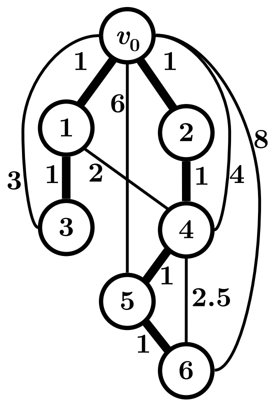


Figure 3.4: An SPT problem.

3.3.3 Two Further Methods of Generating Core Allocations for the Shortest Path Tree Games

We introduce two further methods of generating core allocations. For each method, we first define a cost allocation rule for the SPT problem and then show that this rule generates a core element for the corresponding SPT game. Note that the proposed methods are both based on solving a shortest path tree problem in the absence of a special player, and this problem can be solved using Dijkstra's algorithm (Dijkstra, 1959).

Recall that $H(\mathcal{T}) = \{i : V(i,\mathcal{T}) = \{i\}\}$ denotes the set of hubs of $\mathcal{T}$ where $V(i,\mathcal{T})$ is the set of players on the shortest path of i on $\mathcal{T}$, excluding v_0 but including i. Furthermore, $Hub(i,\mathcal{T})$ denotes the hub of agent i on $\mathcal{T}$ and $Hub(i,\mathcal{T}) = h$ if $i \in B_h(\mathcal{T})$ where $B_h(\mathcal{T})$ is the branch of N induced by hub h of $\mathcal{T}$. Let $i \in N\backslash H(\mathcal{T})$. First, we introduce the allocation rule σ^i. This allocation rule is such that all the agents in $N\backslash\{i, Hub(i,\mathcal{T})\}$ pay their shortest path cost on $\mathcal{T}$, agent i pays its shortest path cost on the tree formed by all the agents excluding its hub and the remaining costs are paid by the hub of agent i on $\mathcal{T}$.

Definition 3.2. *Let $((N \cup \{v_0\}, E), t)$ be an SPT problem and let $\mathcal{T}$ be a shortest path tree. Let $i \in N\backslash H(\mathcal{T})$. Then the σ^i rule is defined as follows*

$$\sigma^i_j((N \cup \{v_0\}, E), t) = \begin{cases} t^{E(j,\mathcal{T})} & \textit{for } j \in N\backslash\{i, Hub(i,\mathcal{T})\}, \\ t^{E(i,\mathcal{T}^{-Hub(i,\mathcal{T})})} & \textit{for } j = i, \\ r^{\sigma}_{Hub(i,\mathcal{T})} & \textit{for } j = Hub(i,\mathcal{T}). \end{cases}$$

where $r^{\sigma}_{Hub(i,\mathcal{T})} = t(\mathcal{T}) - \sum_{k \in N\backslash\{i, Hub(i,\mathcal{T})\}} t^{E(k,\mathcal{T})} - t^{E(i,\mathcal{T}^{-Hub(i,\mathcal{T})})}$.

Example 3.5. Consider the SPT problem displayed in Figure 3.5. Consider the cost allocation $\sigma^3((N \cup \{v_0\}, E), t)$. We have $Hub(3,\mathcal{T}) = 1$. Agents in $N\backslash\{1,3\}$ pay the cost of their shortest path on $\mathcal{T}$. Agent 3 pays the cost of its shortest path on $\mathcal{T}^{-1}$ and the remaining costs are allocated to agent 1. Since $t(\mathcal{T}) = 10$, $t^{E(3,\mathcal{T}^{-1})} = 11$ and $t^{E(2,\mathcal{T})} = 3$, we get $\sigma^3((N \cup \{v_0\}, E), t) = (10-3-11, 3, 11) = (-4, 3, 11)$. Moreover, $\sigma^2((N \cup \{v_0\}, E), t) = (-4, 9, 5)$. ◇

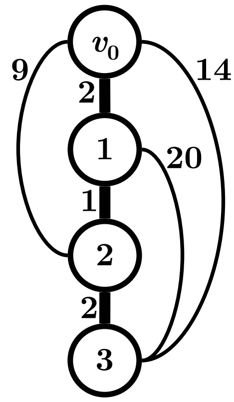


Figure 3.5: An SPT problem.

The following theorem shows that the σ^i rule generates a core allocation for the corresponding SPT game.

Theorem 3.5. *Let $((N \cup \{v_0\}, E), t)$ be an SPT problem and let $\mathcal{T}$ be a shortest path tree. Let (N, c) be the corresponding SPT game. Let $i \in N \backslash H(\mathcal{T})$. Then $\sigma^i((N \cup \{v_0\}, E), t) \in Core(c)$.*

Proof: It is trivial to show that σ^i is efficient. Thus, we have $\sum_{j \in N} \sigma^i_j((N \cup \{v_0\}, E), t) = t(\mathcal{T}) = c(N)$. Now, we show that $\sum_{j \in S} \sigma^i_j((N \cup \{v_0\}, E), t) \leq c(S)$ holds for all $S \subset N$ considering the four cases below.

Case 1. $i, Hub(i, \mathcal{T}) \notin S$.

$$\begin{aligned}\sum_{j \in S} \sigma^i_j((N \cup \{v_0\}, E), t) &= \sum_{j \in S} t^{E(j, \mathcal{T})} \\ &\leq \sum_{j \in S} t^{E(j, \mathcal{T}^S)} \\ &= t(\mathcal{T}^S) = c(S)\end{aligned}$$

where the inequality follows from Lemma 2.3.

Case 2. $i, Hub(i, \mathcal{T}) \in S$.

$$\begin{aligned}
\sum_{j \in S} \sigma_j^i((N \cup \{v_0\}, E), t) = \quad & \sigma_i^i((N \cup \{v_0\}, E), t) \\
& + \sigma_{Hub(i,\mathcal{T})}^i((N \cup \{v_0\}, E), t) \\
& + \sum_{j \in S \setminus \{i, Hub(i,\mathcal{T})\}} \sigma_j^i((N \cup \{v_0\}, E), t) \\
= \quad & t^{E(i, \mathcal{T}^{-Hub(i,\mathcal{T})})} \\
& + t(\mathcal{T}) - \sum_{j \in N \setminus \{i, Hub(i,\mathcal{T})\}} t^{E(j,\mathcal{T})} - t^{E(i, \mathcal{T}^{-Hub(i,\mathcal{T})})} \\
& + \sum_{j \in S \setminus \{i, Hub(i,\mathcal{T})\}} t^{E(j,\mathcal{T})} \\
= & \sum_{j \in N} t^{E(j,\mathcal{T})} - \sum_{j \in N \setminus S} t^{E(j,\mathcal{T})} \\
= & \sum_{j \in S} t^{E(j,\mathcal{T})} \\
\leq & \sum_{j \in S} t^{E(j,\mathcal{T}^S)} \\
= & t(\mathcal{T}^S) = c(S)
\end{aligned}$$

where the inequality follows from Lemma 2.3.

Case 3. $i \in S$, $Hub(i, \mathcal{T}) \notin S$.

$$\begin{aligned}
\sum_{j \in S} \sigma_j^i((N \cup \{v_0\}, E), t) &= \sigma_i^i((N \cup \{v_0\}, E), t) + \sum_{j \in S \setminus \{i\}} \sigma_j^i((N \cup \{v_0\}, E), t) \\
&= t^{E(i, \mathcal{T}^{-Hub(i,\mathcal{T})})} + \sum_{j \in S \setminus \{i\}} t^{E(j,\mathcal{T})} \\
&\leq t^{E(i,\mathcal{T}^S)} + \sum_{j \in S \setminus \{i\}} t^{E(j,\mathcal{T}^S)} \\
&= \sum_{j \in S} t^{E(j,\mathcal{T}^S)} = t(\mathcal{T}^S) = c(S)
\end{aligned}$$

where the inequality follows from Lemma 2.3.

Case 4. $i \notin S$, $Hub(i,\mathcal{T}) \in S$.

$$\begin{aligned}
\sum_{j\in S}\sigma^i_j((N\cup\{v_0\},E),t) = \quad & \sigma^i_{Hub(i,\mathcal{T})}((N\cup\{v_0\},E),t) \\
& + \sum_{j\in S\setminus\{Hub(i,\mathcal{T})\}} \sigma^i_j((N\cup\{v_0\},E),t) \\
= \quad & t(\mathcal{T}) - \sum_{j\in N\setminus\{i,Hub(i,\mathcal{T})\}} t^{E(j,\mathcal{T})} - t^{E(i,\mathcal{T}^{-Hub(i,\mathcal{T})})} \\
& + \sum_{j\in S\setminus\{Hub(i,\mathcal{T})\}} t^{E(j,\mathcal{T})} \\
=& \sum_{j\in N} t^{E(j,\mathcal{T})} - \sum_{j\in N\setminus(S\cup\{i\})} t^{E(j,\mathcal{T})} - t^{E(i,\mathcal{T}^{-Hub(i,\mathcal{T})})} \\
=& \sum_{j\in S} t^{E(j,\mathcal{T})} + t^{E(i,\mathcal{T})} - t^{E(i,\mathcal{T}^{-Hub(i,\mathcal{T})})} \\
\leq& \sum_{j\in S} t^{E(j,\mathcal{T})} \\
\leq& \sum_{j\in S} t^{E(j,\mathcal{T}^S)} \\
=& t(\mathcal{T}^S) = c(S)
\end{aligned}$$

where the inequalities follow from Lemma 2.3. □

Next, we introduce another cost allocation rule for SPT problems. We start with some definitions and notation.

Consider the coalition $F(j,\mathcal{T}) \subseteq N$, that is, the coalition of the followers of a player j on $\mathcal{T}$. Recall that $j \in F(j,\mathcal{T})$. Let $i,j \in N$ and $i \neq j$. We let $Sup(i,F(j,\mathcal{T}))$ to denote the *superior of player i with respect to the followers of player j on $\mathcal{T}$* where

$$Sup(i,F(j,\mathcal{T})) = \begin{cases} j & \text{if } i \in F(j,\mathcal{T}) \\ Hub(i,\mathcal{T}) & \text{if } i \notin F(j,\mathcal{T}). \end{cases}$$

Example 3.6. Let us consider the shortest path tree $\mathcal{T}$ illustrated in Figure 3.6. Consider the coalition $F(5,\mathcal{T}) = \{5,6\}$. We have $Sup(6, F(5,\mathcal{T})) = 5$ since $6 \in F(5,\mathcal{T})$ and $Sup(3, F(5,\mathcal{T})) = 1$ since $3 \notin F(5,\mathcal{T})$. ◇

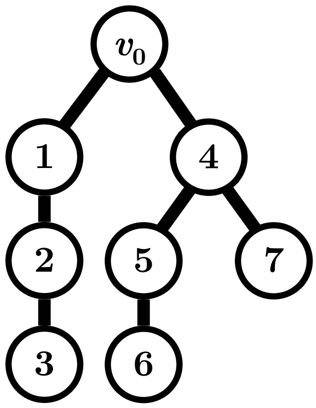


Figure 3.6: A shortest path tree.

Now, we are ready to define a new cost allocation rule for the SPT problems. Let $i, j \in N\backslash H(\mathcal{T})$ and $i \neq j$. The cost allocation rule $\rho^{i,F(j,\mathcal{T})}$ is defined with respect to an agent i and the followers of agent j. The cost allocation rule is such that all the agents except for i and its superior with respect to $F(j,\mathcal{T})$ pay their shortest path cost on $\mathcal{T}$, agent i pays its shortest path cost on the tree formed by all the agents excluding its superior and its superior pays the remaining costs.

Definition 3.3. *Let $((N \cup \{v_0\}, E), t)$ be an SPT problem and let $\mathcal{T}$ be a shortest path tree. Let $i, j \in N\backslash H(\mathcal{T})$ and $i \neq j$. Then the $\rho^{i,F(j,\mathcal{T})}$ rule is defined as*

$$\rho_k^{i,F(j,\mathcal{T})}((N \cup \{v_0\}, E), t) = \begin{cases} t^{E(k,\mathcal{T})} & \text{for } k \in N\backslash\{i, Sup(i, F(j,\mathcal{T}))\}, \\ t^{E(i,\mathcal{T}^{-Sup(i,F(j,\mathcal{T}))})} & \text{for } k = i, \\ r^{\rho}_{Sup(i,F(j,\mathcal{T}))} & \text{for } k = Sup(i, F(j,\mathcal{T})). \end{cases}$$

where $r^{\rho}_{Sup(i,F(j,\mathcal{T}))} = t(\mathcal{T}) - \sum_{l \in N\backslash\{i, Sup(i,F(j,\mathcal{T}))\}} t^{E(l,\mathcal{T})} - t^{E(i,\mathcal{T}^{-Sup(i,F(j,\mathcal{T}))})}.$

Example 3.7. Consider the SPT problem displayed in Figure 3.5. Consider the cost allocation $\rho^{3,F(2,\mathcal{T})}((N \cup \{v_0\}, E), t)$. We have $Sup(3, F(2,\mathcal{T})) = 2$. Agents in $N\backslash\{2,3\}$ pay the cost of their shortest path on $\mathcal{T}$. Agent 3 pays the cost of its shortest path on

$\mathcal{T}^{-2}$ and the remaining costs are allocated to agent 2. Since $t(\mathcal{T}) = 10$, $t^{E(3,\mathcal{T}^{-2})} = 14$ and $t^{E(1,\mathcal{T})} = 2$, we get $\rho^{3,F(2,\mathcal{T})}((N \cup \{v_0\}, E), t) = (2, 10-2-14, 14) = (2, -6, 14)$. Moreover, $\rho^{2,F(3,\mathcal{T})}((N \cup \{v_0\}, E), t) = (-4, 9, 5)$. ◇

Observe that $\rho^{i,F(j,\mathcal{T})}((N \cup \{v_0\}, E), t) = \sigma^i((N \cup \{v_0\}, E), t)$ if $i \notin F(j, \mathcal{T})$. Finally, we show that the $\rho^{i,F(j,\mathcal{T})}$ rule generates a core allocation for the corresponding SPT game.

Theorem 3.6. *Let $((N \cup \{v_0\}, E), t)$ be an SPT problem and let $\mathcal{T}$ be a shortest path tree. Let (N, c) be the corresponding SPT game. Let $i, j \in N \backslash H(\mathcal{T})$ and $i \neq j$. Then $\rho^{i,F(j,\mathcal{T})}((N \cup \{v_0\}, E), t) \in Core(c)$.*

Proof: It is trivial to show that $\rho^{i,F(j,\mathcal{T})}$ is efficient. Thus, $\sum_{k \in N} \rho_k^{i,F(j,\mathcal{T})}((N \cup \{v_0\}, E), t) = t(\mathcal{T}) = c(N)$. Now, we show that $\sum_{k \in S} \rho_k^{i,F(j,\mathcal{T})}((N \cup \{v_0\}, E), t) \leq c(S)$ holds for all $S \subset N$ considering the four cases below.

Case 1. $i, Sup(i, F(j, \mathcal{T})) \notin S$.

$$\begin{aligned} \sum_{k \in S} \rho_k^{i,F(j,\mathcal{T})}((N \cup \{v_0\}, E), t) &= \sum_{k \in S} t^{E(k,\mathcal{T})} \\ &\leq \sum_{k \in S} t^{E(k,\mathcal{T}^S)} \\ &= t(\mathcal{T}^S) = c(S) \end{aligned}$$

where the inequality follows from Lemma 2.3.

Case 2. $i, Sup(i, F(j, \mathcal{T})) \in S$.

$$\begin{aligned} \sum_{k \in S} \rho_k^{i,F(j,\mathcal{T})}((N \cup \{v_0\}, E), t) = \quad & \rho_i^{i,F(j,\mathcal{T})}((N \cup \{v_0\}, E), t) \\ & + \rho_{Sup(i,F(j,\mathcal{T}))}^{i,F(j,\mathcal{T})}((N \cup \{v_0\}, E), t) \\ & + \sum_{k \in S \backslash \{i, Sup(i,F(j,\mathcal{T}))\}} \rho_k^{i,F(j,\mathcal{T})}((N \cup \{v_0\}, E), t) \\ = \quad & t^{E(i,\mathcal{T}^{-Sup(i,F(j,\mathcal{T}))})} \\ & + t(\mathcal{T}) - \sum_{k \in N \backslash \{i, Sup(i,F(j,\mathcal{T}))\}} t^{E(k,\mathcal{T})} \\ & - t^{E(i,\mathcal{T}^{-Sup(i,F(j,\mathcal{T}))})} \end{aligned}$$

$$
\begin{aligned}
&+ \sum_{k \in S \setminus \{i, Sup(i, F(j, \mathcal{T}))\}} t^{E(k,\mathcal{T})} \\
&= \sum_{k \in N} t^{E(k,\mathcal{T})} - \sum_{k \in N \setminus S} t^{E(k,\mathcal{T})} \\
&= \sum_{k \in S} t^{E(k,\mathcal{T})} \\
&\leq \sum_{k \in S} t^{E(k,\mathcal{T}^S)} \\
&= t(\mathcal{T}^S) = c(S)
\end{aligned}
$$

where the inequality follows from Lemma 2.3.

Case 3. $i \in S$, $Sup(i, F(j, \mathcal{T})) \notin S$.

$$
\begin{aligned}
\sum_{k \in S} \rho_k^{i,F(j,\mathcal{T})}((N \cup \{v_0\}, E), t) = \quad & \rho_i^{i,F(j,\mathcal{T})}((N \cup \{v_0\}, E), t) \\
& + \sum_{k \in S \setminus \{i\}} \rho_k^{i,F(j,\mathcal{T})}((N \cup \{v_0\}, E), t)) \\
&= t^{E(i,\mathcal{T}^{-Sup(i,F(j,\mathcal{T}))})} + \sum_{k \in S \setminus \{i\}} t^{E(k,\mathcal{T})} \\
&\leq t^{E(i,\mathcal{T}^S)} + \sum_{k \in S \setminus \{i\}} t^{E(k,\mathcal{T}^S)} \\
&= \sum_{k \in S} t^{E(k,\mathcal{T}^S)} \\
&= t(\mathcal{T}^S) = c(S)
\end{aligned}
$$

where the inequality follows from Lemma 2.3.

Case 4. $i \notin S$, $Sup(i, F(j, \mathcal{T})) \in S$.

$$
\begin{aligned}
\sum_{k \in S} \rho_k^{i,F(j,\mathcal{T})}((N \cup \{v_0\}, E), t) = \quad & \rho_{Sup(i,F(j,\mathcal{T}))}^{i,F(j,\mathcal{T})}((N \cup \{v_0\}, E), t) \\
& + \sum_{k \in S \setminus \{Sup(i,F(j,\mathcal{T}))\}} \rho_k^{i,F(j,\mathcal{T})}((N \cup \{v_0\}, E), t) \\
= \quad & t(\mathcal{T}) - \sum_{k \in N \setminus \{i, Sup(i,F(j,\mathcal{T}))\}} t^{E(k,\mathcal{T})} \\
& - t^{E(i,\mathcal{T}^{-Sup(i,F(j,\mathcal{T}))})}
\end{aligned}
$$

$$
\begin{aligned}
&+ \sum_{k\in S\setminus\{Sup(i,F(j,\mathcal{T}))\}} t^{E(k,\mathcal{T})} \\
&= \sum_{k\in N} t^{E(k,\mathcal{T})} - \sum_{k\in N\setminus(S\cup\{i\})} t^{E(k,\mathcal{T})} \\
&- t^{E(i,\mathcal{T}^{-Sup(i,F(j,\mathcal{T}))})} \\
&= \sum_{k\in S} t^{E(k,\mathcal{T})} + t^{E(i,\mathcal{T})} - t^{E(i,\mathcal{T}^{-Sup(i,F(j,\mathcal{T}))})} \\
&\leq \sum_{k\in S} t^{E(k,\mathcal{T})} \\
&\leq \sum_{k\in S} t^{E(k,\mathcal{T}^S)} \\
&= t(\mathcal{T}^S) = c(S)
\end{aligned}
$$

where the inequalities follow from Lemma 2.3. □

3.4 Extreme Points, Dimension and Facets of the Core of the Shortest Path Tree Games

In this section, we analyse the core of the SPT games. We do so firstly by presenting a class of extreme points and secondly by determining the dimension of the core of the SPT games. Finally, we identify a class of facets of the core of the SPT games that correspond to SPT problems with a unique shortest path tree.

3.4.1 Preliminaries on Polyhedral Theory

We present some definitions and results from polyhedral theory that will be used in the subsequent three sections on the extreme points, dimension and facets of the core of the shortest path tree games, respectively.

The following preliminaries are adapted from the books of Nemhauser and Wolsey (1988) and Wolsey (1998). A *polyhedron* $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ is a subset of $\mathbb{R}^n$ described by a finite set of linear constraints. Let $\alpha \in \mathbb{R}^n$ and $\alpha_0 \in \mathbb{R}$. The inequality $\alpha x \leq \alpha_0$ is a *valid inequality* for $P \subseteq \mathbb{R}^n$ if $\alpha x \leq \alpha_0$ for all $x \in P$. The points $x^1, x^2, \ldots, x^k \in \mathbb{R}^n$ are *linearly independent* if the unique solution of $\sum_{i=1}^{k} \lambda_i x^i = 0$ is $\lambda_i = 0$ for all $i = 1, 2, \ldots, k$. The points $x^1, x^2, \ldots, x^k \in \mathbb{R}^n$ are *affinely independent* if $x^2 - x^1, x^3 - x^1, \ldots, x^k - x^1$ are linearly independent. The dimension of P is denoted by $dim(P)$. If the maximum number of affinely independent points in P is $k + 1$ then $dim(P) = k$. Therefore, if there exist $k + 1$ affinely independent points in P then $dim(P) \geq k$. A set of linear equalities are linearly independent if none of the equations can be derived from other equations algebraically. If the maximum number of linearly independent equalities that hold for $P \subseteq \mathbb{R}^n$ is k then $dim(P) = n - k$. Hence, if there exist k linearly independent equalities that hold for $P \subseteq \mathbb{R}^n$ then $dim(P) \leq n - k$. A *face* of P is $F = \{x \in P : \alpha x = \alpha_0\}$ if $\alpha x \leq \alpha_0$ is a valid inequality for P. A *facet* of P is a face F of P if $dim(F) = dim(P) - 1$. An *extreme point* of P is a face F of P if $dim(F) = 0$.

We have the following example, which is adapted from Wolsey (1998), to illustrate the above concepts[1].

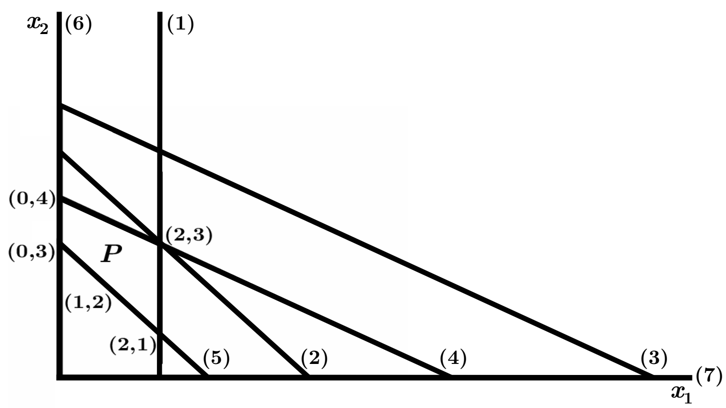


Figure 3.7: A polyhedron P.

Example 3.8. Consider the polyhedron $P \subseteq \mathbb{R}^2$ displayed in Figure 3.7, which is defined by the following inequalities:

$$
\begin{array}{lr}
x_1 \qquad\quad \leq 2 & (1) \\
x_1 + \ \ x_2 \leq 5 & (2) \\
x_1 + 2x_2 \leq 12 & (3) \\
x_1 + 2x_2 \leq 8 & (4)
\end{array}
$$

[1]For illustrations of further MP related concepts see Gebreiter (2011).

$$\begin{array}{rcll} x_1 + & x_2 & \geq 3 & (5) \\ x_1 & & \geq 0 & (6) \\ & x_2 & \geq 0 & (7). \end{array}$$

Firstly, since $P \subseteq \mathbb{R}^2$, $dim(P) \leq 2$. Secondly, since $(2,1)$, $(1,2)$ and $(2,3)$ are three affinely independent points in P, $dim(P) \geq 2$. Therefore, $dim(P) = 2$. The inequality $x_1 \leq 2$ defines a facet of P since $(2,1)$ and $(2,3)$ are two affinely independent points in P satisfying $x_1 \leq 2$ as an equality. Similarly, the inequalities $x_1 + 2x_2 \leq 8$, $x_1 \geq 0$ and $x_1 + x_2 \geq 3$ define facets of P. The inequality $x_1 + x_2 \leq 5$ defines a face consisting only of point $(2,3)$ of P. The inequalities $x_1 + x_2 \leq 5$, $x_1 + 2x_2 \leq 12$ and $x_2 \geq 0$ are redundant in the description of P. The minimal description of P is given by:

$$\begin{array}{rcl} x_1 & & \leq 2 \\ x_1 + & 2x_2 & \leq 8 \\ x_1 + & x_2 & \geq 3 \\ x_1 & & \geq 0. \end{array}$$

Finally, the points $(0,3)$, $(0,4)$, $(2,1)$ and $(2,3)$ are the extreme points of P. $\diamond$

3.4.2 Extreme Points of the Core

This section presents a class of extreme points of the core of the SPT games. This class of extreme points has been initially identified during our simulations on PORTA[2], a free software consisting of routines for analysing polytopes and polyhedra. Firstly, note that the class of extreme points presented in this section does not constitute all the extreme points of the core of the SPT games. Furthermore, note that there did not appear any other class of points that were always extreme points of the core of the SPT games in our extensive simulations.

We start by identifying a subset of permutations of N that define marginal vectors that correspond to allocations $\delta^i((N \cup \{v_0\}, E), t)$ for $i \in N$. Let $\Pi(N)$ denote the set

[2]For further information on PORTA and to download the software please visit http://www2.iwr.uni-heidelberg.de/groups/comopt/software/PORTA/

of all permutations of N. Let π denote a permutation of N, and let $\pi(j)$ denote the order of player j in permutation π.

Let $i \in N$. We consider the permutations $\pi \in \Pi(N)$ where

i. $\pi(i) = |N|$, that is, player i is the last player to enter,

ii. $\pi(j) > \pi(k)$ for all $k \in V(j, \mathcal{T}^{-i})\backslash\{j\}$, that is, all the players on the shortest path of a player $j \in N\backslash\{i\}$, excluding j itself, on $\mathcal{T}^{-i}$ enter before player j.

The next lemma shows that the marginal vectors corresponding to such permutations π give the allocation $\delta^i((N \cup \{v_0\}, E), t)$.

Lemma 3.1. *Let* $((N \cup \{v_0\}, E), t)$ *be an SPT problem and* (N, c) *be the corresponding SPT game. Let* $i \in N$. *Let* $\pi \in \Pi(N)$ *denote a permutation of the players* N *such that* $\pi(i) = |N|$ *and* $\pi(j) > \pi(k)$ *for all* $j \in N\backslash\{i\}$ *and for all* $k \in V(j, \mathcal{T}^{-i})\backslash\{j\}$*, and* $m^\pi(c)$ *be the corresponding marginal vector. Then*

$$m^\pi(c) = \delta^i((N \cup \{v_0\}, E), t).$$

Proof: Let $i \in N$. We have

$$m_i^\pi(c) = c(N) - c(N\backslash\{i\}) = t(\mathcal{T}) - t(\mathcal{T}^{-i}) = \delta_i^i((N \cup \{v_0\}, E), t).$$

Let $j \in N\backslash\{i\}$ and let $\pi^j = \{l : \pi(j) > \pi(l)\}$ denote the set of players preceding player j in permutation π. Since $\pi(j) > \pi(k)$ for all $k \in V(j, \mathcal{T}^{-i})\backslash\{j\}$, all the vertices on the shortest path of j on $\mathcal{T}^{-i}$, including j, are in $\pi^j \cup \{j\}$. Thus,

$$t^{E(j, \mathcal{T}^{\pi^j \cup \{j\}})} = t^{E(j, \mathcal{T}^{-i})} \text{ for all } j \in N\backslash\{i\}. \tag{3.1}$$

Furthermore, we have

$$\pi^l \cup \{l\} \subseteq \pi^j \subset \pi^j \cup \{j\} \text{ for } l \in \pi^j. \tag{3.2}$$

We get

$$t^{E(l, \mathcal{T}^{\pi^l \cup \{l\}})} = t^{E(l, \mathcal{T}^{\pi^j})} = t^{E(l, \mathcal{T}^{\pi^j \cup \{j\}})} = t^{E(l, \mathcal{T}^{-i})} \text{ for all } l \in \pi^j \tag{3.3}$$

from (3.2) and since all the vertices on the shortest path of l on $\mathcal{T}^{-i}$ are in $\pi^l \cup \{l\}$.

Therefore, we have

$$\begin{aligned}
m_j^{\pi}(c) &= c(\pi^j \cup \{j\}) - c(\pi^j) \\
&= \sum_{l \in \pi^j \cup \{j\}} t^{E(l,\mathcal{T}^{\pi^j \cup \{j\}})} - \sum_{l \in \pi^j} t^{E(l,\mathcal{T}^{\pi^j})} \\
&= t^{E(j,\mathcal{T}^{\pi^j \cup \{j\}})} + \sum_{l \in \pi^j} \left[t^{E(l,\mathcal{T}^{\pi^j \cup \{j\}})} - t^{E(l,\mathcal{T}^{\pi^j})} \right] \\
&= t^{E(j,\mathcal{T}^{-i})} + \sum_{l \in \pi^j} \left[t^{E(l,\mathcal{T}^{-i})} - t^{E(l,\mathcal{T}^{-i})} \right] \\
&= t^{E(j,\mathcal{T}^{-i})} \\
&= \delta_j^i((N \cup \{v_0\}, E), t)
\end{aligned}$$

for $j \in N \backslash \{i\}$ where the fourth equality follows from (3.1) and (3.3). □

The marginal vectors are either outside the core or are extreme points of the core (Meinhardt, 2002). From Theorem 3.4, the allocations $\delta^i((N \cup \{v_0\}, E), t)$ for $i \in N$ are in the core of the SPT games. Therefore, we have the following result.

Theorem 3.7. *Let* $((N \cup \{v_0\}, E), t)$ *be an SPT problem. Let* (N, c) *be the corresponding SPT game. Let* $i \in N$*. Then* $\delta^i((N \cup \{v_0\}, E), t)$ *is an extreme point of* $Core(c)$*.*

The next proposition states that the tree solution always generates an extreme point of the core of the SPT games.

Proposition 3.1. *Let* $((N \cup \{v_0\}, E), t)$ *be an SPT problem. Let* (N, c) *be the corresponding SPT game. Then* $\theta((N \cup \{v_0\}, E), t)$ *is an extreme point of* $Core(c)$*.*

Proof: Let $\mathcal{T}$ be a shortest path tree. Let $L(\mathcal{T})$ denote the set of leaves of $\mathcal{T}$. We have $L(\mathcal{T}) \neq \emptyset$. Let $l \in L(\mathcal{T})$. From Proposition 2.1, we have $\theta((N \cup \{v_0\}, E), t) = \delta^l((N \cup \{v_0\}, E), t)$. Thus, $\theta((N \cup \{v_0\}, E), t)$ is an extreme point of the core of the SPT games from Theorem 3.7. □

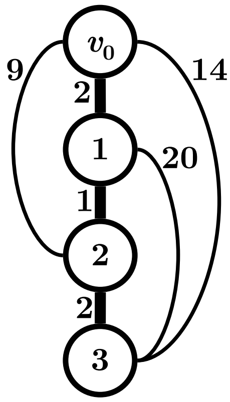


Figure 3.8: An SPT problem.

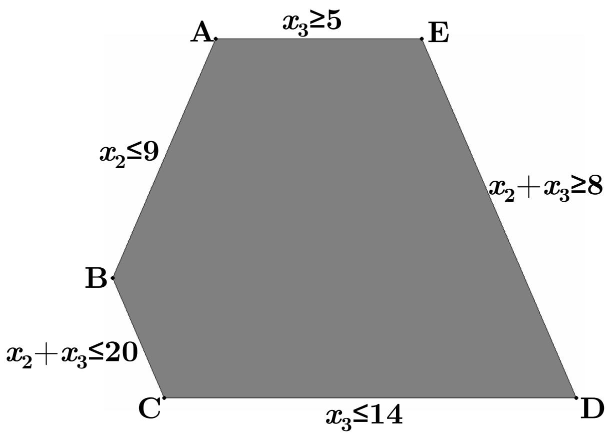


Figure 3.9: The core of the SPT game in Figure 3.8.

Example 3.9. Consider the SPT game displayed in Figure 3.8 and its core as illustrated in Figure 3.9. The extreme points corresponding to the $\delta^i((N \cup \{v_0\}, E), t)$ allocations are $B = \delta^1((N \cup \{v_0\}, E), t) = (-10, 9, 11)$, $D = \delta^2((N \cup \{v_0\}, E), t) = (2, -6, 14)$ and $E = \delta^3((N \cup \{v_0\}, E), t) = \theta((N \cup \{v_0\}, E), t) = (2, 3, 5)$. $\diamond$

3.4.3 Dimension of the Core

This section presents the results on the dimension of the core of SPT games.

We start with the following lemma, which states that for any allocation in the core of the SPT games the total cost allocated to the players in $B_h(\mathcal{T})$ equals the cost of the coalition formed by this branch.

Lemma 3.2. *Let* $((N \cup \{v_0\}, E), t)$ *be an SPT problem and let* $\mathcal{T}$ *be a shortest path tree. Let* (N, c) *be the corresponding SPT game. Let* $x \in Core(c)$*. Then*

$$\sum_{i \in B_h(\mathcal{T})} x_i = c(B_h(\mathcal{T})) \text{ for all } h \in H(\mathcal{T}).$$

Proof: We have $\sum_{i \in N} x_i = c(N)$ since x is a core allocation. The branches of N with respect to $\mathcal{T}$ form a partition, therefore

$$\sum_{h \in H(\mathcal{T})} \sum_{i \in B_h(\mathcal{T})} x_i = c(N). \tag{3.4}$$

Furthermore,

$$c(N) = t(\mathcal{T}) = \sum_{h \in H(\mathcal{T})} t(\mathcal{T}^{B_h(\mathcal{T})}) = \sum_{h \in H(\mathcal{T})} c(B_h(\mathcal{T})) \tag{3.5}$$

where the second equality follows from Lemma 2.2. From (3.4) and (3.5), we get

$$\sum_{h \in H(\mathcal{T})} \sum_{i \in B_h(\mathcal{T})} x_i = \sum_{h \in H(\mathcal{T})} c(B_h(\mathcal{T})). \tag{3.6}$$

Since $B_h(\mathcal{T}) \subseteq N$ for all $h \in H(\mathcal{T})$, we have

$$\sum_{i \in B_h(\mathcal{T})} x_i \leq c(B_h(\mathcal{T})) \text{ for all } h \in H(\mathcal{T}). \tag{3.7}$$

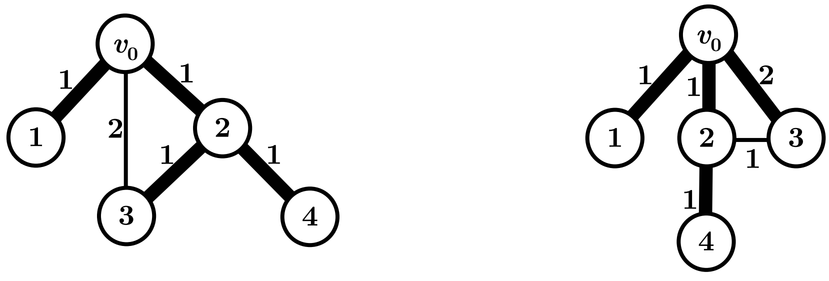
From (3.6) and (3.7), we get

$$\sum_{i \in B_h(\mathcal{T})} x_i = c(B_h(\mathcal{T})) \text{ for all } h \in H(\mathcal{T}).$$

□

Next, we have the following definitions and notation. A shortest path tree is called a *maximal shortest path tree* if it is an optimal solution to $((N \cup \{v_0\}, E), t)$ and it has the maximum number of branches among all solutions to $((N \cup \{v_0\}, E), t)$. Let $MaxTree(N)$ denote the set of maximal shortest path trees of $((N \cup \{v_0\}, E), t)$. Observe that if an agent i is a hub for any shortest path tree $\mathcal{T}$, then it is a hub for all

$\mathcal{T} \in MaxTree(N)$. Therefore, the set of hubs is the same for all maximal shortest path trees. Let H denote the set of hubs $H(\mathcal{T})$ for $\mathcal{T} \in MaxTree(N)$. Note that if the shortest path tree problem $((N \cup \{v_0\}, E), t)$ has a unique optimal solution $\mathcal{T}$ then $MaxTree(N) = \{\mathcal{T}\}$, and $H = H(\mathcal{T})$.



(a) Tree $\mathcal{T}$

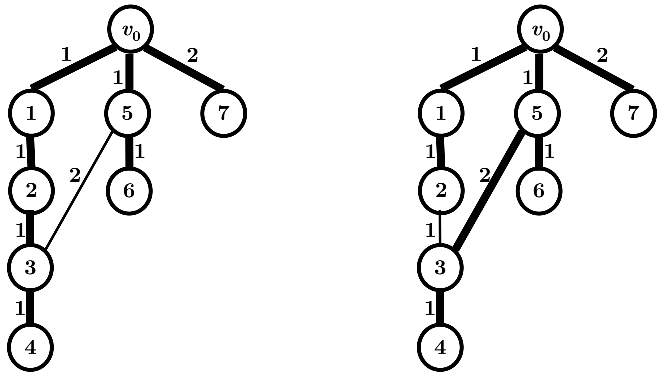
(b) Tree $\widehat{\mathcal{T}}$

Figure 3.10: An example illustrating maximal shortest path trees.

Example 3.10. Consider the SPT problem displayed in Figure 3.10. Assume that all the edges that are not shown have cost 100. The shortest path trees $\mathcal{T}$ and $\widehat{\mathcal{T}}$ are both optimal solutions to this SPT problem. For this example, $\mathcal{T} \notin MaxTree(N)$ and $\widehat{\mathcal{T}} \in MaxTree(N)$ since $\widehat{\mathcal{T}}$ has maximum possible number of branches. Furthermore, $H = \{1, 2, 3\}$. $\diamond$

An agent is called a *multihub agent* if there exists $\mathcal{T}, \widehat{\mathcal{T}} \in MaxTree(N)$ such that $Hub(k, \mathcal{T}) \neq Hub(k, \widehat{\mathcal{T}})$ for all $k \in F(i, \mathcal{T})$ and $Hub(j, \mathcal{T}) = Hub(j, \widehat{\mathcal{T}})$ for all $N \backslash F(i, \mathcal{T})$. Therefore, an agent is a multihub agent if there exists two maximal shortest path trees such that all the followers of agent i on one of those trees, which includes itself, have a different hub on the other one and the rest of the agents have the same hub on both trees. Let $MultiHub$ denote the set of multihub agents. Note that if the shortest path tree problem $((N \cup \{v_0\}, E), t)$ has a unique optimal solution $\mathcal{T}$ then $Multihub = \emptyset$.

Example 3.11. Consider the SPT problem displayed in Figure 3.11. Assume that all the edges that are not shown have cost 100. The shortest path trees $\mathcal{T}$ and $\widehat{\mathcal{T}}$ are both optimal solutions to this SPT problem. Furthermore, $\mathcal{T}, \widehat{\mathcal{T}} \in MaxTree(N)$. We have $Multihub = \{3\}$. Agent 3 and all of its followers on tree $\mathcal{T}$ belong to the branch $B_1(\mathcal{T})$. For tree $\widehat{\mathcal{T}}$, agent 3 and all of its followers on this tree are in branch $B_5(\widehat{\mathcal{T}})$. $\diamond$



(a) Tree $\mathcal{T}$ (b) Tree $\widehat{\mathcal{T}}$

Figure 3.11: An example illustrating multihub agents.

The main result of this section is the following theorem. The theorem follows from Lemma 3.4 and Lemma 3.5 which are stated and proven subsequently.

Theorem 3.8. *Let* $((N \cup \{v_0\}, E), t$ *be an SPT situation. Let* $\mathcal{T} \in MaxTree(N)$ *be a maximal shortest path tree. Let* (N, c) *be the corresponding SPT game. Then*

$$dim(Core(c)) = |N| - |H| - |MultiHub|.$$

Example 3.12. Consider the SPT problem displayed in Figure 3.11. Assume that all the edges that are not shown have cost 100. We have $\mathcal{T}, \widehat{\mathcal{T}} \in MaxTree(N)$. Furthermore, $H = \{1, 5, 7\}$ and $Multihub = \{3\}$. Therefore, $dim(Core(c)) = |N| - |H| - |Multihub| = 7 - 3 - 1 = 3$. From Lemma 3.2, we have $x_1 + x_2 + x_3 + x_4 = 1 + 2 + 3 + 4 = 10$ and $x_5 + x_6 = 1 + 2 = 3$. Secondly, we have $B_1(\widehat{\mathcal{T}}) = \{1, 2\}$ and $B_5(\widehat{\mathcal{T}}) = \{5, 6, 3, 4\}$. In this case, we get $x_1 + x_2 = 1 + 2 = 3$ and $x_5 + x_6 + x_3 + x_4 = 1 + 2 + 3 + 4 = 10$. Therefore, in the core of this SPT game at least four equations hold $x_1 + x_2 = 1 + 2 = 3$, $x_3 + x_4 = 3 + 4 = 7$, $x_5 + x_6 = 1 + 2 = 3$ and $x_7 = 2$ where $x_3 + x_4 = 3 + 4 = 7$ is implied from $x_1 + x_2 + x_3 + x_4 = 1 + 2 + 3 + 4 = 10$ and $x_1 + x_2 = 1 + 2 = 3$. Below will show that these are the only equalities that hold for the core of the SPT games. $\diamond$

As a special instance of this main result, we also state the result for SPT games that correspond to SPT problems that have a unique optimal solution. Let $\mathcal{USPT}(N)$ denote the collection of SPT problems that have a unique shortest path tree. Since there are no multihub agents for a unique shortest path tree, we have the following corollary.

Corollary 3.2. *Let* $((N \cup \{v_0\}, E), t) \in \mathcal{USPT}(N)$. *Let* $\mathcal{T}$ *be the shortest path tree. Let* (N, c) *be the corresponding SPT game. Then*

$$dim(Core(c)) = |N| - |H(\mathcal{T})|.$$

Example 3.13. Consider the SPT game displayed in Figure 3.8. The core of this game is illustrated in Figure 3.9. We have $H(\mathcal{T}) = \{1\}$, thus $dim(Core(c)) = |N| - |H(\mathcal{T})| = 3 - 1 = 2$. $\diamond$

We have the following lemma stating that for an allocation in the core of the SPT games corresponding to SPT situations, the sum of the costs allocated to the followers of a multihub agent is equal to the sum of their shortest path costs, which, as stated previously, is the same for all shortest path trees of any given SPT situation.

Lemma 3.3. *Let* $((N \cup \{v_0\}, E), t)$ *be an SPT situation. Let* $\mathcal{T} \in MaxTree(N)$ *be a maximal shortest path tree. Let* (N, c) *be the corresponding SPT game. Let* $i \in MultiHub$. *Let* $x \in Core(c)$. *Then*

$$\sum_{j \in F(i,\mathcal{T})} x_j = \sum_{j \in F(i,\mathcal{T})} t^{E(j,\mathcal{T})}.$$

Proof: Let $i \in MultiHub$. Let $\mathcal{T}, \widehat{\mathcal{T}} \in MaxTree(N)$ such that $Hub(k, \mathcal{T}) \neq Hub(k, \widehat{\mathcal{T}})$ for all $k \in F(i, \mathcal{T})$ and $Hub(j, \mathcal{T}) = Hub(j, \widehat{\mathcal{T}})$ for all $N \backslash F(i, \mathcal{T})$. Therefore, only the branch of the followers of i on $\mathcal{T}$ changes in $\widehat{\mathcal{T}}$. On $\widehat{\mathcal{T}}$, the followers of i are in the branch $B_{Hub(i,\widehat{\mathcal{T}})}(\widehat{\mathcal{T}})$. On $\mathcal{T}$, this branch consists of the players in $B_{Hub(i,\widehat{\mathcal{T}})}(\mathcal{T})$. Therefore,

$$B_{Hub(i,\widehat{\mathcal{T}})}(\widehat{\mathcal{T}}) = B_{Hub(i,\widehat{\mathcal{T}})}(\mathcal{T}) \cup F(i, \mathcal{T}).$$

Thus,

$$B_{Hub(i,\widehat{\mathcal{T}})}(\widehat{\mathcal{T}}) \backslash B_{Hub(i,\widehat{\mathcal{T}})}(\mathcal{T}) = F(i, \mathcal{T}). \tag{3.8}$$

We have $t^{E(j,\mathcal{T})} = t^{E(j,\widehat{\mathcal{T}})}$. Using this and Lemma 3.2, we get

$$\sum_{j \in B_{Hub(i,\widehat{\mathcal{T}})}(\widehat{\mathcal{T}})} x_j = \sum_{j \in B_{Hub(i,\widehat{\mathcal{T}})}(\widehat{\mathcal{T}})} t^{E(j,\mathcal{T})} \tag{3.9}$$

and

$$\sum_{j \in B_{Hub(i,\widehat{\mathcal{T}})}(\mathcal{T})} x_j = \sum_{j \in B_{Hub(i,\widehat{\mathcal{T}})}(\mathcal{T})} t^{E(j,\mathcal{T})}. \tag{3.10}$$

Subtracting (3.10) from (3.9), and using (3.8), we have

$$\sum_{j \in F(i,\mathcal{T})} x_j = \sum_{j \in F(i,\mathcal{T})} t^{E(j,\mathcal{T})}. \qquad \square$$

Example 3.14. Consider the SPT problem displayed in Figure 3.11. Assume that all the edges that are not shown have cost 100. For tree $\mathcal{T}$, we have $B_1(\mathcal{T}) = \{1, 2, 3, 4\}$ and from Lemma 3.2, we get $x_1 + x_2 + x_3 + x_4 = 1 + 2 + 3 + 4 = 10$. We have $Multihub = \{3\}$. Furthermore, for tree $\widehat{\mathcal{T}}$, we have $B_1(\widehat{\mathcal{T}}) = \{1, 2\}$ and $x_1 + x_2 = 1 + 2 = 3$. These two equations imply $x_3 + x_4 = 3 + 4 = 7$. $\diamond$

We have the following lemma giving us an upper bound on the dimension of the core of the SPT games corresponding to SPT situations with multiple shortest path trees.

Lemma 3.4. *Let $((N \cup \{v_0\}, E), t)$ be an SPT situation. Let $\mathcal{T} \in MaxTree(N)$ be a maximal shortest path tree. Let (N, c) be the corresponding SPT game. Then*

$$dim(Core(c)) \leq |N| - |H| - |MultiHub|.$$

Proof: Let $\mathcal{T} \in MaxTree(N)$. From Lemma 3.2, $\sum_{i \in B_h(\mathcal{T})} x_i = c(B_h(\mathcal{T}))$ holds for all $h \in H$ for $x \in Core(c)$. Since, $\mathcal{T}$ is a maximal shortest path tree, $|H|$ is the maximum number of branch equalities. The branch equalities have disjoint sets of variables and therefore are linearly independent. Moreover, we have $H \cap MultiHub = \emptyset$. This is because if a multihub agent is in H, then it cannot have two different hubs on two different maximal shortest path trees, and thus cannot be a multihub agent. Therefore, $B_{Hub(i,\mathcal{T})}(\mathcal{T}) \backslash F(i, \mathcal{T}) \neq \emptyset$ for $i \in MultiHub$. Thus, for each $i \in MultiHub$, using

Lemma 3.3, the branch equality

$$\sum_{j \in B_{Hub(i,\mathcal{T})}(\mathcal{T})} x_j = c(B_{Hub(i,\mathcal{T})}(\mathcal{T}))$$

is decomposed into two linearly independent equalities with disjoint sets of variables as

$$\sum_{j \in F(i,\mathcal{T})} x_j = \sum_{j \in F(i,\mathcal{T})} t^{E(j,\mathcal{T})}$$

and

$$\sum_{j \in B_{Hub(i,\mathcal{T})}(\mathcal{T}) \setminus F(i,\mathcal{T})} x_j = c(B_{Hub(i,\mathcal{T})}(\mathcal{T})) - \sum_{j \in F(i,\mathcal{T})} t^{E(j,\mathcal{T})}.$$

Finally, consider two agents $i, j \in MultiHub$ and $i \neq j$. There are two cases to consider. Firstly, $F(i,\mathcal{T})$ and $F(j,\mathcal{T})$ can be disjoint if $j \notin V(i,\mathcal{T})$ and $i \notin V(j,\mathcal{T})$. In this case the equalities introduced by the followers of i and j are linearly independent. Secondly, one of the agents can be on the shortest path of the other agent, such that $j \in V(i,\mathcal{T})$. In this case, $F(i,\mathcal{T}) \subseteq F(j,\mathcal{T})$. We have

$$\sum_{k \in F(i,\mathcal{T})} x_k = \sum_{k \in F(i,\mathcal{T})} t^{E(k,\mathcal{T})} \tag{3.11}$$

and

$$\sum_{k \in F(j,\mathcal{T})} x_k = \sum_{k \in F(j,\mathcal{T})} t^{E(k,\mathcal{T})}$$

which imply

$$\sum_{k \in F(j,\mathcal{T}) \setminus F(i,\mathcal{T})} x_k = \sum_{k \in F(j,\mathcal{T}) \setminus F(i,\mathcal{T})} t^{E(k,\mathcal{T})}. \tag{3.12}$$

Since the variables in (3.11) and (3.12) are disjoint, these equalities are linearly independent. Therefore, in $Core(c)$ of an SPT game, at least $|H| + |MultiHub|$ linearly independent equalities hold. □

Lemma 3.5. *Let $((N \cup \{v_0\}, E), t)$ be an SPT situation. Let $\mathcal{T} \in MaxTree(N)$ be a maximal shortest path tree. Let (N, c) be the corresponding SPT game. Then*

$$dim(Core(c)) \geq |N| - |H| - |MultiHub|.$$

Proof: We prove this lemma by identifying $|N| - |H| - |MultiHub| + 1$ allocations that belong to the core of the SPT games and are affinely independent.

Firstly, we consider the tree solution $\theta((N \cup \{v_0\}, E), t)$, which is unique. Secondly, we consider the $\sigma^i((N \cup \{v_0\}, E), t)$ rule for $i \in N \backslash (H \cup Multihub)$. Recall that $H \cap MultiHub = \emptyset$. Therefore, we have $|N| - |H| - |MultiHub| + 1$ allocations that belong to the core from Theorems 3.1 and 3.5.

We conclude the proof by showing that these $|N| - |H| - |MultiHub| + 1$ allocations are affinely independent. We do so by defining $z^i \in \mathbb{R}^N$ such that $z^i = \sigma^i((N \cup \{v_0\}, E), t) - \theta((N \cup \{v_0\}, E), t)$ and showing that z^i are linearly independent for $i \in N \backslash (H \cup Multihub)$.

Let $j \in N$. The j^{th} element of z_i is $z^i(j) = \sigma^i_j((N \cup \{v_0\}, E), t) - \theta_j((N \cup \{v_0\}, E), t)$.

We consider the following cases for the j^{th} element of z_i.

Case 1. $j \in N \backslash \{i, Hub(i, \mathcal{T})\}$. We have

$$\begin{aligned} z^i(j) &= \sigma^i_j((N \cup \{v_0\}, E), t) - \theta_j((N \cup \{v_0\}, E), t) \\ &= t^{E(j,\mathcal{T})} - t^{E(j,\mathcal{T})} = 0. \end{aligned}$$

Case 2. $j = i$. If player i has a unique shortest path, that is, the shortest path of i is the same on all shortest path trees, then $Hub(i, \mathcal{T})$ is on the unique shortest path of i. Therefore, the cost of the shortest path of i on $\mathcal{T}^{-Hub(i,\mathcal{T})}$ is greater than the cost of the shortest path of i on $\mathcal{T}$, that is, $t^{E(i,\mathcal{T}^{-Hub(i,\mathcal{T})})} > t^{E(i,\mathcal{T})}$. We know that player i does not have a shortest path via a hub in $H \backslash \{Hub(i, \mathcal{T})\}$ since $i \notin Multihub$. There is only one case left to be considered, which is when i has an alternative shortest path via the players in the same branch. Since H is maximal, the players in the same branch as i, excluding the hub of this branch, cannot be a hub on any shortest path tree. Therefore, if there exists an alternative shortest path via the same branch, this path has to go through $Hub(i, \mathcal{T})$. Thus, also in this final case, $t^{E(i,\mathcal{T}^{-Hub(i,\mathcal{T})})} > t^{E(i,\mathcal{T})}$. We have

$$\begin{aligned} z^i(i) &= \sigma^i_i((N \cup \{v_0\}, E), t) - \theta_i((N \cup \{v_0\}, E), t) \\ &= t^{E(i,\mathcal{T}^{-Hub(i,\mathcal{T})})} - t^{E(i,\mathcal{T})} > 0. \end{aligned}$$

Case 3. $j = Hub(i, \mathcal{T})$. Following from the argument of the previous case, we have

$$\begin{aligned} z^i(Hub(i,\mathcal{T})) =& \sigma^i_{Hub(i,\mathcal{T})}((N \cup \{v_0\}, E), t) - \theta_{Hub(i,\mathcal{T})}((N \cup \{v_0\}, E), t) \\ = & \quad t(\mathcal{T}) - \sum_{k \in N \backslash \{i, Hub(i,\mathcal{T})\}} t^{E(k,\mathcal{T})} \\ & - t^{E(i,\mathcal{T}^{-Hub(i,\mathcal{T})})} - t^{E(Hub(i,\mathcal{T}),\mathcal{T})} \\ =& \sum_{k \in N} t^{E(k,\mathcal{T})} - \sum_{k \in N \backslash \{i, Hub(i,\mathcal{T})\}} t^{E(k,\mathcal{T})} \\ & - t^{E(i,\mathcal{T}^{-Hub(i,\mathcal{T})})} - t^{E(Hub(i,\mathcal{T}),\mathcal{T})} \\ =& t^{E(i,\mathcal{T})} + t^{E(Hub(i,\mathcal{T}),\mathcal{T})} - t^{E(i,\mathcal{T}^{-Hub(i,\mathcal{T})})} - t^{E(Hub(i,\mathcal{T}),\mathcal{T})} \\ =& t^{E(i,\mathcal{T})} - t^{E(i,\mathcal{T}^{-Hub(i,\mathcal{T})})} < 0. \end{aligned}$$

Thus,

$$z^i(j) = \begin{cases} 0 & \text{for } j \in N \backslash \{i, Hub(i,\mathcal{T})\}, \\ t^{E(i,\mathcal{T}^{-Hub(i,\mathcal{T})})} - t^{E(i,\mathcal{T})} & \text{for } j = i, \\ t^{E(i,\mathcal{T})} - t^{E(i,\mathcal{T}^{-Hub(i,\mathcal{T})})} & \text{for } j = Hub(i,\mathcal{T}). \end{cases}$$

Let $\hat{e}_k \in \mathbb{R}^N$ be the unit vector with a 1 in the k^{th} coordinate and 0's elsewhere. We get

$$z^i = (t^{E(i,\mathcal{T}^{-Hub(i,\mathcal{T})})} - t^{E(i,\mathcal{T})}) \cdot (\hat{e}_i - \hat{e}_{Hub(i,\mathcal{T})}).$$

Since $(\hat{e}_i - \hat{e}_{Hub(i,\mathcal{T})})$ for $i \in N \backslash (H \cup Multihub)$ are linearly independent and $t^{E(i,\mathcal{T}^{-Hub(i,\mathcal{T})})} - t^{E(i,\mathcal{T})} > 0$, z_i for $i \in N \backslash (H \cup Multihub)$ are linearly independent. □

3.4.4 Facets of the Core (Unique Shortest Path Tree Case)

In this section, we identify a class of facets of the core of SPT games that correspond to SPT problems with a unique shortest path tree. Based on the simulations we have performed in PORTA, we could not identify a class of coalitions that always defines facets of the core of the SPT games that correspond to SPT problems with multiple optimal solutions. Furthermore, note that the class of facets identified in this section does not generate all the facets of the core of an SPT game that corresponds to an SPT problem with a unique shortest path tree. Finally, we would like to highlight that we could not identify any other class of coalitions in our simulations such that the corresponding equalities always defined facets of the core of the SPT games.

We have the following result.

Theorem 3.9. *Let* $((N \cup \{v_0\}, E), t) \in \mathcal{USPT}(N)$. *Let* $\mathcal{T}$ *be the shortest path tree. Let* $j \in N \backslash H(\mathcal{T})$. *Let* (N, c) *be the corresponding SPT game. Then*

$$\sum_{k \in F(j,\mathcal{T})} x_k \geq c(N) - c(N \backslash F(j, \mathcal{T}))$$

defines a facet of $Core(c)$.

Proof: From Theorem 3.2, the dimension of the core of the SPT games corresponding to SPT problems with a unique shortest path is $|N| - |H(\mathcal{T})|$. Therefore, any facet of the core is of dimension $|N| - |H(\mathcal{T})| - 1$. We prove that

$$\sum_{k \in F(j,\mathcal{T})} x_k \geq c(N) - c(N \backslash F(j, \mathcal{T}))$$

defines a facet by identifying $|N| - |H(\mathcal{T})|$ allocations that are in the core, satisfy this inequality as an equality and are affinely independent.

Firstly, we consider the tree solution θ, which is unique. Secondly, we let $j \in N \backslash H(\mathcal{T})$ and consider allocations generated by the $\rho^{i,F(j,\mathcal{T})}$ rule for $i \in N \backslash H(\mathcal{T})$ and $i \neq j$. There are $|N| - |H(\mathcal{T})| - 1$ such allocations. Since the tree solution θ and the $\rho^{i,F(j,\mathcal{T})}$ rule generate core allocations from Theorems 3.1 and 3.6, respectively, we can generate $|N| - |H(\mathcal{T})|$ allocations that belong to the core.

We proceed by showing that these $|N| - |H(\mathcal{T})|$ allocations lie on the facet. Consider the SPT problem $((N \backslash S \cup \{v_0\}, E^{N \backslash S}), t^{N \backslash S})$ and recall that we denote an optimal solution to this problem by $\mathcal{T}^{-S}$. Firstly, we have

$$\begin{aligned} c(N) - c(N \backslash F(j, \mathcal{T})) &= \sum_{k \in N} t^{E(k,\mathcal{T})} - \sum_{k \in N \backslash F(j,\mathcal{T})} t^{E(k,\mathcal{T}^{-F(j,\mathcal{T})})} \\ &= \sum_{k \in N} t^{E(k,\mathcal{T})} - \sum_{k \in N \backslash F(j,\mathcal{T})} t^{E(k,\mathcal{T})} \\ &= \sum_{k \in F(j,\mathcal{T})} t^{E(k,\mathcal{T})} \end{aligned}$$

where the second equality follows from Lemma 2.1.

For the tree solution, we have $\sum_{k\in F(j,\mathcal{T})} \theta_k((N\cup\{v_0\},E),t) = \sum_{k\in F(j,\mathcal{T})} t^{E(k,\mathcal{T})}$.

For the $\rho^{i,F(j,\mathcal{T})}$ rule, we have two cases.

Case 1. $i \in F(j,\mathcal{T})$. Then $Sup(i,F(j,\mathcal{T})) = j$ and thus $Sup(i,F(j,\mathcal{T})) \in F(j,\mathcal{T})$. Therefore,

$$\begin{aligned}\sum_{k\in F(j,\mathcal{T})} \rho_k^{i,F(j,\mathcal{T})}((N\cup\{v_0\},E),t) = & \sum_{k\in F(j,\mathcal{T})\setminus\{i,Sup(i,F(j,\mathcal{T}))\}} t^{E(k,\mathcal{T})} \\ & + t^{E(i,\mathcal{T}^{-Sup(i,F(j,\mathcal{T}))})} \\ & + t(\mathcal{T}) - \sum_{k\in N\setminus\{i,Sup(i,F(j,\mathcal{T}))\}} t^{E(k,\mathcal{T})} \\ & - t^{E(i,\mathcal{T}^{-Sup(i,F(j,\mathcal{T}))})} \\ = & \sum_{k\in N} t^{E(k,\mathcal{T})} - \sum_{k\in N\setminus F(j,\mathcal{T})} t^{E(k,\mathcal{T})} \\ = & \sum_{k\in F(j,\mathcal{T})} t^{E(k,\mathcal{T})}.\end{aligned}$$

Case 2. $i \notin F(j,\mathcal{T})$. Then $Sup(i,F(j,\mathcal{T})) = Hub(i,\mathcal{T})$ and thus $Sup(i,F(j,\mathcal{T})) \notin F(j,\mathcal{T})$. Therefore,

$$\sum_{k\in F(j,\mathcal{T})} \rho_k^{i,F(j,\mathcal{T})}((N\cup\{v_0\},E),t) = \sum_{k\in F(j,\mathcal{T})} t^{E(k,\mathcal{T})}.$$

Thus, the $|N| - |H(\mathcal{T})|$ points considered satisfy the facet inducing inequality as an equality.

We conclude the proof by showing that these $|N| - |H(\mathcal{T})|$ allocations are affinely independent. We do so by defining $z^i \in \mathbb{R}^N$ such that $z^i = \rho^{i,F(j,\mathcal{T})}((N\cup\{v_0\},E),t) - \theta((N\cup\{v_0\},E),t)$ and showing that z^i are linearly independent for $i,j \in N\setminus H(\mathcal{T})$ and $i \neq j$. Let $k \in N$. The k^{th} element of z^i is $z^i(k) = \rho_k^{i,F(j,\mathcal{T})}((N\cup\{v_0\},E),t) - \theta_k((N\cup\{v_0\},E),t)$.

We consider the following cases for the k^{th} element of z^i.

Case 1. $k \in N\setminus\{i, Sup(i,F(j,\mathcal{T}))\}$. We have

$$z^i(k) = \rho_k^{i,F(j,\mathcal{T})}((N\cup\{v_0\},E),t) - \theta_k((N\cup\{v_0\},E),t)$$

$$= t^{E(k,\mathcal{T})} - t^{E(k,\mathcal{T})} = 0.$$

Case 2. $k = i$. We have

$$\begin{aligned} z^i(i) &= \rho_i^{i,F(j,\mathcal{T})}((N \cup \{v_0\}, E), t) - \theta_i((N \cup \{v_0\}, E), t) \\ &= t^{E(i,\mathcal{T}^{-Sup(i,F(j,\mathcal{T}))})} - t^{E(i,\mathcal{T})}. \end{aligned}$$

Case 3. $k = Sup(i, F(j, \mathcal{T}))$. We have

$$\begin{aligned} z^i(Sup(i, F(j, \mathcal{T}))) = \quad & \rho^{i,F(j,\mathcal{T})}_{Sup(i,F(j,\mathcal{T}))}((N \cup \{v_0\}, E), t) \\ & - \theta_{Sup(i,F(j,\mathcal{T}))}((N \cup \{v_0\}, E), t) \\ = \quad & t(\mathcal{T}) - \sum_{k \in N \setminus \{i, Sup(i,F(j,\mathcal{T}))\}} t^{E(k,\mathcal{T})} \\ & - t^{E(i,\mathcal{T}^{-Sup(i,F(j,\mathcal{T}))})} - t^{E(Sup(i,F(j,\mathcal{T})),\mathcal{T})} \\ =& \sum_{k \in N} t^{E(k,\mathcal{T})} - \sum_{k \in N \setminus \{i, Sup(i,F(j,\mathcal{T}))\}} t^{E(k,\mathcal{T})} \\ & - t^{E(i,\mathcal{T}^{-Sup(i,F(j,\mathcal{T}))})} - t^{E(Sup(i,F(j,\mathcal{T})),\mathcal{T})} \\ =& t^{E(i,\mathcal{T})} + t^{E(Sup(i,F(j,\mathcal{T})),\mathcal{T})} - t^{E(i,\mathcal{T}^{-Sup(i,F(j,\mathcal{T}))})} \\ & - t^{E(Sup(i,F(j,\mathcal{T})),\mathcal{T})} \\ =& t^{E(i,\mathcal{T})} - t^{E(i,\mathcal{T}^{-Sup(i,F(j,\mathcal{T}))})}. \end{aligned}$$

Thus,

$$z^i(k) = \begin{cases} 0 & \text{for } k \in N \setminus \{i, Sup(i, F(j, \mathcal{T}))\} \\ t^{E(i,\mathcal{T}^{-Sup(i,F(j,\mathcal{T}))})} - t^{E(i,\mathcal{T})} & \text{for } k = i \\ t^{E(i,\mathcal{T})} - t^{E(i,\mathcal{T}^{-Sup(i,F(j,\mathcal{T}))})} & \text{for } k = Sup(i, F(j, \mathcal{T})). \end{cases}$$

We get

$$z^i = (t^{E(i,\mathcal{T}^{-Sup(i,F(j,\mathcal{T}))})} - t^{E(i,\mathcal{T})}) \cdot (\hat{e}_i - \hat{e}_{Sup(i,F(j,\mathcal{T}))}).$$

Since $((N \cup \{v_0\}, E), t) \in \mathcal{USPT}(N)$, the shortest path of i on $\mathcal{T}$ is unique. Furthermore, since $Sup(i, F(j, \mathcal{T}))$ is on the unique shortest path of i on $\mathcal{T}$, the cost of the shortest path of i on $\mathcal{T}^{-Sup(i,F(j,\mathcal{T}))}$ is greater than the cost of the shortest path of i on

$\mathcal{T}$. This gives us $t^{E(i,\mathcal{T}^{-Sup(i,F(j,\mathcal{T}))})} - t^{E(i,\mathcal{T})} > 0$. Moreover, $(\hat{e}_i - \hat{e}_{Sup(i,F(j,\mathcal{T}))})$ are linearly independent for $i \in N \backslash H(\mathcal{T})$ and $i \neq j$. Thus, z^i are linearly independent for $i \in N \backslash H(\mathcal{T})$ and $i \neq j$. □

Example 3.15. Consider the SPT game displayed in Figure 3.8 and its core as illustrated in Figure 3.9. We have $N \backslash H(\mathcal{T}) = \{2, 3\}$. Furthermore, $F(2, \mathcal{T}) = \{2, 3\}$ and $F(3, \mathcal{T}) = \{3\}$. The facet inducing inequalities generated by Theorem 3.9 are $x_2 + x_3 \geq 8$ and $x_3 \geq 5$. Observe that the remaining facet inducing inequalities $x_2 \leq 9$, $x_2 + x_3 \leq 20$ and $x_3 \leq 14$ do not being to the class of facets generated by 3.9. ◇

3.5 Redundant Core Inequalities of the Shortest Path Tree Games

This section identifies a collection of coalitions for which the corresponding inequalities are redundant in the description of the core of the SPT games. We start with some definitions. Let $x \in R^N$ be an allocation for the SPT game (N, c) and let x_i denote the cost allocated to a player i by x. Recall that the core of (N, c) is defined as

$$Core(c) = \left\{ x \in \mathbb{R}^N | \sum_{i \in N} x_i = c(N) \text{ and } \sum_{i \in S} x_i \leq c(S) \text{ for all } S \subset N \right\}.$$

The collection of inequalities $\sum_{i \in S} x_i \leq c(S)$ for all $S \subset N$ guarantees that the players in a coalition S are never charged more than what they would have paid if they broke away from the rest of the players. Thus, this principle is referred to as the *stand-alone* principle (Moulin, 1988). Now, consider the *no-subsidy* principle (Moulin, 1988) that ensures that the players in a coalition S must at least pay the marginal cost of being served. Therefore, this principle implies that $\sum_{i \in S} x_i \geq c(N) - c(N \backslash S)$ holds for all $S \subset N$. In fact, the no-subsidy principle is equivalent to the stand-alone principle for $x \in Core(c)$. Firstly, the no-subsidy principle implies the stand-alone principle since $\sum_{i \in N} x_i = c(N)$ gives us $\sum_{i \in S} x_i \geq \sum_{i \in N} x_i - c(N \backslash S)$ for all $S \subset N$, which in turn is equivalent to $\sum_{i \in N \backslash S} x_i \leq c(N \backslash S)$ for all $S \subset N$. Secondly, the stand-alone principle implies the no-subsidy principle since subtracting $\sum_{i \in N} x_i$ from both sides of $\sum_{i \in S} x_i \leq c(S)$ and $\sum_{i \in N} x_i = c(N)$ give $-\sum_{i \in N \backslash S} x_i \leq c(S) - c(N)$, which is equivalent to $\sum_{i \in N \backslash S} x_i \geq c(N) - c(S)$ for all $S \subseteq N$. In this section, we consider the definition of the core based on the no-subsidy principle, which is

$$Core(c) = \left\{ x \in \mathbb{R}^N | \sum_{i \in N} x_i = c(N) \text{ and } \sum_{i \in S} x_i \geq c(N) - c(N \backslash S) \text{ for all } S \subset N \right\}.$$

Before identifying a collection of coalitions for which the corresponding inequalities are always redundant in the description of the core of the SPT games, we introduce some notation.

Let $((N \cup \{v_0\}, E), t)$ be an SPT problem and let $\mathcal{T}$ be a shortest path tree. Let $S \subset N$. Let

$$V^S(i, \mathcal{T}) = \{k : k \in S \cap V(i, \mathcal{T})\}.$$

The vertices in $V^S(i, \mathcal{T})$ are the vertices on the shortest path of i on $\mathcal{T}$ and in S. A player i in S is called an *S-hub* if $V^S(i, \mathcal{T}) = \{i\}$. Thus, S-hubs are players who do not have any other player in S on their shortest path on $\mathcal{T}$. Let

$$H^S(\mathcal{T}) = \{i : V^S(i, \mathcal{T}) = \{i\}\}$$

denote the set of S-hubs on $\mathcal{T}$. Let $h \in H^S(\mathcal{T})$, then

$$B_h^S(\mathcal{T}) = S \cap F(h, \mathcal{T})$$

is called the *S-branch induced by S-hub h on $\mathcal{T}$*. Thus, $B_h^S(\mathcal{T})$ denotes the set of players in S that are followers of S-hub h on $\mathcal{T}$. Observe that $\{B_h^S(\mathcal{T}) : h \in H^S(\mathcal{T})\}$ is a partition of S.

Example 3.16. Consider the shortest path tree $\mathcal{T}$ illustrated in Figure 3.12. Let $S = \{1, 3, 5, 7\}$. Then $V^S(7, \mathcal{T}) = \{5, 7\}$. The set of S-hubs is $H^S(\mathcal{T}) = \{1, 5\}$ since there does not exist any other player in S that is on the shortest path of these players on $\mathcal{T}$. The two S-branches on $\mathcal{T}$, which form a partition of S, are $B_1^S(\mathcal{T}) = \{1, 3\}$ and $B_5^S(\mathcal{T}) = \{5, 7\}$. $\diamond$

We furthermore let $F^{-S}(H^S(\mathcal{T}))$ denote the set of players in $N \backslash S$ that have an S-hub on their shortest path on $\mathcal{T}$. Let

$$B_h^{-S}(\mathcal{T}) = F^{-S}(H^S(\mathcal{T})) \cap F(h, \mathcal{T}).$$

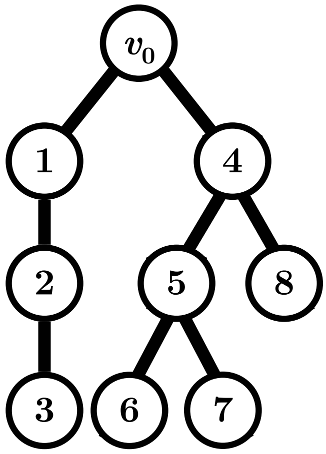


Figure 3.12: A shortest path tree.

Thus, $B_h^{-S}(\mathcal{T})$ denotes the set of players that are in $F^{-S}(H^S(\mathcal{T}))$ and that are the followers of S-hub h on $\mathcal{T}$. Observe that $\{B_h^{-S}(\mathcal{T}) : h \in H^S(\mathcal{T})\}$ is a partition of $F^{-S}(H^S(\mathcal{T}))$.

Example 3.17. Consider the shortest path tree $\mathcal{T}$ illustrated in Figure 3.12. Let $S = \{1, 3, 5, 7\}$. Then, $F^{-S}(H^S(\mathcal{T})) = \{2, 6\}$ is the set of players in $N \backslash S$ that have an S-hub on their shortest path on $\mathcal{T}$, and this set is partitioned into $B_1^{-S}(\mathcal{T}) = \{2\}$ and $B_5^{-S}(\mathcal{T}) = \{6\}$. $\diamond$

Let $\mathcal{S}^B(\mathcal{T}) = \{B_h^S(\mathcal{T}) : S \in 2^N \backslash \{\emptyset, N\}, h \in H^S(\mathcal{T})\}$ denote the collection of S-branches for $S \in 2^N \backslash \{\emptyset, N\}$. Note that $|H^S(\mathcal{T})| = 1$ for a coalition $S \in \mathcal{S}^B(\mathcal{T})$ and $|H^S(\mathcal{T})| > 1$ for a coalition $S \in (2^N \backslash \{\emptyset, N\}) \backslash \mathcal{S}^B(\mathcal{T})$. Furthermore, for $S \in (2^N \backslash \{\emptyset, N\}) \backslash \mathcal{S}^B(\mathcal{T})$, the S-branches $B_h^S(\mathcal{T}) \in \mathcal{S}^B(\mathcal{T})$ for all $h \in H^S(\mathcal{T})$.

Example 3.18. Consider the shortest path tree $\mathcal{T}$ illustrated in Figure 3.12. We have $\mathcal{S}^B(\mathcal{T}) = \{\{1\}, \{1, 2\}, \{1, 3\}, \{1, 2, 3\}, \{2\}, \{2, 3\}, \{3\}, \{4\}, \{4, 5\}, \{4, 6\}, \{4, 7\}, \{4, 8\}, \{4, 5, 6\}, \{4, 5, 7\}, \{4, 5, 8\}, \{4, 6, 7\}, \{4, 6, 8\}, \{4, 7, 8\}, \{4, 5, 6, 7\}, \{4, 5, 6, 8\}, \{4, 5, 7, 8\}, \{4, 6, 7, 8\}, \{4, 5, 6, 7, 8\}, \{5\}, \{5, 6\}, \{5, 7\}, \{5, 6, 7\}, \{6\}, \{7\}, \{8\}\}$.Thus, of the 254 coalitions in $S \in 2^N \backslash \{\emptyset, N\}$ only 30 are in $\mathcal{S}^B(\mathcal{T})$. Furthermore, let $S = \{5, 6, 8\}$. Then $S \in (2^N \backslash \{\emptyset, N\}) \backslash \mathcal{S}^B(\mathcal{T})$, $H^S(\mathcal{T}) = \{5, 8\}$, and S-branches $B_1^S(\mathcal{T}) = \{5, 6\}$ and $B_5^S(\mathcal{T}) = \{8\}$ are in $\mathcal{S}^B(\mathcal{T})$. $\diamond$

The next lemma states that the inequality $\sum_{i \in S} x_i \geq c(N) - c(N \backslash S)$ corresponding to a coalition $S \in (2^N \backslash \{\emptyset, N\}) \backslash \mathcal{S}^B(\mathcal{T})$ is redundant in the description of the core.

Lemma 3.6. *Let $((N \cup \{v_0\}, E), t)$ be an SPT problem and let $\mathcal{T}$ be a shortest path tree. Let $S \in (2^N \backslash \{\emptyset, N\}) \backslash \mathcal{S}^B(\mathcal{T})$. Let (N, c) be the corresponding SPT game. Let $x \in \mathbb{R}^N$. Then*

$$\sum_{i \in S} x_i \geq c(N) - c(N \backslash S)$$

is redundant in the description of $Core(c)$.

Proof: We show that there exists $|H^S(\mathcal{T})| > 1$ valid inequalities

$$\sum_{i \in B_h^S(\mathcal{T})} x_i \geq c(N) - c(N \backslash B_h^S(\mathcal{T})) \text{ for all } h \in H^S(\mathcal{T})$$

whose sum gives us the inequality

$$\sum_{h \in H^S(\mathcal{T})} \sum_{i \in B_h^S(\mathcal{T})} x_i \geq \sum_{h \in H^S(\mathcal{T})} \left[c(N) - c(N \backslash B_h^S(\mathcal{T}))\right] \tag{3.13}$$

which dominates

$$\sum_{i \in S} x_i \geq c(N) - c(N \backslash S). \tag{3.14}$$

Since $\{B_h^S(\mathcal{T}) : h \in H^S(\mathcal{T})\}$ is a partition of S, the left hand sides of (3.13) and (3.14) are equal. Let α_0 denote the right hand side value of (3.13), and let β_0 denote the right hand side value of (3.14). In order to show that (3.14) is dominated by (3.13), we need to show that $\beta_0 \leq \alpha_0$.

We start with the following two results. Recall that $\mathcal{T}^{-S}$ denotes a shortest path tree of the SPT problem $((N \backslash S \cup \{v_0\}, E^{N \backslash S}), t^{N \backslash S})$. Firstly,

$$t^{E(i, \mathcal{T})} = t^{E(i, \mathcal{T}^{-S})} \text{ for } i \in (N \backslash S) \backslash F^{-S}(H^S(\mathcal{T})). \tag{3.15}$$

Since $i \notin F^{-S}(H^S(\mathcal{T}))$, i does not have any player in $H^S(\mathcal{T})$ on its shortest path on $\mathcal{T}$, which in turn implies that it does not have any player in S on its shortest path on $\mathcal{T}$. Hence, (3.15) holds. Secondly,

$$t^{E(i, \mathcal{T})} = t^{E(i, \mathcal{T}^{-B_h^S(\mathcal{T})})} \text{ for } i \in (N \backslash B_h^S(\mathcal{T})) \backslash B_h^{-S}(\mathcal{T}). \tag{3.16}$$

Since $i \notin B_h^{-S}(\mathcal{T})$, i does not have h on its shortest path on $\mathcal{T}$, which in turn implies that it does not have any player in $B_h^S(\mathcal{T})$ on its shortest path on $\mathcal{T}$. Hence, (3.16) holds.

Now, we are ready to show that $\beta_0 \leq \alpha_0$. We have

$$\begin{aligned}
\beta_0 &= c(N) - c(N\backslash S) \\
&= t(\mathcal{T}) - t(\mathcal{T}^{-S}) \\
&= \sum_{i\in N} t^{E(i,\mathcal{T})} - \sum_{i\in N\backslash S} t^{E(i,\mathcal{T}^{-S})} \\
&= \sum_{i\in S} t^{E(i,\mathcal{T})} + \sum_{i\in N\backslash S} \left[t^{E(i,\mathcal{T})} - t^{E(i,\mathcal{T}^{-S})}\right] \\
&= \sum_{i\in S} t^{E(i,\mathcal{T})} + \sum_{i\in F^{-S}(H^S(\mathcal{T}))} \left[t^{E(i,\mathcal{T})} - t^{E(i,\mathcal{T}^{-S})}\right] \\
&= \sum_{h\in H^S(\mathcal{T})} \sum_{i\in B_h^S(\mathcal{T})} t^{E(i,\mathcal{T})} + \sum_{h\in H^S(\mathcal{T})} \sum_{i\in B_h^{-S}(\mathcal{T})} \left[t^{E(i,\mathcal{T})} - t^{E(i,\mathcal{T}^{-S})}\right] \\
&\leq \sum_{h\in H^S(\mathcal{T})} \sum_{i\in B_h^S(\mathcal{T})} t^{E(i,\mathcal{T})} + \sum_{h\in H^S(\mathcal{T})} \sum_{i\in B_h^{-S}(\mathcal{T})} \left[t^{E(i,\mathcal{T})} - t^{E(i,\mathcal{T}^{-B_h^S(\mathcal{T})})}\right] \\
&= \sum_{h\in H^S(\mathcal{T})} \left[\sum_{i\in B_h^S(\mathcal{T})} t^{E(i,\mathcal{T})} + \sum_{i\in B_h^{-S}(\mathcal{T})} \left[t^{E(i,\mathcal{T})} - t^{E(i,\mathcal{T}^{-B_h^S(\mathcal{T})})}\right]\right] \\
&= \sum_{h\in H^S(\mathcal{T})} \left[\sum_{i\in B_h^S(\mathcal{T})} t^{E(i,\mathcal{T})} + \sum_{i\in N\backslash B_h^S(\mathcal{T})} \left[t^{E(i,\mathcal{T})} - t^{E(i,\mathcal{T}^{-B_h^S(\mathcal{T})})}\right]\right] \\
&= \sum_{h\in H^S(\mathcal{T})} \left[\sum_{i\in N} t^{E(i,\mathcal{T})} - \sum_{i\in N\backslash B_h^S(\mathcal{T})} t^{E(i,\mathcal{T}^{-B_h^S(\mathcal{T})})}\right] \\
&= \sum_{h\in H^S(\mathcal{T})} \left[c(N) - c(N\backslash B_h^S(\mathcal{T}))\right] \\
&= \alpha_0
\end{aligned}$$

where the fifth equality follows from (3.15), the sixth inequality follows since $\{B_h^S(\mathcal{T}) : h \in H^S(\mathcal{T})\}$ and $\{B_h^{-S}(\mathcal{T}) : h \in H^S(\mathcal{T})\}$ are partitions of S and $F^{-S}(H^S(\mathcal{T}))$ respectively, the inequality follows since $t^{E(i,\mathcal{T}^{-B_h^S(\mathcal{T})})} \leq t^{E(i,\mathcal{T}^{-S})}$ from Lemma 2.3, and the eighth equality follows from (3.16). □

We next consider an example illustrating the above result.

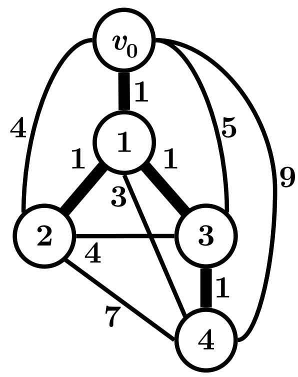


Figure 3.13: An SPT problem.

Example 3.19. Consider the SPT problem illustrated in Figure 3.13 and the corresponding SPT game (N, c). The core of (N, c) using the no-subsidy principle is described by the constraints

$$
\begin{aligned}
x_1 + x_2 + x_3 + x_4 &= 8 \\
x_1 + x_2 + x_3 \qquad &\geq -1 \\
x_1 + x_2 \qquad + x_4 &\geq 3 \\
x_1 \qquad + x_3 + x_4 &\geq 4 \\
x_2 + x_3 + x_4 &\geq 7 \\
x_1 + x_2 \qquad\qquad &\geq -3 \\
x_1 \qquad + x_3 \qquad &\geq -5 \\
x_1 \qquad\qquad + x_4 &\geq -1 \\
x_2 + x_3 \qquad &\geq 3 \\
x_2 \qquad + x_4 &\geq 5 \\
x_3 + x_4 &\geq 5 \\
x_1 \qquad\qquad\qquad &\geq -7 \\
x_2 \qquad\qquad &\geq 2 \\
x_3 \qquad &\geq 1 \\
x_4 &\geq 3.
\end{aligned}
$$

There exists an inequality for each of the coalitions $S \subset N$ in the description of the core. Nonetheless, in order to characterise the core not all of these inequalities are always needed. The inequalities that are needed to describe the core depend on the coalitional costs, which are determined by the edge costs. Therefore, it is not trivial to identify the most compact description of the core for every SPT game. However, using the result of Lemma 3.6, we are able to identify a collection of inequalities that are always dominated by other inequalities independent of the edge costs, and thus can always be eliminated from the description of the core. Firstly, consider $x_2 + x_3 \geq 3$. This inequality is redundant since it is implied by $x_2 \geq 2$ and $x_3 \geq 1$. Similarly, $x_2+x_4 \geq 5$ and $x_2+x_3+x_4 \geq 7$ are redundant since they are implied by $x_2 \geq 2$ and $x_4 \geq 3$, and by $x_2 \geq 2$ and $x_3 + x_4 \geq 5$ respectively. Therefore, if we remove the inequalities corresponding to coalitions $\{2,3\}$, $\{2,4\}$ and $\{2,3,4\}$, the core does not change. For this example, $\mathcal{S}^B(\mathcal{T}) = \{\{1\},\{1,2\},\{1,3\},\{1,4\},\{1,2,3\},\{1,2,4\},\{1,3,4\},\{2\},\{3\},\{3,4\},\{4\}\}$. Observe that the coalitions $\{2,3\}$, $\{2,4\}$ and $\{2,3,4\}$ are not in $\mathcal{S}^B(\mathcal{T})$ and therefore, the corresponding inequalities are always redundant in the description of the core.

We conclude this example by providing the most compact description of the core of this game generated by PORTA below.

$$\begin{aligned}
x_1 + x_2 + x_3 + x_4 &= 8\\
x_1 + x_2 + x_3 \qquad &\geq -1\\
x_1 + x_2 \qquad + x_4 &\geq 3\\
x_1 \qquad + x_3 + x_4 &\geq 4\\
x_1 + x_2 \qquad\qquad &\geq -3\\
x_3 + x_4 &\geq 5\\
x_2 \qquad\qquad &\geq 2\\
x_3 \qquad &\geq 1\\
x_4 &\geq 3.
\end{aligned}$$

Observe that a number inequalities including the ones corresponding to $\{2,3\}$, $\{2,4\}$ and $\{2,3,4\}$, which are generated by the method proposed above, are redundant in the description of the core of (N,c). ◇

3.6 Reduced Description of the Core of the Shortest Path Tree Games

This section first presents the theorem on the reduced description of the core using our previous results on branches and redundant inequalities. Secondly, this section illustrates the reduction in the number of coalitions whose costs need to be computed to describe the core of the SPT games.

For simplicity of notation, let us denote the collection of branches of a shortest path tree $\mathcal{T}$ by $\mathcal{B}(\mathcal{T})$, that is, $\mathcal{B}(\mathcal{T}) = \{B_h(\mathcal{T}) : h \in H(\mathcal{T})\}$.

Theorem 3.10. *Let $((N \cup \{v_0\}, E), t)$ be an SPT problem and let $\mathcal{T}$ be a shortest path tree. Let (N, c) be the corresponding SPT game. The core of (N, c) is defined as*

$$Core(c) = \left\{ x \in \mathbb{R}^N | \sum_{i \in S} x_i = c(S)\ \forall S \in \mathcal{B}(\mathcal{T}), \sum_{i \in S} x_i \leq c(S)\ \forall N \backslash S \in \mathcal{S}^B(\mathcal{T}) \backslash \mathcal{B}(\mathcal{T}) \right\}.$$

Proof: Firstly, from Lemma 3.2, we know that in the core of the SPT games the total cost allocated to the players in a branch equals the cost of the coalition formed by this branch, that is, $\sum_{i \in S} x_i = c(S)$ for all $S \in \mathcal{B}(\mathcal{T})$. Adding these equalities imply the efficiency condition $\sum_{i \in N} x_i = c(N)$ from Lemma 2.2, which states that the cost of a shortest path tree is the sum of the costs of its tree branches.

Secondly, from Lemma 3.6, we know that $\sum_{i \in S} x_i \geq c(N) - c(N \backslash S)$ for $S \in \mathcal{S}^B(\mathcal{T})$ are non-redundant in the description of the core of the SPT games. Since we have $\sum_{i \in N} x_i = c(N)$, we get $\sum_{i \in S} x_i \geq \sum_{i \in N} x_i - c(N \backslash S)$ for all $S \in \mathcal{S}^B(\mathcal{T})$. Rearranging this inequality gives us $\sum_{i \in N \backslash S} x_i \leq c(N \backslash S)$ for all $S \in \mathcal{S}^B(\mathcal{T})$ and exchanging S with $N \backslash S$ implies that $\sum_{i \in S} x_i \leq c(S)$ for all $N \backslash S \in \mathcal{S}^B(\mathcal{T})$ are non-redundant in the description of the core of the SPT games.

Finally, consider $S \in \mathcal{B}(\mathcal{T})$, that is, a branch of N with respect to $\mathcal{T}$. For such S, the player that lies on the shortest path of all other players in a branch is in fact the hub of this branch, so $H^S(\mathcal{T}) = \{h\}$ and $|H^S(\mathcal{T})| = 1$. Since we know that S such that $|H^S(\mathcal{T})| = 1$ are in $\mathcal{S}^B(\mathcal{T})$, we get $\mathcal{B}(\mathcal{T}) \subseteq \mathcal{S}^B(\mathcal{T})$. In other words, the inequalities corresponding to the branches of N with respect to $\mathcal{T}$ are always non-redundant from Lemma 3.6. However, we do not need to consider the inequalities corresponding to coalitions in $\mathcal{B}(\mathcal{T})$ as inequalities since they are in fact always satisfied as equalities from Lemma 3.2. Therefore, it is sufficient to consider the inequalities $\sum_{i \in S} x_i \leq c(S)$ for all $N \backslash S \in \mathcal{S}^B(\mathcal{T}) \backslash \mathcal{B}(\mathcal{T})$. □

Next, we illustrate the reduction in the number of coalitions, which are required to describe the core of the SPT games, introduced by Theorem 3.10. The theorem implies that to describe the core of the SPT games, we need to compute the costs of $|\mathcal{S}^B(\mathcal{T})|$ non-redundant coalitions in total for the equalities and the inequalities of the reduced description of the core of the SPT games.

Let $S \in 2^N \backslash \{\emptyset, N\}$. In the previous section, we have discussed that the non-redundant coalitions $S \in \mathcal{S}^B(\mathcal{T})$ are $S \subset N$ for which $|H^S(\mathcal{T})| = 1$ where $H^S(\mathcal{T})$ is the set of S-hubs. In other words, there exists a single player that lies on the shortest path of all other players in S for every coalition in $\mathcal{S}^B(\mathcal{T})$. Recall that $i \in F(i, \mathcal{T})$ where $F(i, \mathcal{T})$ denotes the set of the followers of a player i on $\mathcal{T}$. In order to compute the number of coalitions that are in $\mathcal{S}^B(\mathcal{T})$, we need to consider each player $i \in N$ and all the coalitions it forms with the players in $F(i, \mathcal{T}) \backslash \{i\}$. Therefore, each $i \in N$ induces $2^{|F(i,\mathcal{T})-1|}$ coalitions that belong to $\mathcal{S}^B(\mathcal{T})$ except for when the shortest path tree formed by all the players has exactly one hub, who is a player that is directly connected to v_0, and if this is the case h lies on the shortest paths of all players in $N \backslash \{h\}$ on $\mathcal{T}$. Since the coalition that h forms with all of its followers is in fact the set of all players N, h induces $2^{|F(i,\mathcal{T})-1|} - 1$ coalitions that are in $\mathcal{S}^B(\mathcal{T})$.

Consider $\mathcal{T}_1$ in Figure 3.14 (see next page) where player 1 is the only hub. We have $|\mathcal{S}^B(\mathcal{T}_1)| = 2^{n-1} - 1 + \sum_{i=2}^{n} 2^{n-i} = 2^n - 2$. Therefore, for this case none of the coalitions in $S \in 2^N \backslash \{\emptyset, N\}$ is redundant. Next, consider $\mathcal{T}_2$ in Figure 3.14. In this case, all the players are hubs and have no followers other than themselves, and thus $\mathcal{S}^B(\mathcal{T}_2) = \{\{1\}, \{2\}, \ldots, \{n\}\}$ and the coalitions in $S \in 2^N \backslash \{\emptyset, \{1\}, \{2\}, \ldots, \{n\}, N\}$ are redundant. Observe that $\mathcal{T}_2$ is a star graph rooted at v_0 and has depth 1. Next consider $\mathcal{T}_3$ where the depth of the tree rooted at v_0 is 2. For this tree, player 1 is the only hub and we have $|\mathcal{S}^B(\mathcal{T}_2)| = 2^{n-1} - 1$. Therefore, for this tree of depth 2 the exponent is only reduced by one compared to $\mathcal{T}_1$, which was a line graph. In general for a shortest path tree, say for $\mathcal{T}_4$ in Figure 3.14, as long as there is a player with followers other than itself, such as player 4, there is an exponential number of inequalities that are not redundant imposed by this player.

On the other hand, in practice, this method can reduce the number of coalitional costs $c(S)$ that are needed to describe the core of the SPT games despite the fact that in general the exponential nature of the problem still persists. For instance, in Example

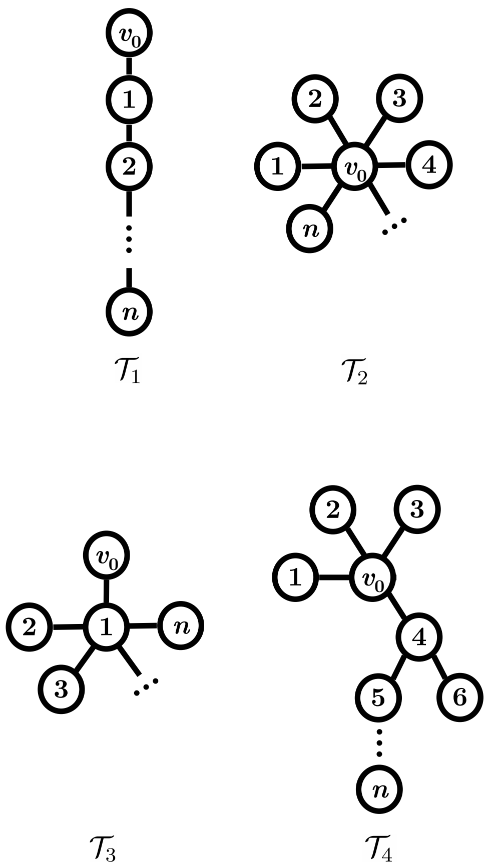


Figure 3.14: Different types of trees.

3.19, we have shown that only 30 of the 254 coalitions in $S \in 2^N \backslash \{\emptyset, N\}$ are in $\mathcal{S}^B(\mathcal{T})$, which is in fact a significant reduction.

We illustrate our computational results on the reduction of the core for the application of Wireless-Multihop Networks in the next chapter.

3.7 The Nucleolus of the Shortest Path Tree Games

In this section, we discuss some aspects of computing the nucleolus of an SPT game from a theoretical aspect.

The nucleolus of a cooperative cost game is defined as the unique imputation that lexicographically maximises the excesses of all coalitions. Due to the comparison of vectors of exponential size, algorithms that compute the nucleolus of a general cooperative game based on this definition would take an exponential time (Granot et al., 1998). There exist various examples of computing the nucleolus in polynomial time by exploiting the special structure of a particular class of cooperative games such as the nucleolus of fixed cost spanning forest games by Granot and Granot (1992), of tree games by Megiddo (1978) and by Granot et al. (1996), and of assignment games by Solymosi and Raghavan (1994). Elkind and Pasechnik (2009) propose a general framework for computing the nucleolus of weighted voting games, which can potentially be applied to a wider class of games. Granot et al. (1998) introduce the notion of a characterisation set for the nucleolus, which is based on identification of the minimum relevant information to characterise the nucleolus of a class of games. Our attempts to computing the nucleolus of the SPT games were inspired by this notion.

We use Kopelowitz's algorithm (Kopelowitz (1967) and Maschler et al. (1979)) to compute the nucleolus, which is discussed in detail in Chapter 4. The algorithm is based on solving linear programmes (LPs) sequentially until a unique solution, which is the nucleolus, is found. At each of the iterations of the algorithm, an LP that includes a constraint corresponding to a coalition $S \subseteq N$ containing the cost $c(S)$ of this coalition is solved. Therefore, for the preparation of the input of the algorithm for SPT games, a shortest path tree problem is solved for each $S \subseteq N$ using Dijkstra's algorithm in order to determine $c(S)$. Consequently, the time spent for data generation grows exponentially with the number of players. Firstly, from our result in Theorem 3.10, we know that to describe the core of the SPT games we only need to compute the costs of $|\mathcal{S}^B(\mathcal{T})|$ coalitions. Recall that in Section 3.6, we have argued that in general there are still an exponential number of coalitions whose costs need to be computed to describe the core of the SPT games. Based on the simulations we have performed, a smaller set of non-redundant inequalities does not appear to exist to describe the core of any SPT game.

We would like to note that our attempts for identifying a characterisation set of the nucleolus of SPT games did not prove successful. The structural similarities between the minimum cost spanning tree games and the SPT games have been mentioned previously, thus we would like to point out that Granot et al. (1998) state that the computation of the nucleolus of a general minimum cost spanning tree game would require exponential time. Moreover, Faigle et al. (1998) show that computing the nucleolus of general minimum cost spanning tree games is NP-hard and hence the nucleolus is unlikely to be computed efficiently.

Despite the difficulties of obtaining a theoretical reduction in the exponential nature of the computation of the nucleolus, in the next chapter we illustrate that using our result of the reduction in the description of the core coupled with a constraint generation approach made it possible to find the nucleolus for cases that were otherwise intractable for the application of Wireless-Multihop Networks.

3.8 Special Case: Triangle Inequality Holds

In this section, we discuss the special case of the triangle inequality holding for the SPT problem and its implications for the properties of the SPT games.

Firstly, consider the SPT problem in Figure 3.15. If the triangle inequality holds then we have $t_{v_01} < t_{v_02} + t_{12}$. In this case, the shortest path for player 1 goes through edge t_{v_01}. On the SPT of this graph players 1 and 2 will be directly connected to the source vertex.

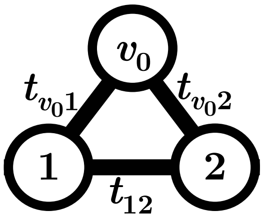


Figure 3.15: An SPT problem.

It is easy to observe that if we assume that the triangle inequality holds for the SPT problems, the optimal solution will always be a star graph where all players are hubs. Therefore, each branch will consist exactly of one player that is connected to the source vertex through its direct link. Firstly, note that $c(\{i\}) = t_{v_0i}$. Secondly, from Lemma 3.2, which states that in the core of the SPT games the total cost allocated the players in a branch equals exactly to the cost of this coalition, we have

$$x_i = t_{v_0i} \text{ for all } i \in N.$$

The implication of the above result is that in the case where the triangle inequality holds the core of the SPT game consists of exactly one point and this point is the tree solution. Since we know that if the core is nonempty then the nucleolus is always a member of the core, for this special case $\theta((N \cup \{v_0\}, E), t) = \eta(c)$ where $\eta(c)$ denotes the nucleolus of the SPT game (N, c).

Let us now consider the monotonicity and the submodularity of the SPT games when the triangle inequality holds. Firstly, observe that $c(S) = \sum_{i \in S} t_{v_0 i}$. That is, when the triangle inequality holds, the cost of a coalition is just the sum of the costs of the direct links of all its members. Since edge costs are assumed to be nonnegative, we have $c(S) \leq c(T)$ for all $S \subseteq T$. Therefore, for this special case SPT games satisfy monotonicity. Furthermore, we have $c(S \cup \{i\}) - c(S) = t_{v_0 i} = c(T \cup \{i\}) - c(T)$ for all $i \in N$ and for all $S \subset T \subseteq N \backslash \{i\}$ and hence for this special case SPT games also satisfy submodularity.

Finally, we mention that when the triangle inequality holds $\theta((N \cup \{v_0\}, E), t) = \phi(c)$ where $\phi(c)$ denotes the Shapley value of the game (N, c). This is because we know that for a submodular game the Shapley value is always in the core, and for this special case the core only consists of the tree solution.

Chapter 4

Computing the Core and the Nucleolus of the Shortest Path Tree Games in Wireless Multihop Networks

This chapter presents our computational results on the core and the nucleolus of the SPT games. The results in this chapter hold for any application of the SPT games but here we will demonstrate these results through the application of cost allocation in Wireless Multihop Networks (WMNs). We start, in Section 4.1, with a discussion on the definition and properties of WMNs. In Section 4.2, the results of our simulations on the reduction of the definition of the core of the SPT games are presented. These simulations are performed to observe the application of our theoretical result on the non-redundant coalitions in the definition of the core of the SPT games to the WMN example. Section 4.3 first proposes to compute the nucleolus of the SPT games for the WMN application using the linear programming based algorithm of Kopelowitz. Next, we incorporate our reduction in the description of the core result to this algorithm. Finally, we employ a constraint generation approach for the computation of the nucleolus in order to generate the cost of the non-redundant coalitions on the fly. We conclude this section by a comparison of the performance of these three approaches.

4.1 An Application: Cost Allocation in Wireless Multihop Networks

An application of the SPT games is the problem of cost allocation in Wireless Multihop Networks (WMNs). WMNs are cellular networks distributed over geographical areas called cells, each served by one fixed-location base station that provides connectivity to the internet. In WMNs, users can relay information for other users in order to reduce the total power used for the signal to reach the base station. Each wireless link has a cost associated with the power needed to transmit, and power is proportional to d^a where d is the distance between users and a is the path loss exponent (a small number in the range $2-4$). Thus, it is often cheaper in terms of power for a signal to hop several times before reaching the base station instead of direct transmission.

WMNs have two major benefits. Firstly, they help save energy due to the power savings, so they are considered to be a green technology. Moreover, for cases where there is no wired connection or there is limited wireless transmission range, WMNs make connectivity possible.

In WMNs, each base station has a capacity that depends on the number of users in a cell that the base station can provide connectivity for. In the simulations of this chapter, we consider up to 20 users in a cell. This is considered to be a realistic and reasonable number of users in a cell. For geographical areas with denser population of users such as urban areas, the cells are smaller compared to suburban/rural areas. Moreover, since the size of the cells are smaller in urban areas the distances between the base station and the users are also smaller. As a result if this, although in urban areas relaying signals saves energy for some users, generally the shortest path tree formed by the users tend to include more direct links. However, in suburban/rural areas the cells are larger. The WMN technology in such areas is not only crucial for energy savings but also provides connectivity for users who are otherwise outside the range of a base station. Therefore, in suburban/rural areas there is more relaying and shortest path trees use fewer direct links. For this reason, we assume a cell that is approximately $2km \times 2km$, which is representative of a suburban area, and perform our simulations for the reduction in the definition of the core and nucleolus for WMNs in suburban areas. Finally, we would like to highlight that for the WMNs the triangle inequality does not hold, therefore the cost allocation problem for the WMNs does not correspond to the (trivial) case as discussed in Section 3.8.

Formally, let $N = \{1, 2, \ldots, n\}$ be the set of users in the WMN. We consider the undirected graph $G = (N \cup \{v_0\}, E)$ where the source vertex v_0 denotes the base station and the set of edges $E = \{\{i, j\} : i, j \in N \cup \{v_0\} \text{ and } i \neq j\}$ represents the set of all possible transmissions. We associate a cost t_{ij} with each edge $\{i, j\}$, which is the power needed to transmit from user i to user j (and vice versa). The base station chooses the optimal solution, that is, a shortest path tree $\mathcal{T}$ of the graph G. Then, the total cost of the shortest path tree, $t(\mathcal{T})$, needs to be shared among the users. The cost sharing problem in WMNs can be formulated as an SPT game where the cost represents power used.

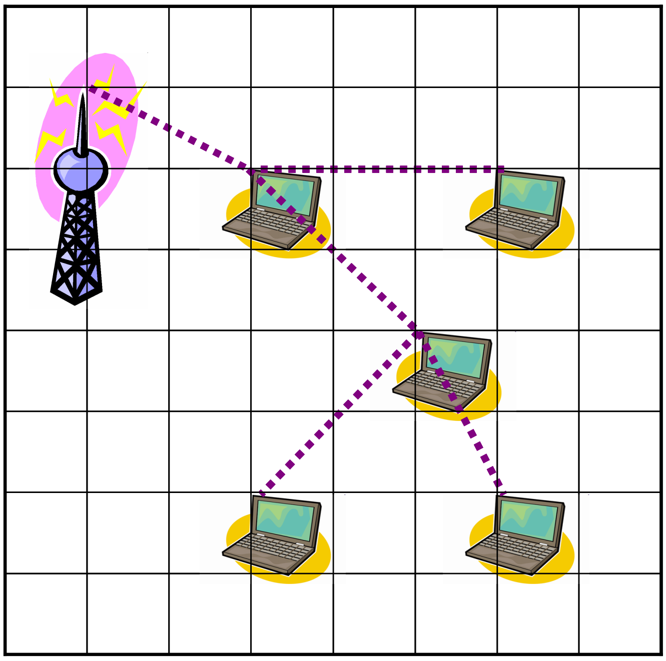


Figure 4.1: A Wireless Multihop Network

Let us assume that the optimal solution to a WMN problem in a cell with five users is the shortest path tree displayed in Figure 4.1. On this tree, all the users are connected to the base station. Now, the base station should allocate the cost of this shortest path tree in a “fair” and “mutually satisfactory” manner. For example, let us consider the tree solution θ such that each user is allocated its own connection cost. In this case, could it be considered “fair” for the hub user, that is, the user closest to the base station, to be allocated its own connection cost? The hub user helps reduce the connection cost of all the other four users in this example. The base station could consider allocating a lower cost to such users in order to motivate them to stay and thus, reduce the connection cost of other users.

4.2 Reduction in the Description of the Core of the Shortest Path Tree Games

In this section, we present the results of our simulations on the reduction of the core of the SPT games discussed in Section 3.6. Recall that in Theorem 3.10, we have presented a reduced description of the core of the SPT games. As a result, we have argued that in order to describe the core of the SPT games we only need to consider the costs of $|\mathcal{S}^B(\mathcal{T})|$ non-redundant coalitions.

To demonstrate the reduction in complexity, we generated random graphs G based on the application of WMNs where the costs are calculated as power needed for transmission. We consider a $2km \times 2km$ cell with n users for a number of different values of n. Table 4.1 shows the distribution of the proportion of non-redundant coalitions. These proportions are based on 100 scenarios for each value of n. As we can see from the table, there is a high probability that the number of non-redundant coalitions will be a small percentage $(0-5\%)$ of the total number of coalitions especially as n increases.

% of Non-redundant Coalitions:	0-5%	5-50%	50-100%
n=8	0.06	0.86	0.08
n=10	0.38	0.56	0.06
n=12	0.53	0.40	0.07
n=14	0.68	0.20	0.12
n=16	0.75	0.16	0.09
n=18	0.78	0.12	0.10
n=20	0.79	0.16	0.05

Table 4.1: Distributions of the percentages of non-redundant coalitions: each row shows the empirical probability distribution derived over 100 randomly generated graphs with n vertices.

Furthermore, given that the users are spread approximately uniformly within the $2km \times 2km$ area, the depth of the shortest path tree is likely to grow according to $O(log(n))$ where n is the number of users. Thus, the percentage of non-redundant coalitions is likely to decrease as n gets larger as shown in Table 4.1.

4.3 Computing the Nucleolus of the Shortest Path Tree Games

In this section, we compute the nucleolus of the SPTs using three different approaches. First, in Section 4.3.1, we introduce a linear programming based algorithm (Kopelowitz's algorithm) for the calculation of the nucleolus of the SPT games. Secondly, in Section 4.3.2, we modify this algorithm such that it only takes the non-redundant coalitions in the description of the core of the SPT games into consideration. Finally, in Section 4.3.3, we furthermore introduce a constraint generation approach to the algorithm of Section 4.3.2. In Section 4.3.4, we compare the performance of these three approaches to computing the nucleolus of the SPT games for the WMN application.

4.3.1 A Linear Programming Based Algorithm to Compute the Nucleolus

The nucleolus of a cooperative game (N, c) can be computed using Kopelowitz's algorithm (Kopelowitz (1967) and Maschler et al. (1979)), which is based on solving linear programmes (LPs) sequentially until a unique solution, which is the nucleolus, is found. In any one of the LPs, there are three types of coalitions: the ones initially set as equalities, the inequalities and the ones that have been fixed during the steps prior to solving this linear programme. Therefore, we identify an iteration of the algorithm with the LP that we need to solve at this iteration using three types of coalitions and denote it as $LP(\mathcal{Q}, \mathcal{I}, \Phi_k)$ where $\mathcal{Q}$ is the set of coalitions whose costs are initially set as equalities, $\mathcal{I}$ is the set coalitions that indicate the inequalities, and $\Phi_k = \mathcal{F}_1 \cup \mathcal{F}_2 \cup \cdots \cup \mathcal{F}_k$ is the collection of coalitions that indicate the inequalities fixed as equalities until the current iteration k such that $\mathcal{F}_k$ is the set of inequalities fixed as equalities at iteration $(k-1)$ and the algorithm starts with iteration $k = 0$.

We will now define iteration 0. We have $\mathcal{Q} = \{N\}$, $\mathcal{I}_0 = 2^N \backslash \{\emptyset, N\}$ and $\Phi_0 = \mathcal{F}_0 = \{\emptyset\}$. We solve

$$LP(\{N\}, 2^N \backslash \{\emptyset, N\}, \{\emptyset\}) \qquad \begin{aligned} &\max \epsilon \\ &\text{s.t.} \quad c(S) - \sum_{i \in S} x_i \geq \epsilon \qquad \text{for all } S \in 2^N \backslash \{\emptyset, N\} \\ &\qquad \sum_{i \in N} x_i = c(N) \; . \end{aligned}$$

We stop if $LP(\{N\}, 2^N \backslash \{\emptyset, N\}, \{\emptyset\})$ has a unique solution. Otherwise, let $y_1(S)$ denote the dual variable corresponding to the constraint for coalition S in the above LP. Let (ϵ_1^*, x^*) be an optimal solution to $LP(\{N\}, 2^N \backslash \{\emptyset, N\}, \{\emptyset\})$ and $y_1^*(S)$ be the corresponding dual solution. Let $\mathcal{F}_1 \subseteq \mathcal{I}_0$ be the set of coalitions that are binding at the optimum solution, so $\mathcal{F}_1 = \{S \in \mathcal{I}_0 | \quad y_1^*(S) > 0\}$. Let $\mathcal{I}_1 = \mathcal{I}_0 \backslash \mathcal{F}_1$ and $\Phi_1 = \mathcal{F}_1$. For iteration 1, we solve

$$
\begin{aligned}
LP(\{N\}, \mathcal{I}_1, \Phi_1) \qquad & \max \epsilon \\
& \text{s.t.} \quad c(S) - \sum_{i \in S} x_i \geq \epsilon \qquad \text{for all } S \in \mathcal{I}_1 \\
& \qquad\quad c(S) - \sum_{i \in S} x_i = \epsilon_1^* \quad \text{for all } S \in \mathcal{F}_1 \\
& \qquad\quad \sum_{i \in N} x_i = c(N) \; .
\end{aligned}
$$

We stop if $LP(\{N\}, \mathcal{I}_1, \Phi_1)$ has a unique solution, otherwise we continue in the fashion described as above.

Assume that we have $\mathcal{F}_k = \{S \in \mathcal{I}_{k-1} | \quad y_{k-1}^*(S) > 0\}$, $\mathcal{I}_k = \mathcal{I}_{k-1} \backslash \mathcal{F}_k$. For iteration k, we solve

$$
\begin{aligned}
LP(\{N\}, \mathcal{I}_k, \Phi_k) \qquad & \max \epsilon \\
& \text{s.t.} \quad c(S) - \sum_{i \in S} x_i \geq \epsilon \qquad \text{for all } S \in \mathcal{I}_k \\
& \qquad\quad c(S) - \sum_{i \in S} x_i = \epsilon_1^* \quad \text{for all } S \in \mathcal{F}_1 \\
& \qquad\quad c(S) - \sum_{i \in S} x_i = \epsilon_2^* \quad \text{for all } S \in \mathcal{F}_2 \\
& \qquad\qquad\qquad \vdots \\
& \qquad\quad c(S) - \sum_{i \in S} x_i = \epsilon_k^* \quad \text{for all } S \in \mathcal{F}_k \\
& \qquad\quad \sum_{i \in N} x_i = c(N) \; .
\end{aligned}
$$

which has a unique optimal solution (ϵ_{k+1}^*, x^*) and x^* is the nucleolus $\eta(c)$ of (N, c).

Note that in order to determine the uniqueness of the solution after any iteration of the algorithm, the Gaussian elimination method described in Fromen (1997) can be employed. We construct a 0-1 matrix M_{k+1} after the k^{th} iteration, which consists of rows m^S_{k+1} for each coalition $S \in \mathcal{Q} \cup \Phi_{k+1}$ such that

$$m^S_{k+1} = \begin{cases} 1 & \text{if } i \in S \\ 0 & \text{otherwise.} \end{cases}$$

There will be a unique solution x^* which gives us the nucleolus, only if $M_{k+1} \in \mathbb{R}^{(|\mathcal{Q}|+|\Phi_{k+1}|)\times|N|}$ has full rank, that is, rank$(M_{k+1}) = |N|$.

The above procedure to compute the nucleolus of $\eta(c)$ of (N, c) is summarised below.

Algorithm 1: An LP Based Algorithm to Compute the Nucleolus

Input: $c(S)$ for all $S \in 2^N \backslash \{\emptyset\}$ of (N, c).
begin
  Set k=0;
  Let $\mathcal{I}_0 = 2^N \backslash \{\emptyset, N\}$;
  Let $\Phi_0 = \{\emptyset\}$;
  Let M_0 be an empty matrix;
  while (rank$(M_k) < |N|$) do
    begin
      Solve $LP(\{N\}, \mathcal{I}_k, \Phi_k)$;
      Store $\eta(c)$=x^*;
      $k = k + 1$;
      Let $\mathcal{F}_k = \{S \in \mathcal{I}_{k-1} | \quad y^*_{k-1}(S) > 0\}$;
      Let $\Phi_k = \Phi_{k-1} \cup \mathcal{F}_k$;
      Let $\mathcal{I}_k = \mathcal{I}_{k-1} \backslash \mathcal{F}_k$;
      Construct M_k;
    end;
end;
Output: $\eta(c)$ is the nucleolus of (N, c).

4.3.2 A Linear Programming Based Algorithm to Compute the Nucleolus Using Non-redundant Coalitions

In this section, to obtain a better performance for the LP based nucleolus algorithm, we present a modified version of the aforementioned algorithm where we exploit the reduction in the description of the core of the SPT games as discussed in Section 3.6.

Recall that from Theorem 3.10, the reduced description of the core is

$$Core(c) = \left\{ x \in \mathbb{R}^N | \sum_{i \in S} x_i = c(S) \; \forall S \in \mathcal{B}(\mathcal{T}), \sum_{i \in S} x_i \leq c(S) \; \forall N \backslash S \in \mathcal{S}^B(\mathcal{T}) \backslash \mathcal{B}(\mathcal{T}) \right\}.$$

Thus, in order to prepare the input of the modified LP based algorithm to compute the nucleolus, we only need to compute the $c(S)$ values for $S \in \mathcal{B}(\mathcal{T})$ and for $N \backslash S \in \mathcal{S}^B(\mathcal{T}) \backslash \mathcal{B}(\mathcal{T})$. In return, instead of running $2^{|N|} - 1$ Dijkstra's algorithms to prepare the input, we only need to run $|\mathcal{S}^B(\mathcal{T})|$ Dijkstra's algorithms, which is shown to be significantly less than $2^{|N|} - 1$ in Section 3.6 except for the case where the shortest path tree is a line.

Based on Theorem 3.10, we can redefine the initial linear programme as follows and denote it by $\overline{LP}(\mathcal{B}(\mathcal{T}), \mathcal{S}^B(\mathcal{T}) \backslash \mathcal{B}(\mathcal{T}), \{\emptyset\})$. Note that the difference between the notation of LP and $\overline{LP}$ is due to the fact that $\overline{LP}$ considers coalitions $N \backslash S$ to indicate the inequalities $c(S) - \sum_{i \in S} x_i \geq \epsilon$ to be considered whereas LP considers coalitions S. We let $\mathcal{Q} = \mathcal{B}(\mathcal{T}), \mathcal{I}_0 = \mathcal{S}^B(\mathcal{T}) \backslash \mathcal{B}(\mathcal{T})$ and $\Phi_0 = \{\emptyset\}$. We solve

$$\begin{aligned} \overline{LP}(\mathcal{B}(\mathcal{T}), \mathcal{S}^B(\mathcal{T}) \backslash \mathcal{B}(\mathcal{T}), \{\emptyset\}) \quad & \max \epsilon \\ \text{s.t.} \quad & c(S) - \sum_{i \in S} x_i \geq \epsilon \quad \text{for all } N \backslash S \in \mathcal{S}^B(\mathcal{T}) \backslash \mathcal{B}(\mathcal{T}) \\ & \sum_{i \in S} x_i = c(S) \qquad \text{for all } S \in \mathcal{B}(\mathcal{T}) \; . \end{aligned}$$

The rest of the procedure will be the same as Algorithm 1 where the stopping condition is the rank of the matrix, whose rows represent the coalitions whose costs have been fixed so far, being equal to $|N|$. We summarise this approach in Algorithm 2.

Algorithm 2: An LP Based Algorithm to Compute the Nucleolus Using Non-redundant Coalitions

---

Input: $c(S)$ for all $S \in \mathcal{B}(\mathcal{T})$ and
for all $N \backslash S \in \mathcal{S}^B(\mathcal{T}) \backslash \mathcal{B}(\mathcal{T})$ of (N, c).

begin
  Set k=0;
  Let $\mathcal{I}_0 = \mathcal{S}^B(\mathcal{T}) \backslash \mathcal{B}(\mathcal{T})$;
  Let $\Phi_0 = \{\emptyset\}$;
  Let M_0 be an empty matrix;
  while (rank$(M_k) < |N|$) do
    begin
      Solve $\overline{LP}(\mathcal{B}(\mathcal{T}), \mathcal{I}_k, \Phi_k)$;
      Store $\eta(c)$=x^*;
      $k = k + 1$;
      Let $\mathcal{F}_k = \{S \in \mathcal{I}_{k-1} | \quad y^*_{k-1}(S) > 0\}$;
      Let $\Phi_k = \Phi_{k-1} \cup \mathcal{F}_k$;
      Let $\mathcal{I}_k = \mathcal{I}_{k-1} \backslash \mathcal{F}_k$;
      Construct M_k;
    end;
end;

Output: $\eta(c)$ is the nucleolus of (N, c).

4.3.3 A Linear Programming Based Algorithm to Compute the Nucleolus Using Constraint Generation and Non-redundant Coalitions

Based on the distribution of the percentage of the non-redundant coalitions illustrated in Table 4.1 in Section 4.2, we have shown that the non-redundant coalitions constitute only around 0-5% of all the coalitions for most of the cases. Nevertheless, as pointed out previously in Section 3.6, theoretically there are still an exponential number of non-redundant coalitions and data generation would become intractable as n grows. Moreover, the size of each linear programme solved becomes very large. Due to these two issues, we will now employ the constraint generation procedure introduced by Gilmore and Gomory. This procedure has been used for Linear Production Games in Hallefjord et al. (1995) where the violated coalitions are generated by solving mixed integer programmes (MIPs). For our games, this implies that the cost of a coalition will be generated by solving an MIP only if it is needed to find the nucleolus.

We will start with a summary of the constraint generation approach within the context of our problem. In order to find the nucleolus of the SPT games we need to solve $\overline{LP}(\mathcal{B}(\mathcal{T}), \mathcal{S}^B(\mathcal{T})\backslash\mathcal{B}(\mathcal{T}), \{\emptyset\})$ lexicographically. This problem has only n variables, therefore not all the inequality constraints will be required to obtain an optimal solution. Since we do not have prior knowledge of the inequality constraints that are actually needed, we first run this linear programme with a small subset of the inequalities. In this case, some of the excluded coalitions will not be satisfied with the current allocation. To find the least satisfied one, we solve an MIP. We add the coalition generated by the MIP to the small subset of inequalities initially included and rerun the linear programme. Then, we solve the MIP again to find the least satisfied coalition with respect to the new allocation. The procedure continues until the MIP cannot find the least satisfied coalition, that is, when there is no coalition that is not satisfied with respect to the current allocation. This means that we have all the constraints we needed to solve the initial linear programme $\overline{LP}(\mathcal{B}(\mathcal{T}), \mathcal{S}^B(\mathcal{T})\backslash\mathcal{B}(\mathcal{T}), \{\emptyset\})$ to optimality. Then, we find the coalitions to be fixed and proceed to the next linear programme of the nucleolus algorithm. For every iteration of the nucleolus algorithm, we repeat the above procedure of running an MIP and adding a constraint until the compact version gives the same solution with the corresponding linear programme of the nucleolus algorithm. As a result, we only calculate $c(S)$ values for coalitions that are needed to find the nucleolus, and simultaneously reduce the size of the linear programmes solved to find the nucleolus.

The constraint generation approach as used by the authors in (Hallefjord et al., 1995) would generate any coalition if adapted to the SPT problem formulation. However, since we have already found a reduced description of the core using inequalities $\sum_{i \in S} x_i \leq c(S)$ for all $N \backslash S \in \mathcal{S}^B(\mathcal{T}) \backslash \mathcal{B}(\mathcal{T})$, we modify the MIPs to only generate the least satisfied coalition S such that its complement $N \backslash S$ is in $\mathcal{S}^B(\mathcal{T}) \backslash \mathcal{B}(\mathcal{T})$. Although this makes the MIPs harder to solve, we will use the modified version in light of the very poor performance of the simulations where we allowed the MIPs to generate coalitions that correspond to redundant core inequalities.

We proceed to the formal description of the procedure of finding the nucleolus of the SPT games iteratively using a constraint generation approach. Let $\overline{LP}(\mathcal{B}(\mathcal{T}), \mathcal{V}, \{\emptyset\})$ denote the relaxed version of $\overline{LP}(\mathcal{B}(\mathcal{T}), \mathcal{S}^B(\mathcal{T}) \backslash \mathcal{B}(\mathcal{T}), \{\emptyset\})$ including only the inequalities for $N \backslash S \in \mathcal{V}$ where $\mathcal{V} \subset \mathcal{S}^B(\mathcal{T}) \backslash \mathcal{B}(\mathcal{T})$. We have

$$
\begin{aligned}
\overline{LP}(\mathcal{B}(\mathcal{T}), \mathcal{V}, \{\emptyset\}) \qquad & \max \epsilon \\
& \text{s.t.} \quad c(S) - x(S) \geq \epsilon \qquad \text{for all } N \backslash S \in \mathcal{V} \\
& \qquad\quad \sum_{i \in S} x_i = c(S) \qquad \text{for all } S \in \mathcal{B}(\mathcal{T}) \ .
\end{aligned}
$$

where $\mathcal{V}$ denotes the set of coalitions initially chosen to be included. This set can be the empty set, the singleton coalitions or the set of singleton complements (Hallefjord et al., 1995). For the SPT games, we can also take $\mathcal{V}$ to be the set of facets, which we have identified in Theorem 3.9. Assume that solving the above LP gives us the solution (ϵ^*, x^*).

We will now present the formulation of the MIP to generate the least satisfied coalition S such that $N \backslash S \in \mathcal{S}^B(\mathcal{T})$ with respect to the given solution. Recall that $V(i, \mathcal{T})$ denotes the set of vertices on the shortest path of i on $\mathcal{T}$, excluding v_0 but including i and $V^S(i, \mathcal{T}) = \{k : k \in S \cap V(i, \mathcal{T})\}$. A player i in S is an S-hub if $V^S(i, \mathcal{T}) = \{i\}$. Thus, S-hubs are players who do not have any other player in S on their shortest path on $\mathcal{T}$.

Below are the parameters and variables of the MIP formulation for the constraint generation.

Parameters

N : Set of vertices

n : Number of vertices

t_{ij} : Cost of edge {i,j}

ϵ^* : Optimal value of happiness given by the most recent nucleolus iteration

x_i^* : Allocation given by the nucleolus iteration for vertex $i, \forall i \in N$

$$SP_{ij} := \begin{cases} 1, & \text{if } j \in V(i,\mathcal{T}) \\ 0, & \text{otherwise} \end{cases}$$

(Note: $SP_{ii} = 1$)

Variables

$y_{ij} \geq 0$: Number of shortest paths going through edge {i,j}

$$s_i \text{ binary} := \begin{cases} 1, & \text{if } i \in S \\ 0, & \text{otherwise} \end{cases}$$

$$z_{ij} \text{ binary} := \begin{cases} 1, & \text{if } i,j \in N \backslash S \\ 0, & \text{otherwise} \end{cases}$$

$$u_j := \sum_{i \in N} SP_{ij} \cdot z_{ij} \quad \forall j \in N$$

$$w_i \text{ binary} := \begin{cases} 1, & \text{if } i \in N \backslash S \\ 0, & \text{otherwise} \end{cases}$$

$$b_j \text{ binary} := \begin{cases} 1, & \text{if } u_j = \sum_{k \in N} w_k \\ 0, & \text{otherwise} \end{cases}$$

or

$$b_j \text{ binary} := \begin{cases} 1, & \text{if vertex j is the } N\backslash S\text{-hub} \\ 0, & \text{otherwise} \end{cases}$$

The following formulation finds the most violated inequality $\sum_{i \in S} x_i \leq c(S)$ such that $N \backslash S$ is in $\mathcal{S}^B(\mathcal{T}) \backslash \mathcal{B}(\mathcal{T})$. The variables w_i ensure that the complement of the coalition generated will be in $\mathcal{S}^B(\mathcal{T})\backslash\mathcal{B}(\mathcal{T})$.

$$\text{(MIP)} \quad \min \sum_{i\in\{v_0\}\cup N} \sum_{j\in\{v_0\}\cup N} t_{ij} \cdot y_{ij} - \sum_{i\in N} x_i^* \cdot s_i - \epsilon^* \tag{4.1}$$

$$\text{s.t.} \quad -\sum_{i\in\{v_0\}\cup N} y_{iv_0} + \sum_{k\in\{v_0\}\cup N} y_{v_0k} = \sum_{j\in N} s_j \tag{4.2}$$

$$\sum_{i\in\{v_0\}\cup N} y_{ij} - \sum_{k\in\{v_0\}\cup N} y_{jk} = s_j \quad \forall j \in N \tag{4.3}$$

$$y_{ij} \leq n \cdot s_i \quad \forall i \in \{v_0\} \cup N, \forall j \in \{v_0\} \cup N \tag{4.4}$$

$$y_{ij} \leq n \cdot s_j \quad \forall i \in \{v_0\} \cup N, \forall j \in \{v_0\} \cup N \tag{4.5}$$

$$s_{v_0} = 1 \tag{4.6}$$

$$\sum_{j\in N} s_j \geq 1 \tag{4.7}$$

$$\sum_{j\in N} s_j \leq n - 1 \tag{4.8}$$

$$w_i = 1 - s_i \quad \forall i \in N \tag{4.9}$$

$$z_{ij} \leq w_i \quad \forall i \in N, j \in N \tag{4.10}$$

$$z_{ij} \leq w_j \quad \forall i \in N, j \in N \tag{4.11}$$

$$z_{ij} \geq w_i + w_j - 1 \quad \forall i \in N, j \in N \tag{4.12}$$

$$u_j = \sum_{i\in N} SP_{ij} \cdot z_{ij} \quad \forall j \in N \tag{4.13}$$

$$\sum_{i\in N} b_i = 1 \tag{4.14}$$

$$u_i - \sum_{k\in N} w_k \leq n \cdot (1 - b_i) \quad \forall i \in N \tag{4.15}$$

$$u_i - \sum_{k\in N} w_k \geq n \cdot (b_i - 1) \quad \forall i \in N \tag{4.16}$$

The objective function (4.1) minimises satisfaction by minimising the difference between the actual cost of the coalition and the current allocation. The constraints (4.2)-(4.5) are derived from the shortest path tree problem formulation. (4.2) makes sure that the number of shortest paths going out of the source vertex equals to the number of vertices that will be in S. The set of equalities in (4.3) is to guarantee that exactly one shortest path reaches a vertex provided that it will be in the coalition. The equalities in (4.4) and (4.5) ensure that the corresponding shortest path variables are only positive if both the source and the destination are in the coalition. (4.6) forces the source vertex to always be considered along with every coalition since we are trying to find the shortest path to the source vertex. (4.7) and (4.8) limit the number of players in a coalition to be between 1 and $(n-1)$ since we do not wish to generate the empty set and the grand coalition. We have (4.9) to find make sure that S will be the complement of a coalition in $\mathcal{S}^B(\mathcal{T})$. In other words, (4.9) provides a link between the shortest path formulation and the rest of the formulation where we guarantee that the MIP will only produce coalitions whose complements are in $\mathcal{S}^B(\mathcal{T})$. Therefore, we will only be adding non-redundant inequalities to the LPs.

By definition, a coalition in $\mathcal{S}^B(\mathcal{T})$ satisfies the following where w variables represent inclusion in such a coalition

$$\max_{j \in N} \sum_{i \in N} SP_{ij} \cdot z_{ij} = \sum_{k \in N} w_k. \tag{4.17}$$

since there exists a vertex that is on the shortest path of every vertex in the coalition.

The constraints (4.10)-(4.16) are the linear representation of the expression in (4.17). The z_{ij} are the binary variables that indicate both vertices i and j are in a coalition that belongs to $\mathcal{S}^B(\mathcal{T})$. Thus, using constraints (4.10)-(4.12), we link the w_i variables to the z_{ij} variables. Constraints (4.13) define variables u. The binary variable b_j takes value 1 if vertex j is $N \backslash S$-hub. Since there is only one such hub, (4.14) should hold. If vertex j is the $N \backslash S$-hub, that is, $b_j = 1$ or equivalently (4.17) holds, then (4.15) and (4.16) force $u_j = \sum_{i \in N} SP_{ij} \cdot z_{ij} = \sum_{k \in N} w_k$. Otherwise, they only provide some tight bounds on u. Note that instead of using $(n-1)$, we can use the cardinality of the branch that satisfies max $|B_h(\mathcal{T})|$ for all $h \in H(\mathcal{T})$ and obtain a tighter bound on (4.15) and (4.16).

Solving the MIP gives us the most violated coalition denoted by $S^* = \{i | s_i^* = 1\}$ and the cost of this constraint $c(S^*) = \sum_{i \in \{v_0\} \cup N} \sum_{j \in \{v_0\} \cup N} t_{ij} \cdot y_{ij}$. We add the constraint $x(S^*) - c(S^*) \geq \epsilon$ to the initial LP, that is, we expand $\mathcal{V}$ with $N \backslash S^*$, and solve $\overline{LP}(\mathcal{B}(\mathcal{T}), \mathcal{V}, \{\emptyset\})$ to obtain a new solution. The loop that consists of solving the MIP and resolving the linear programme with the added constraints will continue until

$$\min \sum_{i \in \{v_0\} \cup N} \sum_{j \in \{v_0\} \cup N} t_{ij} \cdot y_{ij} - \sum_{i \in N} x_i^* \cdot s_i - \epsilon^* \geq 0.$$

If the above condition holds, then there is no coalition that is dissatisfied and therefore all the constraints of $\overline{LP}(\mathcal{B}(\mathcal{T}), \mathcal{S}^B(\mathcal{T}), \{\emptyset\})$ are satisfied. Thus, $\overline{LP}(\mathcal{B}(\mathcal{T}), \mathcal{V}, \{\emptyset\})$ will give us the optimal solution of $\overline{LP}(\mathcal{B}(\mathcal{T}), \mathcal{S}^B(\mathcal{T}), \{\emptyset\})$ with a smaller number of inequalities since $\mathcal{V} \subset \mathcal{S}^B(\mathcal{T})$. Now, we can fix the cost of the coalitions that are binding at this solution and repeat the constraint generation procedure for the next iteration of the nucleolus iteration.

It is important to note here that the MIP will not generate the same coalition at a certain step of the nucleolus iteration. However, the problem changes as we move to the next iteration. Therefore, a cut is added to the MIP to make sure a coalition will not be regenerated (Hallefjord et al., 1995). Let us define such a cut as

$$Cut(S^*) := \sum_{i|s_i^*=0} s_i + \sum_{i|s_i^*=1} (1 - s_i) \geq 1.$$

Note that for the SPT games, we start with cutting the branches and the coalitions we initially choose to include in $\mathcal{V}$ since they will already be considered in the initial linear programme $\overline{LP}(\mathcal{B}(\mathcal{T}), \mathcal{V}, \{\emptyset\})$.

All the aforementioned steps are integrated into Algorithm 3.

Algorithm 3: An LP Based Algorithm to Compute the Nucleolus Using Constraint Generation and Non-redundant Coalitions

Input: $c(S)$ for all $S \in \mathcal{B}(\mathcal{T})$ and
for all $N \backslash S \in \mathcal{V}$ of (N, c).

```
begin
    Set k=0;
    Add Cut(S) for all S ∈ B(T) ∪ V to MIP;
    Let Φ_0 = {∅};
    Let M_0 be an empty matrix;
    while (rank(M_k) < n) do
        begin
            Set MinSatisfaction=-100;
            while MinSatisfaction < 0
                begin
                    Solve LP‾(B(T), V, Φ_k);
                    Solve MIP;
                    Let MinSatisfaction be the optimal objective function value of the MIP;
                    if MinSatisfaction ≥ 0
                      begin
                          Store η(c)=x*;
                          k = k + 1;
                          F_k = {S ∈ V|   y*_k(S) > 0};
                          Φ_k = Φ_{k-1} ∪ F_k;
                          V = V\F_k;
                          Construct M_k;
                      end;
                    else
                      begin
                          Add Cut(S*) to MIP;
                          V = V ∪ {N\S*};
                      end;
                end;
        end;
end;
```

Output: $\eta(c)$ is the nucleolus of (N, c).

4.3.4 Comparison of the Performance of the Nucleolus Algorithms

This section discusses the computational performance of the algorithms to compute the nucleolus of an SPT game. All our simulations in this section assume a WMN problem with $2km \times 2km$ cells and applies the algorithms introduced in the previous sections to compute the nucleolus.

We first present results comparing the running times of Algorithm 1 and Algorithm 2. The results are shown in Table 4.2.

% of Non-redundant Coalitions:		**0-25%**	**25-50%**	**50-100%**
% Reduction in Comp. Time:	**n=8**	81	61	29
	n=10	87	65	38
	n=12	90	78	47
	n=14	97	83	52
% Reduction in Nucleolus Iter.:	**n=8**	81	57	16
	n=10	84	54	24
	n=12	85	64	35
	n=14	87	62	32

Table 4.2: The table entries are average % reductions in time and iterations when using the nucleolus algorithm with non-redundant coalitions (Algorithm 2) as opposed to Algorithm 1. This is performed for 3 types of graphs depending on the % of non-redundant coalitions.

As can be seen from the table the reduction is significant both in computational times but also in the number of iterations (number of linear programmes solved) that the algorithm performs. The reduction tends to increase as n increases as the percentage of non-redundant coalitions decreases with n.

Since we can see above that the nucleolus algorithm that only uses non-redundant coalitions is faster, we will use Algorithm 2 as a basis of comparison with the iterative nucleolus algorithm utilising the constraint generation approach. Thus, next we compare the performance of Algorithm 2 and Algorithm 3. As mentioned earlier, with the constraint generation approach, the cost of a coalition is generated upon solving the MIP only if it is needed to find the nucleolus whereas without constraint generation, coalitional costs for all the non-redundant coalitions must be generated using Dijkstra's algorithm before the procedure starts. Furthermore, if we use the constraint generation approach, the linear programmes that are solved will only have the inequalities that are required and therefore will be much smaller in size. On the other hand, in general MIPs are harder to solve as compared to linear programmes. Thus, there is a trade-off in terms of the running times of the two algorithms, which find the nucleolus with and without using constraint generation. Note that both algorithms make use of our results on non-redundant coalitions. The results are illustrated in Table 4.3.

Tree	**% of all coalitions that are non-redundant**	**% of non-redundant coalitions that are needed**	**Algorithm 2: Total time (in secs) without constraint generation [of which data generation]**	**Algorithm 3: Total time (in secs) with constraint generation**
1	0.20	9.45	562 [488]	607
2	0.11	15.06	310 [292]	365
3	0.06	24.26	202 [173]	378
4	0.13	10.33	365 [340]	326
5	0.08	9.86	232 [224]	177
6	0.01	75.00	59 [49]	145
7	0.10	16.28	359 [313]	423
8	9.43	0.34	42295 [13751]	23258

Table 4.3: Comparison of running times of Algorithm 2 and Algorithm 3 for finding the nucleolus of SPT games ($n = 20$), without and with constraint generation respectively.

Each row of Table 4.3 corresponds to a simulation that is based on generating 20 random vertices on a cell of size $2km \times 2km$ and calculating the shortest path tree. For every tree, both Algorithm 2 and Algorithm 3 are performed to compare the running times. The second column of the table shows the percentage of all coalitions, which are non-redundant for each tree. In line with our previous results, the number of non-redundant coalitions constitute a small portion of the $1,048,575$ $(2^{20} - 1)$ coalitions. The third column shows the percentage of non-redundant coalitions that are actually needed to identify the nucleolus. The actual number of non-redundant coalitions needed is an indication of the number of times that the MIP was solved, excluding the initial cuts added. In order to exemplify the aforementioned trade-off concerning the running time of the two algorithms, we consider trees 6 and 8. Tree 6 has a very low number of non-redundant coalitions. For this tree, Algorithm 2, which does not use constraint generation, takes significantly less time since a low number of LPs are solved for data preparation and at each iteration of the nucleolus algorithm the LP solved has a low number of constraints. On the other hand, the constraint generation approach uses MIPs to generate coalitional costs thus Algorithm 3 takes much longer than Algorithm 2 to compute the nucleolus. Now, let us consider tree 8. Tree 8 has a very high number of non-redundant coalitions. Therefore, firstly the data generation phase of Algorithm 2 takes a long time generating costs of all non-redundant coalitions. Furthermore, at each iteration of the nucleolus algorithm the LP solved has a large number of constraints. For tree 8, Algorithm 3, which uses a constraint generation approach, performs significantly better than Algorithm 2. As mentioned earlier, for the constraint generation approach the coalitional costs are generated on the fly and at each iteration of the nucleolus algorithm the LP solved only include the coalitions required to find the nucleolus.

4.4 Concluding Remarks on Computational Results

In Section 3.6, we have presented our result on the reduced description of the core of the SPT games. We have furthermore shown that theoretically there are an exponential number of coalitions whose costs have to be calculated to identify the core of an SPT game in general. This chapter has aimed at assessing the consequences of our reduction result for the application of cost allocation in WMNs. We have demonstrated that we can achieve significant reductions both in the number of coalitional costs needed to be computed to describe the core as well as the time it takes to compute the nucleolus with a realistic number of users (up to 20). In fact, the computation of the nucleolus of the examples with 20 users would have been intractable with the basic LP based algorithm due to the requirement of calculating the costs of $1,048,575$ $(2^{20} - 1)$ coalitions.

For the computation of the nucleolus, we first compared the performance of the basic LP based algorithm (Algorithm 1) that computes the cost of all coalitions to the performance of the LP based algorithm that incorporates our result on the non-redundant inequalities of the core of the SPT games (Algorithm 2). We showed that there are considerable savings in terms of time and number of iterations required to find the nucleolus. Since Algorithm 2 performed better, we modified it further and incorporated a constraint generation approach to the computation of the nucleolus (Algorithm 3). When we compared the performance of Algorithm 2 to the performance of Algorithm 3, we observed that when there are lower number of non-redundant coalitions Algorithm 2 performed better whereas when there are higher number of non-redundant coalitions Algorithm 3 performed better.

We would like to finally highlight that the constraint generation approach, as employed in Hallefjord et al. (1995), is designed to generate the most violated coalition. Our constraint generation approach, on the other hand, specifically generates the most violated non-redundant coalition. For this reason, we had to define a special MIP. This MIP is harder to solve than the MIP that generates any coalition. The MIPs that generate any violated coalition run faster but we need to run too many of such MIPs unnecessarily since they also generate redundant coalitions. The special MIPs that we have developed are harder to solve but they only generate non-redundant coalitions. Despite the fact that this situation might sound like a trade-off, we know that the MIPs that only generate non-redundant coalitions consistently perform significantly better than the MIPs that generate any violated coalition based on our simulations.

Chapter 5

Weighted Minimum Colouring Games

The weighted minimum colouring problem is a combinatorial optimisation problem where there is a positive integer weight associated with each vertex of a graph representing the number of colours required to colour this vertex and the objective is to find the minimum number of colours k such that adjacent vertices are coloured with disjoint sets of colours where k is referred to as the weighted chromatic number of the graph. An application of this problem is the channel assignment in cellular telephone networks (McDiarmid and Reed, 2000). The problem is to assign sets of frequency bands to transmitters, each of which demands a different number of bands, and if unacceptable interference might occur between two transmitters, they should be assigned disjoint sets of bands. If a conflict graph is constructed such that each transmitter is represented by a vertex, the number of frequency bands required by a transmitter is represented by the positive integer weight of the corresponding vertex and the interference relation between two transmitters is represented by an edge between the corresponding vertices, then the minimum number of frequency bands needed is the weighted chromatic number of this graph. Consider a scenario where a number of mobile network operators are to provide cell phone service to a geographical area. Assume that all the frequency bands have the same cost and that the transmitters are owned by different operators. In order to provide the cell phone service with the minimum number of frequency bands, the operators should cooperate with each other. In this chapter, we tackle the allocation of the total cost of the minimum number of frequency bands amongst the operators involved using cooperative game theory. We define a new class of cooperative games called weighted minimum colouring (WMC) games where the cost of a subset of players is equal to the weighted chromatic number of the conflict subgraph induced by this subset.

A special case of the weighted colouring problem is when all the vertex weights are equal to 1. This problem is called a minimum colouring problem. The objective is to find the minimum number of colours k such that adjacent vertices are not assigned the same colour and k is referred to as the chromatic number of the graph. Therefore, the minimum colouring games defined by Deng et al. (1999) can be considered an instance of the WMC games. The cost of a subset of players in a minimum colouring game is equal to the chromatic number of the conflict subgraph induced by this subset. The class of minimum colouring games as well as the WMC games belong to the more general class of cooperative games arising from combinatorial optimisation problems. In general, the core of a minimum colouring game can be empty. Nonetheless, Deng et al. (2000) show that a minimum colouring game is totally balanced if and only if the underlying graph is perfect. A graph is perfect if for all its subgraphs the chromatic number is equal to the clique number. Furthermore, Okamoto (2003) characterises the submodularity of the minimum colouring games by showing that this property is satisfied if and only if the underlying graph is complete r-partite. A graph is complete r-partite if its vertices can be partitioned into r nonempty partition classes, and two vertices are adjacent if and only if they belong to different partition classes. More recently, Hamers et al. (2011) show that a minimum colouring game allows a population monotonic allocation scheme if and only if the underlying graph is $(2K_2, P_4)$-free. A $(2K_2, P_4)$-free graph is a graph that does not have a subgraph isomorphic to the union of two complete graphs of size 2 or to a line graph of size 4.

In this chapter, we characterise total balancedness and submodularity of the WMC games using the properties of the underlying graph. We show that a graph G induces a totally balanced WMC game for all positive integer weight vectors if and only if it is perfect and that any graph G induces a totally balanced WMC game for at least one positive integer weight vector. Furthermore, we show that a graph G induces a submodular WMC game for all positive integer weight vectors if and only if it is complete r-partite and that a graph G induces a submodular WMC game for at least one positive integer weight vector if and only if it is $(2K_2, P_4)$-free. These graph classes will be defined formally in relevant sections.

This chapter is organised as follows. We start with some graph theoretical definitions and notation in Section 5.1, which will be used throughout this chapter. In Section 5.2, we first define and illustrate the minimum colouring games. Furthermore, we review the existing research on the properties of this class of games. We furthermore formally introduce the WMC games. In Sections 5.3 and 5.4, we characterise total balancedness and submodularity of the WMC games, respectively.

Related Work

Our approach to the characterisation of total balancedness and submodularity of WMC games is in the same spirit with the characterisation of balancedness, total balancedness and submodularity of Chinese postman (CP) and travelling salesman (TS) games by Granot and Hamers (2004). In this paper, the authors define a graph to be globally (respectively, locally) CP balanced (respectively, totally balanced and submodular) if for all vertices (respectively, at least one vertex) and any non-negative weight vector defined on the edges, the corresponding CP game is balanced (respectively, totally balanced and submodular) and study the equivalence between globally and locally CP balanced (respectively, totally balanced and submodular) graphs. Similar results are obtained for the TS case. Moreover, from the existing line of research on characterising game theoretical properties by the properties of the underlying graph, we mention the characterisation of the balancedness (respectively, total balancedness and the submodularity) of CP games by Granot et al. (1999), the characterisation of the submodularity of the Steiner TS games on undirected graphs by Herer and Penn (1995) and on directed graphs by Granot et al. (2000) and of highway games by Çiftçi et al. (2010).

5.1 Preliminaries on Graph Theory

In this section, we present a number of graph theoretical definitions and notation.

Let $G = (N, E)$ be an undirected graph with finite vertex set $N = \{1, 2, \ldots, n\}$ and edge set $E \subseteq \{\{i, j\} : i, j \in N, i \neq j\}$ where each edge represents a connection between an unordered pair of vertices of G. The graph $G^S = (S, E^S)$ is the *subgraph* of G induced by a subset $S \subseteq N$ of its vertices where $E^S = \{\{i, j\} \in E : i, j \in S\}$. A graph $G = (N, E)$ is *isomorphic* to $G' = (N', E')$ if there exists a bijection $v : N \to N'$ such that $\{v(i), v(j)\} \in E'$ if and only if $\{i, j\} \in E$. The *complement* of a graph G is the graph $\bar{G} = (N, \bar{E})$ where $\bar{E} = \{\{i, j\} : i, j \in N, i \neq j, \{i, j\} \notin E\}$, that is, two vertices in $\bar{G}$ are adjacent if and only if they are not adjacent in G. A graph in which there exists an edge between each pair of distinct vertices is a *complete* graph. A *clique* in a graph G is a subset $S \subseteq N$ of its vertices such that G^S is complete. A clique is *maximum* if there are no cliques containing more elements and it is *maximal* if it is not contained within a clique with more elements. Note that maximum cliques are always maximal. The *clique number* of G, denoted by $\omega(G)$, is the number of vertices in a maximum clique in G. Let $w \in \mathbb{Z}_+^N$ be a positive integer weight vector where w_i is the weight associated with vertex $i \in N$. For a subset $S \subseteq N$ of vertices, the weight of S is defined as the sum of the weights of its elements, that is, $\sum_{i\in S} w_i$. We define a *maximum weighted clique* in G with respect to w as a clique $C \subseteq N$ with maximum weight. The corresponding weight is called the *weighted clique number* of G with respect to w and denoted by $\omega_w(G)$. Note that maximum weighted cliques are always maximal. Furthermore, note that a maximum clique in G is not necessarily a maximum weighted clique in G as we illustrate in the next example.

Example 5.1. Consider the graph G and the weight vector w displayed in Figure 5.1. The maximum clique in G is $\{1, 2, 3\}$ and $\omega(G) = 3$. The maximum weighted clique in G with respect to w is $\{3, 4\}$ and $\omega_w(G) = 8$. ◇

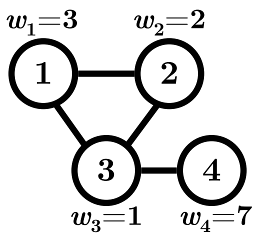


Figure 5.1: Graph G and weight vector w.

A *proper k-colouring* of G is a map $g : N \to \{1, 2, \ldots, k\}$ such that $g(i) \neq g(j)$ for all $\{i, j\} \in E$, that is, adjacent vertices are not assigned the same colour. The *chromatic number* of G, denoted by $\chi(G)$, is the minimum value of k for which a proper k-colouring of G exists. A *proper weighted k-colouring* of G is a function h that assigns a set of w_i different colours to each vertex i such that adjacent vertices i and j receive disjoint sets of colours. Formally, a proper weighted k-colouring of G is a map $h : N \to 2^{\{1,2,\ldots,k\}}$ such that $|h(i)| = w_i$ for all $i \in N$ and $h(i) \cap h(j) = \emptyset$ for all $\{i, j\} \in E$. Accordingly, the *weighted chromatic number* of G with respect to w, denoted by $\chi_w(G)$, is the minimum number k needed for a proper weighted k-colouring of G . Note that the clique number and the weighted clique number are lower bounds for the chromatic number and the weighted chromatic number, respectively. Therefore, $\chi(G) \geq \omega(G)$ and $\chi_w(G) \geq \omega_w(G)$. Furthermore, if we let $w_i = 1$ for all $i \in N$, the weighted clique problem and the proper weighted k-colouring problem are equivalent to the clique problem and the proper k-colouring problem, respectively.

Example 5.2. Consider the graph G and the weight vector w displayed in Figure 5.1. We have $\chi(G) = 3$. A proper 3-colouring of G is given by $g(1) = 1$, $g(2) = 2$, $g(3) = 3$ and $g(4) = 1$. Furthermore, note that $\chi_w(G) = 8$ and that a proper weighted 8-colouring of G is given by $h(1) = \{1, 2, 3\}$, $h(2) = \{4, 5\}$, $h(3) = \{6\}$ and $h(4) = \{1, 2, 3, 4, 5, 7, 8\}$. $\diamond$

Finally, we introduce three graph classes discussed in this chapter. A graph G is *perfect* if $\chi(G^S) = \omega(G^S)$ for all induced subgraphs G^S of G, $S \subseteq N$. A *complete r-partite* graph $G = (N, E)$ is a graph whose vertex set can be partitioned into r nonempty partition classes $N_1, N_2, \ldots, N_r$ such that for $k, l \in \{1, 2, \ldots, r\}$ and any two vertices $i \in N_k$ and $j \in N_l$, $\{i, j\} \in E$ if and only if $k \neq l$. A complete graph with n vertices is denoted by K_n and a line graph with n vertices is denoted by P_n. A $(2K_2, P_4)$-*free* graph is a graph that does not have an induced subgraph isomorphic to $2K_2$ or P_4 (see Figure 5.2).

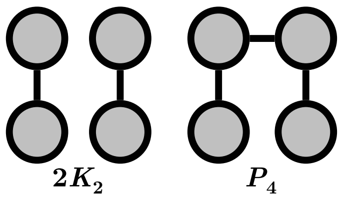


Figure 5.2: $2K_2$ and P_4.

5.2 Definition of the Weighted Minimum Colouring Games

This section introduces the class of weighted minimum colouring games. We start this section with a discussion on the minimum colouring games, which are special instances of the weighted minimum colouring games where all the vertex weights are equal to 1.

Let G be a graph. Then the *minimum colouring (MC) game* (N, c^G) is defined by

$$c^G(S) = \chi(G^S) \text{ for all } S \subseteq N.$$

We illustrate the characteristic function c of an MC game in the following example.

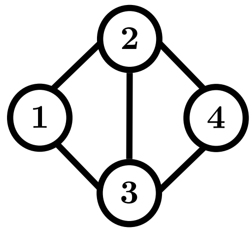


Figure 5.3: Graph G.

Example 5.3. Consider the graph G displayed in Figure 5.3. For the corresponding MC game (N, c^G), we have $N = \{1, 2, 3, 4\}$ and $c^G(N) = \chi(G) = 3$. For $S = \{1, 2\}$, $c^G(S) = \chi(G^S) = 2$ and for $S = \{1, 4\}$, $c^G(S) = \chi(G^S) = 1$. Table 5.1 gives the costs of all the coalitions of the MC game (N, c^G).

S	$\{1\}$	$\{2\}$	$\{3\}$	$\{4\}$	$\{1,2\}$	$\{1,3\}$	$\{1,4\}$	$\{2,3\}$	$\{2,4\}$
$c^G(S)$	1	1	1	1	2	2	1	2	2

S	$\{3,4\}$	$\{1,2,3\}$	$\{1,2,4\}$	$\{1,3,4\}$	$\{2,3,4\}$	$\{1,2,3,4\}$
$c^G(S)$	2	3	2	2	3	3

Table 5.1: Coalitional costs of the MC game (N, c^G).

◇

Properties of the MC games have been characterised in relation to the properties of the graphs that these games are defined on. Firstly, Deng et al. (2000) show that an MC game (N, c^G) is totally balanced if and only if G is a perfect graph. Secondly, Hamers et al. (2011) prove that an MC game (N, c^G) allows a PMAS if and only if G is $(2K_2, P_4)$-free. Finally, Okamoto (2003) demonstrates that an MC game (N, c^G) is submodular if and only if G is complete r-partite. Note that complete r-partite graphs are $(2K_2, P_4)$-free, and $(2K_2, P_4)$-free graphs are perfect, and observe the correspondence between the graph theoretical properties and the game theoretical properties since submodular games allow a PMAS, and games that allow a PMAS are totally balanced.

Example 5.4. Consider the graph G displayed in Figure 5.3 and the MC game (N, c^G) induced by G. In fact, G is a complete 3-partite graph with partition classes $N_1 = \{1, 4\}$, $N_2 = \{2\}$ and $N_3 = \{3\}$. Thus, (N, c^G) is submodular. Consequently, G allows a PMAS and is totally balanced. ◇

Now, we present the class of weighted minimum colouring games.

Definition 5.1. *Let G be a graph and let $w \in \mathbb{Z}_+^N$ be a positive integer weight vector. Then the* weighted minimum colouring (WMC) game $(N, c^{G,w})$ *is defined by*

$$c^{G,w}(S) = \chi_w(G^S) \text{ for all } S \subseteq N.$$

We illustrate the characteristic function c of a WMC game in the following example.

Example 5.5. Consider the graph G and the weight vector w displayed in Figure 5.1. For the corresponding WMC game $(N, c^{G,w})$, we have $N = \{1, 2, 3, 4\}$ and $c^{G,w}(N) = \chi_w(G) = w_3 + w_4 = 1 + 7 = 8$. For $S = \{1, 2\}$, $c^{G,w}(S) = \chi_w(G^S) = w_1 + w_2 = 3 + 2 = 5$ and for $S = \{1, 4\}$, $c^{G,w}(S) = \chi_w(G^S) = w_4 = 7$. Table 5.2 gives the costs of all the coalitions of the WMC game $(N, c^{G,w})$.

S	$\{1\}$	$\{2\}$	$\{3\}$	$\{4\}$	$\{1,2\}$	$\{1,3\}$	$\{1,4\}$	$\{2,3\}$	$\{2,4\}$
$c^{G,w}(S)$	3	2	1	7	5	4	7	3	7

S	$\{3,4\}$	$\{1,2,3\}$	$\{1,2,4\}$	$\{1,3,4\}$	$\{2,3,4\}$	$\{1,2,3,4\}$
$c^{G,w}(S)$	8	6	7	8	8	8

Table 5.2: Coalitional costs of the WMC game $(N, c^{G,w})$.

◇

5.3 Total Balancedness of the Weighted Minimum Colouring Games

In this section, we establish the equivalence of perfect graphs and graphs that induce a totally balanced WMC game for all positive integer weight vectors. Furthermore, we show that any graph induces a totally balanced WMC game for at least one positive integer weight vector. We start by presenting the graph classes considered in this section. Note that, Hamers et al. (2011) point out that for the subclass of minimum coloring games characterising balancedness in terms of properties of the underlying graph seems impossible. Due to this, we focus on characterising total balancedness of the WMC games.

Recall that a graph G is perfect if $\chi(G^S) = \omega(G^S)$ for all induced subgraphs G^S of G, $S \subseteq N$. Let $w \in \mathbb{Z}_+^N$ be a weight vector. We introduce a property, called *w-perfectness*, which states that a graph G is w-perfect if $\chi_w(G^S) = \omega_w(G^S)$ for all

$S \subseteq N$. A graph G is *weighted perfect* if it is w-perfect for all weight vectors $w \in \mathbb{Z}_+^N$. The concept of a weighted perfect graph in graph theory literature can be traced back to the "replication lemma" of Lovász (1972) since repeated application of this lemma implies weighted perfectness (Schrijver, 2003). Note that a graph G that is not perfect can be w-perfect for some $w \in \mathbb{Z}_+^N$. We have the following example.

Example 5.6. Consider the graph G and the weight vector w displayed in Figure 5.4. Note that G is not perfect since $\chi(G) = 3$ and $\omega(G) = 2$. We have $\chi_w(G) = \omega_w(G) = 17$. Moreover, it is easy to verify that $\chi_w(G^S) = \omega_w(G^S)$ for all $S \subset N$. Hence, G is w-perfect. $\diamond$

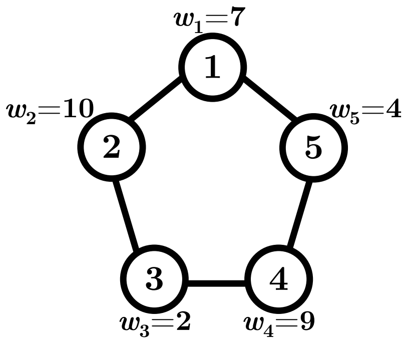


Figure 5.4: G is w-perfect but not perfect.

Before presenting the main results of this section, we have the following lemma stating that a w-perfect graph G induces a totally balanced WMC game.

Lemma 5.1. *Let G be a graph and let $w \in \mathbb{Z}_+^N$. If G is w-perfect, then the corresponding weighted minimum colouring game $(N, c^{G,w})$ is totally balanced.*

Proof: Let $w \in \mathbb{Z}_+^N$. Let G be a w-perfect graph. Let $C \subseteq N$ be a maximum weighted clique in G with respect to w. Let $x^C \in \mathbb{R}^N$ be an allocation for the weighted minimum colouring game $(N, c^{G,w})$ such that

$$x_i^C = \begin{cases} w_i & \text{if } i \in C \\ 0 & \text{otherwise.} \end{cases}$$

We show that this allocation is in the core of $(N, c^{G,w})$. Since G is w-perfect, we have $c^{G,w}(S) = \chi_w(G^S) = \omega_w(G^S)$ for all $S \subseteq N$. For efficiency, we get $\sum_{i \in N} x_i^C = \sum_{i \in C} w_i = \omega_w(G) = c^{G,w}(N)$. For any subset $S \subset N$, we have $\sum_{i \in S} x_i^C = \sum_{i \in (C \cap S)} w_i$.

Moreover, let C^S be a maximum weighted clique in G^S with respect to w. Then $c^{G,w}(S) = \omega_w(G^S) = \sum_{i \in C^S} w_i$. Therefore, for coalitional rationality, we need to show that $\sum_{i \in (C \cap S)} w_i \leq \sum_{i \in C^S} w_i$ for all $S \subset N$. This inequality holds since $C \cap S$ is a clique in G^S and C^S is a maximum weighted clique in G^S with respect to w. Hence, $(N, c^{G,w})$ is balanced. Since every induced subgraph of a w-perfect graph is also w-perfect, $(N, c^{G,w})$ is totally balanced. □

The following theorem characterises graphs that induce a totally balanced WMC game for all positive weight vectors. G is w-perfect, then the corresponding weighted minimum colouring game $(N, c^{G,w})$ is totally balanced.

Theorem 5.1. *For a graph G, the following statements are equivalent.*

(i) G is perfect.

(ii) G is weighted perfect.

(iii) The corresponding weighted minimum colouring game $(N, c^{G,w})$ is totally balanced for all $w \in \mathbb{Z}_+^N$.

Proof: (i) $\Rightarrow$ (ii): This result is due to Grötschel et al. (1988).

(ii) $\Rightarrow$ (iii): This result directly follows from Lemma 5.1.

(iii) $\Rightarrow$ (i): Let G be a graph that induces a totally balanced WMC game for all $w \in \mathbb{Z}_+^N$. Let $w_i = 1$ for all $i \in N$. In fact, the WMC game $(N, c^{G,w})$ is the minimum colouring game on G. Since Deng et al. (2000) showed that a minimum colouring game is totally balanced if and only if G is a perfect graph, G is perfect. □

Next, we show that any graph induces a totally balanced WMC game for at least one positive integer weight vector.

Theorem 5.2. *Let G be a graph. Then there exists at least one $w \in \mathbb{Z}_+^N$ such that the corresponding weighted minimum colouring game $(N, c^{G,w})$ is totally balanced.*

Proof: In order to prove this theorem, we will first show the existence of a positive integer weight vector such that for any graph the maximum weighted clique is unique. Next, we show that in this case the vertices of the graph can be coloured using no more colours than the weighted clique number. Since the weighted clique number is always a lower bound on the weighted chromatic number, the weighted clique number and the weighted chromatic number are equal to each other. The result holds for all subgraphs of this graph and hence from Lemma 5.1 the graph induces a totally balanced WMC game for this weight vector.

Let G be a graph and let $w_i = 2^{i-1}$ for all $i \in N$. First, we show that G is w-perfect, that is, $\chi_w(G^S) = \omega_w(G^S)$ for all $S \subseteq N$. We start by discussing some properties of a maximum weighted clique in G with respect to w. Firstly, two different cliques in G do not have the same weight with respect to w since the binary representations of their weights, which are in fact the characteristic vectors of these cliques, are always different. Therefore, the maximum weighted clique in G with respect to w is unique. Let C be the maximum weighted clique in G with respect to w such that $C = \{i_1, i_2, \ldots, i_k\}$. Without loss of generality assume that $i_1 > i_2 > \ldots > i_k$. We have $\omega_w(G) = 2^{i_1-1} + 2^{i_2-1} + \ldots + 2^{i_k-1}$. Next, we show that i_1 is the vertex with the maximum index in N, that is, $i_1 = n$. Assume that $n \notin C$. We have $\sum_{j \in C} w_j \leq 1 + 2 + \ldots + 2^{n-2} < 2^{n-1} = w_n$, which contradicts C being the maximum weighted clique. Therefore, $n \in C$ and $i_1 = n$. Finally, we show that for all l such that $2 \leq l \leq k$, $i_l = \max\{j \in \{1, 2, \ldots, i_{l-1} - 1\} : j \text{ is adjacent to } i_1, i_2, \ldots, i_{l-1}\}$. Consider the case where $l = 2$ and assume that there exists a vertex $m \notin C$ that is adjacent to i_1 such that $m > i_2$. We have $w_{i_1} + w_m = 2^{n-1} + 2^{m-1} > w_{i_1} + w_{i_2} + \ldots + w_{i_k}$, which again contradicts C being the maximum weighted clique. Therefore, $i_2 = \max\{j \in \{1, 2, \ldots, i_1 - 1\} : j \text{ is adjacent to } i_1\}$. With a similar argument, it can be verified that for $3 \leq l \leq k$, i_l is the vertex with the maximum index that is adjacent to vertices $i_1, i_2, \ldots, i_{l-1}$.

Now, we present a colouring of the vertices of G using no more than $\omega_w(G) = \sum_{l=1}^{k} w_{i_l}$ colours. We start by colouring each vertex i_l of C with $w_{i_l} = 2^{i_l-1}$ different colours, therefore using $\omega_w(G)$ different colours in total. Next, we colour the vertices in $N \backslash C$. First, we construct a partition of $N \backslash C$ with k elements. Let A_1 be the set of vertices in $N \backslash C$ that are not adjacent to i_1. Let A_l be the set of vertices in $N \backslash C$ that are adjacent to vertices $i_1, i_2, \ldots, i_{l-1}$ but not to i_l for $2 \leq l \leq k$. Since the maximum weighted clique C is also a maximal clique, there does not exist a vertex in $N \backslash C$ that is adjacent to all the vertices in C. Therefore, $\bigcup_{l=1}^{k} A_l = N \backslash C$ and by construction the elements of the partition are pairwise disjoint. Since a vertex in A_l is not adjacent to i_l, the corresponding w_{i_l} colours can be used to colour this vertex. Furthermore, the vertex with the maximum index in A_l has at most an index of $i_l - 1$. This is due to the two aforementioned properties of C that i_1 is the vertex with the maximum index in N, that is, $i_1 = n$ and that for $2 \leq l \leq k$, i_l is the vertex with the maximum index that is adjacent to vertices $i_1, i_2, \ldots, i_{l-1}$. Hence, for all $l \in \{1, 2, \ldots, k\}$, we have $A_l \subseteq \{1, 2, \ldots, i_l - 1\}$, which in turn implies $\sum_{j \in A_l} w_j \leq 1 + 2 + \ldots + 2^{i_l - 2} < 2^{i_l - 1} = w_{i_l}$. Therefore, the w_{i_l} distinct colours that are used to colour vertex $i_l \in C$ are sufficient to colour all the vertices in A_l. Since $\bigcup_{l=1}^{k} A_l = N \backslash C$, all the vertices in $N \backslash C$ are coloured using no more than $\omega_w(G) = \sum_{l=1}^{k} w_{i_l}$ colours. Thus, $\chi_w(G) \leq \omega_w(G)$. Recall that the weighted clique number is a lower bound on the weighted chromatic number, that is, $\chi_w(G) \geq \omega_w(G)$. Therefore, $\chi_w(G) = \omega_w(G)$.

A similar argument holds for all the weighted subgraphs of G, and thus G is w-perfect for $w_i = 2^{i-1}$ for all $i \in N$. From Lemma 5.1, $(N, c^{G,w})$ is totally balanced. Therefore, there exists at least one $w \in \mathbb{Z}_+^N$ such that the corresponding weighted minimum colouring game $(N, c^{G,w})$ is totally balanced. □

We conclude this section by illustrating the above proof by means of an example.

Example 5.7. Consider the graph G and the weight vector w displayed in Figure 5.5.

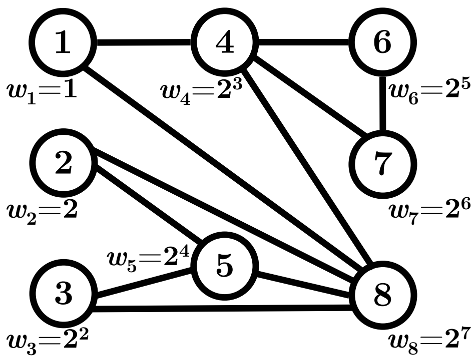


Figure 5.5: Graph G induces a totally balanced WMC game for at least one positive integer weight.

Note that $w_i = 2^{i-1}$ for all $i \in N$. From the proof of Theorem 5.2, we know that in order to show that the WMC game $(N, c^{G,w})$ is totally balanced for at least one $w \in \mathbb{Z}_+^N$, it is sufficient to show that G is w-perfect. The maximum weighted clique in G with respect to w is $C = \{3, 5, 8\}$. Let us denote the vertex in C with the maximum label with i_1, the second largest label by i_2 and finally the third largest label by i_3. Therefore, $i_1 = 8$, $i_2 = 5$ and $i_3 = 3$. Observe that $i_1 = 8$ is the vertex with maximum index in N, that $i_2 = 5$ is the vertex with maximum weight in N that is adjacent to $i_1 = 8$ and that $i_3 = 3$ is the vertex with maximum weight in N that is adjacent to $i_1 = 8$ and $i_2 = 5$.

We have $\omega_w(G) = 2^7 + 2^4 + 2^2$. Next, we colour the vertices in N using no more than $\omega_w(G)$ colours. In order to colour the vertices in C, $\omega_w(G)$ distinct colours are needed. Let us now partition the vertices in $N\backslash C = \{1, 2, 4, 6, 7\}$ in the following manner. Let $A_1 = \{6, 7\}$ be the set of vertices that are not connected to $i_1 = 8$. Let $A_2 = \{1, 4\}$ be the set of vertices that are connected to $i_1 = 8$ but not to $i_2 = 5$, and let $A_3 = \{2\}$ be the set of vertices that are connected to $i_1 = 8$ and $i_2 = 5$ but not to $i_3 = 3$. The 2^7 distinct colours that are used to colour $i_1 = 8$ can be used to colour the vertices in A_1 since these vertices are not adjacent to i_1. Furthermore, the 2^7 distinct colours that are used to colour $i_1 = 8$ are sufficient to colour the vertices in A_1 since $2^5 + 2^6 \leq 2^7$. Similarly, the 2^4 and 2^2 distinct colours that are used to colour $i_2 = 5$ and $i_3 = 3$ can be used and are sufficient to colour the vertices in A_2 and A_3, respectively. Thus, all the vertices in $N\backslash C$ are coloured by using no more than $\omega_w(G)$ colours and hence $\chi_w(G) \leq \omega_w(G)$. Recall that $\chi_w(G) \geq \omega_w(G)$ always holds. Therefore, $\chi_w(G) = \omega_w(G) = 2^7 + 2^4 + 2^2$. A similar argument holds for all the weighted subgraphs of G, and thus G is w-perfect. $\diamond$

5.4 Submodularity of the Weighted Minimum Colouring Games

This section establishes the equivalence of complete r-partite graphs and graphs that induce a submodular WMC game for all positive integer weight vectors, as well as of $(2K_2, P_4)$-free graphs that induce a submodular WMC game for at least one positive integer weight vector. Recall that the two graph theoretical properties that are used in this section, namely complete r-partiteness and $(2K_2, P_4)$-freeness, have been formally defined in Section 5.1.

The following theorem establishes the equivalence of complete r-partite graphs and graphs that induce a submodular WMC game for all positive integer weight vectors.

Theorem 5.3. *G is a complete r-partite graph if and only if the corresponding weighted minimum colouring game $(N, c^{G,w})$ is submodular for all $w \in \mathbb{Z}_+^N$.*

Proof: We start with the 'if'-part. Let $w_i = 1$ for all $i \in N$. In fact, the WMC game $(N, c^{G,w})$ is the minimum colouring game on G. Since Okamoto (2003) showed that a minimum colouring game is submodular if and only if G is a complete r-partite graph, G is complete r-partite.

Next, we prove the 'only-if'-part. Let G be a complete r-partite graph and let $N_1, N_2, \ldots, N_r$ be the partition classes of the vertex set N. A maximum clique in G consists of exactly one vertex from each one of the r partition classes and hence has exactly r elements. Let $w \in \mathbb{Z}_+^N$. A maximum weighted clique in G with respect to w is a maximum clique in G. Moreover, each vertex in a maximum weighted clique in G is a maximum weighted vertex in the partition class that it belongs to. Let C be a maximum weighted clique in G with respect to w. Then we have $\sum_{i \in C} w_i = \sum_{k=1}^{r} \max_{i \in N_k} w_i$.

Let $S \subseteq N$. We define $\mathcal{N}_S = \{k \in \{1, 2, \ldots, r\} : S \cap N_k \neq \emptyset\}$ to be the set of indices of the partition classes that have at least one common vertex with S. Now, consider the subgraph G^S and note that G^S is a complete multipartite graph with $|\mathcal{N}_S|$ partition classes, that is, a complete $|\mathcal{N}_S|$-partite graph. Let C^S be a maximum weighted clique in G^S with respect to w. Let $k \in \mathcal{N}_S$. Since G^S is a complete multipartite graph, we know that C^S has exactly one maximum weighted vertex from $S \cap N_k$. Thus, $\omega_w(G^S) = \sum_{j \in C^S} w_j = \sum_{k \in \mathcal{N}_S} \max_{j \in S \cap N_k} w_j$. Moreover, note that a complete multipartite graph is perfect. Therefore, for $S \subseteq N$, we have $c^{G,w}(S) = \chi_w(G^S) = \omega_w(G^S) = \sum_{k \in \mathcal{N}_S} \max_{j \in S \cap N_k} w_j$.

Now, let $i \in N$ and let $S \subseteq N \backslash \{i\}$. Furthermore, let $p(i) \in \{1, 2, \ldots, r\}$ such that $i \in N_{p(i)}$. We have two cases to consider, namely $p(i) \notin \mathcal{N}_S$ and $p(i) \in \mathcal{N}_S$. Firstly, if $p(i) \notin \mathcal{N}_S$, then $\mathcal{N}_{S \cup \{i\}} = \mathcal{N}_S \cup \{p(i)\}$. Moreover, $\max_{j \in (S \cup \{i\}) \cap N_k} w_j = \max_{j \in S \cap N_k} w_j$ for all $k \in \mathcal{N}_S$ and $\max_{j \in (S \cup \{i\}) \cap N_{p(i)}} w_j = w_i$. Therefore,

$$\begin{aligned} c^{G,w}(S \cup \{i\}) - c^{G,w}(S) &= \sum_{k \in \mathcal{N}_{S \cup \{i\}}} \max_{j \in (S \cup \{i\}) \cap N_k} w_j - \sum_{k \in \mathcal{N}_S} \max_{j \in S \cap N_k} w_j \\ &= w_i + \sum_{k \in \mathcal{N}_S} \max_{j \in S \cap N_k} w_j - \sum_{k \in \mathcal{N}_S} \max_{j \in S \cap N_k} w_j \\ &= w_i \end{aligned}$$

if $p(i) \notin \mathcal{N}_S$. Secondly, if $p(i) \in \mathcal{N}_S$, then $\mathcal{N}_S = \mathcal{N}_{S \cup \{i\}}$. Moreover, $\max_{j \in (S \cup \{i\}) \cap N_k} w_j = \max_{j \in S \cap N_k} w_j$ for all $k \in \mathcal{N}_S \backslash \{p(i)\}$ and $\max_{j \in (S \cup \{i\}) \cap N_{p(i)}} w_j = \max(w_i, w_{j^*_{S,i}})$ where $j^*_{S,i} \in S \cap N_{p(i)}$ such that $w_{j^*_{S,i}} = \max_{j \in S \cap N_{p(i)}} w_j$. Therefore,

$$c^{G,w}(S \cup \{i\}) - c^{G,w}(S) = \sum_{k \in \mathcal{N}_{S \cup \{i\}}} \max_{j \in (S \cup \{i\}) \cap N_k} w_j - \sum_{k \in \mathcal{N}_S} \max_{j \in S \cap N_k} w_j$$

$$= \max(w_i, w_{j^*_{S,i}}) + \sum_{k \in \mathcal{N}_S \setminus \{p(i)\}} \max_{j \in S \cap N_k} w_j - w_{j^*_{S,i}} - \sum_{k \in \mathcal{N}_S \setminus \{p(i)\}} \max_{j \in S \cap N_k} w_j$$
$$= \max(w_i - w_{j^*_{S,i}}, 0)$$

if $p(i) \in \mathcal{N}_S$. In order to prove submodularity, let $i \in N$ and $S \subset T \subseteq N \setminus \{i\}$ and consider the following cases.

Case 1. $p(i) \notin \mathcal{N}_T$. Then $p(i) \notin \mathcal{N}_S$ since $S \subset T$. Thus,

$$\begin{aligned} c^{G,w}(S \cup \{i\}) - c^{G,w}(S) &= w_i \\ &= c^{G,w}(T \cup \{i\}) - c^{G,w}(T). \end{aligned}$$

Case 2. $p(i) \in \mathcal{N}_T$ and $p(i) \notin \mathcal{N}_S$. Then we have

$$\begin{aligned} c^{G,w}(S \cup \{i\}) - c^{G,w}(S) &= w_i \\ &\geq \max(w_i - w_{j^*_{T,i}}, 0) \\ &= c^{G,w}(T \cup \{i\}) - c^{G,w}(T). \end{aligned}$$

Case 3. $p(i) \in \mathcal{N}_T$ and $p(i) \in \mathcal{N}_S$.

$$\begin{aligned} c^{G,w}(S \cup \{i\}) - c^{G,w}(S) &= \max(w_i - w_{j^*_{S,i}}, 0) \\ &\geq \max(w_i - w_{j^*_{T,i}}, 0) \\ &= c^{G,w}(T \cup \{i\}) - c^{G,w}(T) \end{aligned}$$

where the inequality holds since $w_{j^*_{T,i}} \geq w_{j^*_{S,i}}$ for $S \subset T$.

Therefore, $c^{G,w}(S \cup \{i\}) - c^{G,w}(S) \geq c^{G,w}(T \cup \{i\}) - c^{G,w}(T)$ for all $i \in N$ and $S \subset T \subseteq N \setminus \{i\}$. Since this result holds for every positive integer weight vector w, the corresponding weighted minimum colouring game $(N, c^{G,w})$ is submodular for all $w \in \mathbb{Z}^N_+$. □

We have an example illustrating the above proof.

Example 5.8. Consider the complete 3-partite graph G with partition classes $N_1 = \{1,2,3\}$, $N_2 = \{4,5\}$ and $N_3 = \{6,7\}$, and the weight vector w displayed in Figure 5.6.

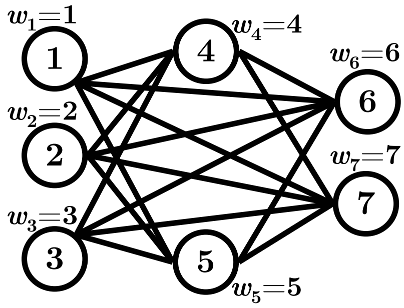


Figure 5.6: Graph G induces a submodular WMC game for all positive integer weight vectors.

The clique $C = \{3,5,7\}$ is the maximum weighted clique in G with respect to w. We have $c^{G,w}(N) = \chi_w(G) = \omega_w(G) = \sum_{i \in C} w_i = w_3 + w_5 + w_7 = 15$. Let $S = \{1\}$ and $T = \{1,2,5\}$. Observe that G^S and G^T are complete multipartite graphs. We have $\mathcal{N}_S = \{1\}$ and $\mathcal{N}_T = \{1,2\}$. Moreover, $C^S = \{1\}$ and $C^T = \{2,5\}$ giving us $c^{G,w}(S) = w_1 = 1$ and $c^{G,w}(T) = w_2 + w_5 = 7$. Next, for a number of different vertices $i \in N$ such that $S \subset T \subseteq N \backslash \{i\}$, we illustrate that

$$c^{G,w}(S \cup \{i\}) - c^{G,w}(S) \geq c^{G,w}(T \cup \{i\}) - c^{G,w}(T).$$

Firstly, let $i = 6$. We have $p(6) = 3$, $3 \notin \mathcal{N}_T$ and hence $3 \notin \mathcal{N}_S$. We get

$$\begin{aligned} c^{G,w}(S \cup \{6\}) - c^{G,w}(S) &= w_1 + w_6 - w_1 \\ &= w_6 \\ &= w_2 + w_5 + w_6 - [w_2 + w_5] \\ &= c^{G,w}(T \cup \{6\}) - c^{G,w}(T). \end{aligned}$$

Secondly, let $i = 4$. We have $p(4) = 2$, $2 \in \mathcal{N}_T$ and $2 \notin \mathcal{N}_S$. We get

$$\begin{aligned} c^{G,w}(S \cup \{4\}) - c^{G,w}(S) &= w_1 + w_4 - w_1 \\ &= w_4 \\ &\geq 0 \\ &= \max(w_4 - w_5, 0) \end{aligned}$$

$$= w_2 + w_5 - [w_2 + w_5]$$
$$= c^{G,w}(T \cup \{4\}) - c^{G,w}(T).$$

Finally, let $i = 3$. We have $p(3) = 1$, $1 \in \mathcal{N}_T$ and $1 \in \mathcal{N}_S$. We get

$$\begin{aligned} c^{G,w}(S \cup \{3\}) - c^{G,w}(S) &= \max(w_3 - w_1, 0) \\ &= w_3 - w_1 \\ &\geq w_3 - w_2 \\ &= \max(w_3 - w_2, 0) \\ &= w_3 + w_5 - [w_2 + w_5] \\ &= c^{G,w}(T \cup \{3\}) - c^{G,w}(T). \end{aligned}$$

$\diamond$

Next, we show the equivalence of $(2K_2, P_4)$-free graphs and graphs that induce a submodular WMC game for at least one positive integer weight vector. To do so, we first mention the relationship between a $(2K_2, P_4)$-free graph and a rooted forest, and introduce a special weighting of the vertices of a rooted forest.

Let (N, A) be a directed graph where $N = \{1, 2, \ldots, n\}$ is the finite vertex set and $A \subseteq \{(i, j) : i, j \in N, i \neq j\}$ is the set of directed arcs. A *rooted tree* is a directed graph with a special vertex $r \in N$, called the *root*, such that for each vertex $i \in N$ there exists a unique path from r to i. The disjoint union of rooted trees is called a *rooted forest*. If $F = (N, A)$ is a rooted forest, then for every $i \in N$ there is a unique path from some root to i. Let $P(i)$ denote the collection of vertices on this path. The set of *descendants* of a vertex $i \in N$ is the set $D(i) = \{j \in N : i \in P(j) \text{ and } i \neq j\}$.

Every rooted forest induces a $(2K_2, P_4)$-free graph in the following manner. Let $F = (N, A)$ be a rooted forest. Let $G(F) = (N, E)$ be a graph such that $\{i, j\} \in E$ if and only if $i \notin P(j)$ and $j \notin P(i)$ in F. Then $G(F)$ is a $(2K_2, P_4)$-free graph and F is referred to as a forest representation of $G(F)$. This result is due to the fact that a $(2K_2, P_4)$-free graph is the complement of a quasi-threshold graph, or equivalently, a (C_4, P_4)-free graph where C_n is a cycle with n vertices (see Figure 5.7).

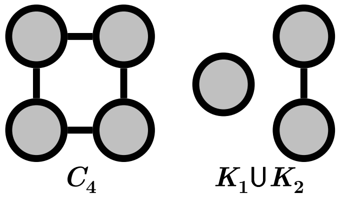


Figure 5.7: C_4 and $K_1 \cup K_2$.

Every quasi-threshold graph is induced by a rooted forest F by adding an edge between vertices i and j if and only if $i \in D(j)$ or $j \in D(i)$ in F (Wölk (1965), Yan et al. (1996)).

Example 5.9. Consider the rooted forest F and graph $G(F)$ in Figure 5.8. The rooted forest F is a forest representation of the $(2K_2, P_4)$-free graph $G(F)$. $\diamond$

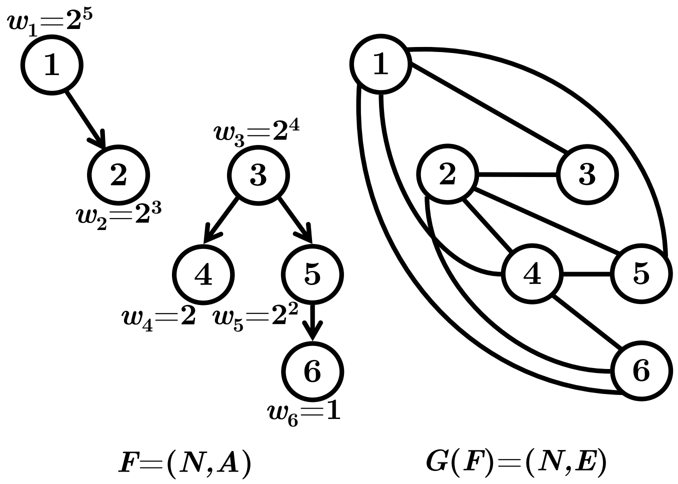


Figure 5.8: A rooted forest F and the $(2K_2, P_4)$-free graph $G(F)$ induced by F.

Now, we introduce a weighting of the vertices of a rooted forest F. We start by considering a partition of the vertices of a rooted forest F. Let $N^0 = \{j \in N : (i,j) \notin A \text{ for all } i \in N\}$. The elements of N^0 are the roots of the rooted trees that constitute F. A root $r \in N^0$ is the root of a vertex i if $i \in D(r)$. Furthermore, let us refer to the distance $d(i,j)$ between $i \in N$ and $j \in N$ as the number of edges on the path from i to j. Let M denote the maximum distance from any of the vertices in N to its root. Let $N^k = \{i \in N : \text{ there exists a root } r \in N^0 \text{ such that } d(r,i) = k\}$. Then $N = \bigcup_{k=0}^{M} N^k$

is a partition of N. Now, consider a permutation of the vertices in N such that all the vertices in N^0 precede all the vertices in N^1, all the vertices in N^1 precede all the vertices in N^2 and so on up to all the vertices in N^{M-1} precede all the vertices in N^M. Let $\sigma = [[N^0], [N^1], \ldots, [N^M]]$ denote such a permutation where $[N^k]$ is a permutation of the vertices in N^k for $k \in \{0, 1, \ldots, M\}$. Note that $[N^k]$ can be chosen arbitrarily. We refer to σ as a *root-first* permutation of the vertices in N. A *root-first* 2*-weighting* of the vertices of a rooted forest $F = (N, A)$ is a bijection $f : N \to \{1, 2, \ldots, 2^{n-1}\}$ such that $f(i) = 2^{n-\sigma(i)}$.

Example 5.10. Consider the rooted forest F in Figure 5.8. We have $M = 2$, and $N^0 = \{1, 3\}$, $N^1 = \{2, 4, 5\}$ and $N^2 = \{6\}$. Furthermore, let $\sigma = [1, 3, 2, 5, 4, 6]$ be a root-first permutation of N. Then the corresponding root-first 2-weighting of N is $f(1) = 2^5$, $f(2) = 2^3$, $f(3) = 2^4$, $f(4) = 2$, $f(5) = 2^2$, $f(6) = 1$. $\diamond$

Observe that a root $r \in N^0$ is not adjacent to any of its descendants $D(r)$ on the corresponding $(2K_2, P_4)$-free graph $G(F)$. Let $w_i = f(i)$ for all $i \in N$. In order to colour a root $r \in N^0$, w_r colours are needed. For the root-first 2-weighting, we have $w_r > \sum_{i \in D(r)} w_i$ for all $r \in N^0$, ensuring that the w_r colours are adequate to colour all the vertices in $D(r)$ on $G(F)$. Therefore, the weighted chromatic number of $G(F)$ with respect to w is $\chi_w(G(F)) = \sum_{r \in N^0} w_r$. In fact, a similar result can be derived for the weighted chromatic number of the subgraph $G^S(F)$ with respect to w in the following manner. Let $S \subset N$. A vertex in S is called an S*-root* if it is not the descendant of any other vertex in S. Let $S^0 = \{j \in S : j \notin D(i) \text{ for all } i \in S\}$ denote the set of S-roots. For root-first 2-weighting, we have $w_r > \sum_{i \in D(r)} w_i$ for all $r \in S^0$. Therefore, $\chi_w(G^S(F)) = \sum_{r \in S^0} w_r$.

Example 5.11. Consider the rooted forest in Figure 5.8. Since $N^0 = \{1, 3\}$, $\chi_w(G(F)) = w_1 + w_3 = 2^5 + 2^4$. Moreover, let $S = \{2, 3, 6\}$. We have $S^0 = \{2, 3\}$ and $\chi_w(G^S(F)) = w_2 + w_3 = 2^3 + 2^4$. $\diamond$

Before establishing the equivalence of $(2K_2, P_4)$-free graphs and graphs that induce a submodular WMC game for at least one positive integer weight vector, we have the following lemma stating that a graph that is isomorphic to $K_1 \cup K_2$ (see Figure 5.7) induces a submodular WMC game if and only if the weight of the vertex to which no edge is incident is greater than or equal to the sum of the weights of the vertices that are connected by an edge.

Lemma 5.2. *Let $G = (N, E)$ be a graph with $N = \{1, 2, 3\}$ and $E = \{\{1, 2\}\}$. Let $w = (w_1, w_2, w_3)$ be a positive weight vector. Then the corresponding WMC game $(N, c^{G,w})$ is submodular if and only if $w_3 \geq w_1 + w_2$.*

Proof: The costs of all the coalitions of the WMC game $(N, c^{G,w})$ are displayed in Table 5.3.

S	$\{1\}$	$\{2\}$	$\{3\}$	$\{1,2\}$	$\{1,3\}$	$\{2,3\}$	$\{1,2,3\}$
$c^{G,w}(S)$	w_1	w_2	w_3	$w_1 + w_2$	$\max(w_1, w_3)$	$\max(w_2, w_3)$	$\max(w_1 + w_2, w_3)$

Table 5.3: Coalitional costs of the WMC game $(N, c^{G,w})$.

First, we prove the 'only-if'-part. Let $(N, c^{G,w})$ be submodular and assume, on the contrary, that $w_3 < w_1 + w_2$. Then we have

$$w_1 + w_2 + w_3 > \max(w_1, w_3) + \max(w_2, w_3)$$

which is equivalent to

$$\max(w_1 + w_2, w_3) + w_3 > \max(w_1, w_3) + \max(w_2, w_3).$$

Hence,

$$c^{G,w}(\{1, 2, 3\}) - c^{G,w}(\{3\}) > c^{G,w}(\{1, 3\}) - c^{G,w}(\{2, 3\})$$

and $(N, c^{G,w})$ is not submodular.

The 'if'-part follows readily by checking that all submodularity conditions are satisfied if $w_3 \geq w_1 + w_2$. □

The following theorem shows that $(2K_2, P_4)$-freeness and local WMC submodularity of a graph G are equivalent.

Theorem 5.4. *A graph G is $(2K_2, P_4)$-free if and only if there exists at least one $w \in \mathbb{Z}_+^N$ such that the corresponding weighted minimum colouring game $(N, c^{G,w})$ is submodular.*

Proof: We start with the 'if'-part. Assume that there exists at least one $w \in \mathbb{Z}_+^N$ such that the WMC game $(N, c^{G,w})$ is submodular but assume, on the contrary, that G has a subgraph isomorphic to $2K_2$ or P_4. Without loss of generality, let G be a graph such that $\{1,2,3,4\} \subseteq N$, $\{1,2\}, \{3,4\} \in E$ and $\{1,3\}, \{2,3\}, \{2,4\} \notin E$. Let $S = \{1,2,3\}$. Let $w \in \mathbb{Z}_+^N$. From Lemma 5.2, (S, c^{G^S,w^S}) is submodular if and only if $w_3 \geq w_1 + w_2$. Now, let $S = \{2,3,4\}$. Similarly, (S, c^{G^S,w^S}) is submodular if and only if $w_2 \geq w_3 + w_4$ from Lemma 5.2. Adding the inequalities $w_3 \geq w_1 + w_2$ and $w_2 \geq w_3 + w_4$ gives us $0 \geq w_1 + w_4$, which contradicts $w_1, w_4 > 0$. Therefore, the WMC game $(N, c^{G,w})$ is not submodular for any $w \in \mathbb{Z}_+^N$. This contradicts our initial assumption that there exists at least one $w \in \mathbb{Z}_+^N$ such that the WMC game $(N, c^{G,w})$ is submodular. Hence, G does not have any subgraph isomorphic to $2K_2$ or P_4 and hence is $(2K_2, P_4)$-free.

Next, we prove the 'only-if'-part. Observe that the characteristic function of a WMC game $c^{G,w}$ satisfies $c^{G,w}(T) = \chi_w(G^T) \geq \chi_w(G^S) = c^{G,w}(S)$ for all $S \subseteq T \subseteq N$ and for all $w \in \mathbb{Z}_+^N$ and thus is monotone.

Let $G = (N, E)$ be a $(2K_2, P_4)$-free graph and let $F = (N, A)$ be a rooted forest representation of G. Let us replace the notation of G with $G(F)$ to represent the fact that it is derived from F. Let f be a root-first 2-weighting and let $w_i = f(i)$ for all $i \in N$. Let $i \in N$ and $S \subset T \subseteq N \backslash \{i\}$. We have to show that

$$c^{G(F),w}(S \cup \{i\}) - c^{G(F),w}(S) \geq c^{G(F),w}(T \cup \{i\}) - c^{G(F),w}(T). \tag{5.1}$$

Firstly, assume that i is not a root in $T \cup \{i\}$, that is, $i \notin (T \cup \{i\})^0$. Then we have $c^{G(F),w}(T \cup \{i\}) - c^{G(F),w}(T) = 0$. Furthermore, we have $c^{G(F),w}(S \cup \{i\}) - c^{G(F),w}(S) \geq 0$ from the monotonicity of $c^{G(F),w}$ and hence (5.1) holds.

Therefore, we assume that i is a root in $T \cup \{i\}$, that is, $i \in (T \cup \{i\})^0$. Let $S_i^0 = \{s \in S^0 : s \in D(i)\}$ and $T_i^0 = \{t \in T^0 : t \in D(i)\}$ be the set of roots in S and T, respectively, that are descendants of i. Then

$$c^{G(F),w}(S \cup \{i\}) - c^{G(F),w}(S) = w_i + \sum_{j \in S^0 \backslash S_i^0} w_j - \sum_{j \in S^0} w_j$$

$$= w_i - \sum_{j \in S_i^0} w_j$$

and similarly

$$c^{G(F),w}(T \cup \{i\}) - c^{G(F),w}(T) = w_i - \sum_{j \in T_i^0} w_j.$$

Hence, in order to prove (5.1), it is sufficient to show that

$$\sum_{j \in T_i^0} w_j \geq \sum_{j \in S_i^0} w_j. \tag{5.2}$$

Let $t \in T_i^0$. Since $T_i^0 \neq \emptyset$, we know that such t always exists. Furthermore, assume that there exists a root $s \in S_i^0$. Since $S \subset T$ and s and t are roots in S and T, respectively, $s \in D(i)$ and $t \in D(i)$ imply either $s = t$ or $s \in D(t)$. Let $R_1 = T_i^0 \cap S_i^0$, $R_2 = T_i^0 \backslash S_i^0$ and $R_3 = S_i^0 \backslash T_i^0$. Obviously, R_1 and R_2 form a partition of T_i^0, and R_1 and R_3 form a partition of S_i^0. Moreover, from the above argument, we get $R_3 \subseteq D(R_2)$ where $D(R_2) = \{j \in N : j \in D(t) \text{ for some } t \in R_2\}$. Now, we prove (5.2). We have

$$\begin{aligned}
\sum_{j \in T_i^0} w_j &= \sum_{j \in R_1} w_j + \sum_{j \in R_2} w_j \\
&\geq \sum_{j \in R_1} w_j + \sum_{j \in D(R_2)} w_j \\
&\geq \sum_{j \in R_1} w_j + \sum_{j \in R_3} w_j \\
&= \sum_{j \in S_i^0} w_j
\end{aligned}$$

where the first inequality follows from root-first 2-weighting since $w_t > \sum_{j \in D(t)} w_j$ for all $t \in R_2$. Therefore, $(N, c^{G(F),w})$ is submodular and hence there exists at least one $w \in \mathbb{Z}_+^N$ such that the WMC game $(N, c^{G(F),w})$ is submodular. □

We have the following example to illustrate the above proof.

Example 5.12. Consider the rooted forest F and the weight vector w, which is a root-first 2-weighting, displayed in Figure 5.9.

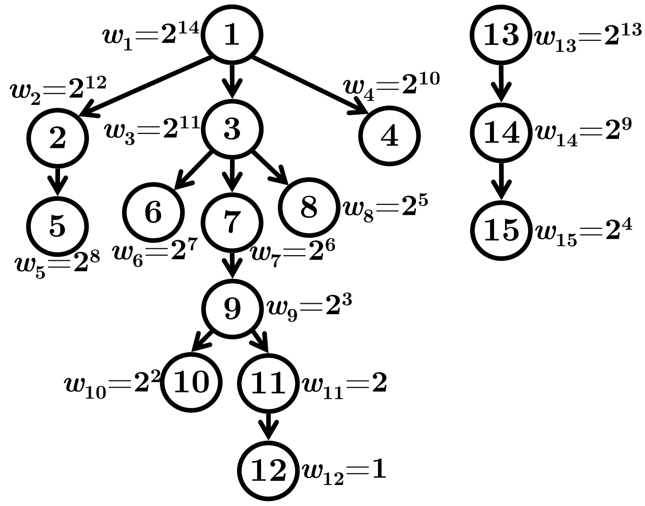


Figure 5.9: Rooted forest F and a weight vector w.

Let $G(F)$ be the graph induced by F. Recall that $P(i)$ denotes the collection of vertices on the unique path from the root of i to i in F and that $\{i, j\} \in E$ if and only if $i \notin P(j)$ and $j \notin P(i)$ in F. We know that $G(F)$ is $(2K_2, P_4)$-free. Let $S = \{6, 10, 11, 12\}$ and $T = \{5, 6, 7, 8, 10, 11, 12\}$. The roots of S and T are $S^0 = \{6, 10, 11\}$ and $T^0 = \{5, 6, 7, 8\}$, respectively. Therefore, $c^{G(F),w}(S) = w_6+w_{10}+w_{11}$ and $c^{G(F),w}(T) = w_5+w_6+w_7+w_8$. Next, for a number of different vertices $i \in N$ such that $S \subset T \subseteq N\backslash\{i\}$, we illustrate that

$$c^{G(F),w}(S \cup \{i\}) - c^{G(F),w}(S) \geq c^{G(F),w}(T \cup \{i\}) - c^{G(F),w}(T). \tag{5.3}$$

Let $i = 9$. Since 9 is not a root in $T\cup\{9\}$, we have $c^{G(F),w}(T\cup\{9\})-c^{G(F),w}(T) = 0$. Similarly, $c^{G(F),w}(T \cup \{4\}) - c^{G(F),w}(T) = 0$ for $i = 4$, and $c^{G(F),w}(T \cup \{12\}) - c^{G(F),w}(T) = 0$ for $i = 12$. Furthermore, $c^{G(F),w}(S \cup \{i\}) - c^{G(F),w}(S) \geq 0$ from the monotonicity of $c^{G(F),w}$. Therefore, for all three cases (5.3) holds.

Now, we concentrate on vertices $i \in N$ that become a root in $T \cup \{i\}$. Therefore, to prove (5.3), we need to show that

$$\sum_{j\in T_i^0} w_j \geq \sum_{j\in S_i^0} w_j. \tag{5.4}$$

Firstly, let $i = 2$. We have $T_2^0 = \{5\}$ and $S_2^0 = \emptyset$. Therefore, (5.4) holds from $w_5 > 0$. Note that $R_1 = R_3 = \emptyset$ and $R_2 = \{5\}$. Furthermore, $D(R_2) = \emptyset$ and $R_3 \subseteq D(R_2)$.

Secondly, let $i = 3$. Then $T_3^0 = \{6, 7, 8\}$ and $S_3^0 = \{6, 10, 11\}$. We have $R_1 = \{6\}$, $R_2 = \{7, 8\}$ and $R_3 = \{10, 11\}$. Moreover, $D(R_2) = \{9, 10, 11, 12\}$ and $R_3 \subseteq D(R_2)$. We get

$$\begin{aligned}
\sum_{j \in T_3^0} w_j &= w_6 + w_7 + w_8 \\
&\geq w_6 + w_9 + w_{10} + w_{11} + w_{12} \\
&\geq w_6 + w_{10} + w_{11} \\
&= \sum_{j \in S_3^0} w_j
\end{aligned}$$

and hence (5.4) holds. ◇

5.5 Existence of Population Monotonic Allocation Schemes for the Weighted Minimum Colouring Games

An open question for further research is to characterise the existence of PMASs for WMC games. We conclude this chapter with some remarks on this question.

Let G be globally (respectively, locally) WMC population monotonic if for all positive integer weight vectors w (respectively, for at least one positive integer weight vector w) the corresponding WMC game allows a PMAS. Firstly, we readily know from the result of Hamers et al. (2011) that for $w_i = 1$ for all $i \in N$, a graph G induces a WMC game that allows a PMAS if and only if it is $(2K_2, P_4)$-free. Therefore, the class of globally WMC population monotonic graphs cannot be larger than the class $(2K_2, P_4)$-free graphs. Furthermore, the class of globally WMC population monotonic graphs cannot be smaller than the class of complete r-partite graphs since we have already shown that complete r-partiteness is equivalent to global WMC submodularity, which in turn implies the existence of a PMAS for a WMC game induced by a complete r-partite graph for all positive integer weight vectors. Finally, we have the following example illustrating that locally WMC population monotonic graphs are not necessarily restricted to be

$(2K_2, P_4)$-free.

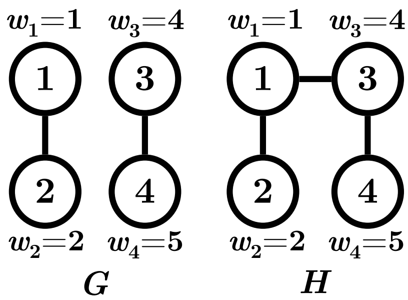


Figure 5.10: Graphs G and H, and the weight vector w.

Example 5.13. Consider the graphs G and H, and the weight vector w displayed in Figure 5.10. Graphs G and H are isomorphic to $2K_2$ and P_4, respectively. Let $N = \{1, 2, 3, 4\}$. A PMAS for the WMC game $(N, c^{G,w})$ and for the WMC game $(N, c^{H,w})$ are displayed in Table 5.4 and 5.5, respectively.

S	1	2	3	4
$\{1,2,3,4\}$	0	0	4	5
$\{1,2,3\}$	0	0	4	$*$
$\{1,2,4\}$	0	0	$*$	5
$\{1,3,4\}$	0	$*$	4	5
$\{2,3,4\}$	$*$	0	4	5
$\{1,2\}$	1	2	$*$	$*$
$\{1,3\}$	0	$*$	4	$*$
$\{1,4\}$	0	$*$	$*$	5
$\{2,3\}$	$*$	0	4	$*$
$\{2,4\}$	$*$	0	$*$	5
$\{3,4\}$	$*$	$*$	4	5
$\{1\}$	1	$*$	$*$	$*$
$\{2\}$	$*$	2	$*$	$*$
$\{3\}$	$*$	$*$	4	$*$
$\{4\}$	$*$	$*$	$*$	5

Table 5.4: A PMAS for the WMC game $(N, c^{G,w})$.

S	1	2	3	4
$\{1,2,3,4\}$	1	0	4	4
$\{1,2,3\}$	1	0	4	$*$
$\{1,2,4\}$	1	0	$*$	4
$\{1,3,4\}$	1	$*$	4	4
$\{2,3,4\}$	$*$	0	4	5
$\{1,2\}$	1	2	$*$	$*$
$\{1,3\}$	1	$*$	4	$*$
$\{1,4\}$	1	$*$	$*$	4
$\{2,3\}$	$*$	0	4	$*$
$\{2,4\}$	$*$	0	$*$	5
$\{3,4\}$	$*$	$*$	4	5
$\{1\}$	1	$*$	$*$	$*$
$\{2\}$	$*$	2	$*$	$*$
$\{3\}$	$*$	$*$	4	$*$
$\{4\}$	$*$	$*$	$*$	5

Table 5.5: A PMAS for the WMC game $(N, c^{H,w})$.

$\diamond$

Concluding Remarks

The thesis concludes with a discussion on a number of further results we have tried to obtain for SPT problems and games as well as possible directions for further research for both SPT and WMC games.

We start with the γ rule that we have introduced in Chapter 2. As we have mentioned earlier we have come up with this novel allocation rule and showed many desirable properties of this rule. We have also made a considerable effort to axiomatically characterise this allocation rule due to the important and central role that the axiomatic characterisations play in the design of allocation rules. After thoroughly researching the literature on the properties of cost allocations rules and consulting with other researchers in this area, we have proved that this allocation rule satisfies many interesting properties but we could not identify a set of rules that uniquely define this rule. Nonetheless, this study has led to the axiomatic characterisation of the tree solution as well as to the alternative definitions of the γ rule as we have discussed in Chapter 2.

In Chapter 3, we have discussed our results on the polyhedral analysis of the core of the SPT games. Previously, the polyhedral analysis of the core of cooperative games has mainly concentrated on extreme point and marginal vectors. With the help of the special polyhedral analysis package PORTA, we have firstly determined the dimension of the core of the SPT games. We have furthermore identified a class of extreme points of the core of the SPT games and a class of facets of the core of the SPT games that correspond to an SPT problem with a unique optimal solution. In light of these initial interesting results, we have performed numerous simulations to identify more facets and extreme points. However, there did not exist any other classes of facets or extreme points that were always present irrespective of the costs of the edges of the graph in consideration. Furthermore, it was not possible to extend our result on the facets to the multiple optimality case since there did not appear to exist a particular class of facets for the multiple optimal case in our simulations.

One of the open problems of the thesis is concerned with the nucleolus of the SPT games. Chapter 4 has presented our computational results on finding the nucleolus of the SPT games for the application of Wireless Multihop Networks. Recall that Algorithm 3 uses constraint generation to find the nucleolus. An issue with this approach is the hardness of solving the mixed integer programme (MIP) to identify the violated constraints. Especially, when the number of players gets larger our MIP will become harder to solve. One possible way of overcoming this problem would be devising a heuristic to solve this MIP. Such a heuristic would overcome this issue by either eliminating or reducing the size of any MIP that need to be solved. Note that we have made some initial investigations into this area and found out that due to the special structure of our MIP (specifically due to the structure of the constraints (4.15) and (4.16) that ensure that we generate non-redundant coalitions) identifying such a heuristic is not a straightforward task. However, it is still an interesting research area that is worth exploring further since it will have a major impact on the size of the problems for which we can identify the nucleolus of the SPT games.

Another research direction worth pursuing on WMC games, which we have introduced in Chapter 5, is the characterisation of the existence of PMASs for this class of games. We have started some investigations into this open question. Recall that for Theorem 5.1and Theorem 5.3 where we have characterised total balancedness and submodularity of WMC games for a graph G and all positive integer weight vectors respectively, the 'if'-parts of our proofs have extended the results of the unweighted minimum colouring games. Our initial results have indicated that proving the existence of PMASs for WMC games would not be a straightforward task since the results of Hamers et al. (2011) on PMASs of unweighted minimum colouring games did not appear to extend to the weighted case. Nonetheless, a possible approach to this open problem would be to devise special algorithms that would always generate PMASs for WMC games defined on certain graph classes.

Bibliography

Bachrach, Y., Markakis, E., Procaccia, A. D., Resnick, E., Rosenschein, J. S., Saberi, A., 2010. Approximating power indices. Journal of Autonomous Agents and Multiagent Systems 20(2), 105–122.

Bergantiños, G., Kar, A., 2010. On obligation rules for minimum cost spanning tree problems. Games and Economic Behavior 69(2), 224–237.

Bergantiños, G., Vidal-Puga, J., 2004. Defining rules in cost spanning tree problems through the canonical form. Game Theory and Information 0402004, EconWPA.

Bergantiños, G., Vidal-Puga, J., 2007. A fair rule in minimum cost spanning tree problems. Journal of Economic Theory 137(1), 326–352.

Bird, C. G., 1976. On cost allocation for a spanning tree: a game theoretic approach. Networks 6(4), 335–350.

Bjøndal, E., Koster, M., Tijs, S., 1999. Weighted allocation rules for standard fixed tree games. Discussion Paper No. 1999-79, Tilburg University, CentER.
URL `http://arno.uvt.nl/show.cgi?fid=3930`

Bondareva, O. N., 1963. Some applications of linear programming methods to the theory of cooperative games (in russian). Problemy Kybernetiki 10, 119–139.

Borm, P., Hamers, H., Hendrickx, R., 2001. Operations research games: A survey. TOP 9(2), 139–199.

Branzei, R., Moretti, S., Norde, H., Tijs, S., 2004. The p-value for cost sharing in minimum cost spanning tree situations. Theory and Decision 56, 47–61.

Çiftçi, B., Borm, P., Hamers, H., 2010. Highway games on weakly cyclic graphs. European Journal of Operational Research 204(1), 117–124.

Claus, A., Kleitman, D. J., 1973. Cost allocation for a spanning tree. Networks 3(4), 289–382.

Curiel, I., 1997. Cooperative game theory and applications: cooperative games arising from combinatorial optimization problems. Kluwer Academic Publishers.

Deng, X., Ibaraki, T., Nagamochi, H., 1999. Algorithmic aspects of the core of combinatorial optimization games. Mathematics of Operations Research 24(3), 751–766.

Deng, X., Ibaraki, T., Nagamochi, H., Zang, W., 2000. Totally balanced combinatorial optimization games. Mathematical Programming 87, 441–452.

Dijkstra, E. W., 1959. A note on two problems in connexion with graphs. Numerische Mathematik 1(1), 269–271.

Elkind, E., Pasechnik, D., 2009. Computing the nucleolus of weighted voting games. SODA 37, 327–335.

Faigle, U., Kern, W., Kuipers, J., 1998. Computing the nucleolus of min-cost spanning tree games is np-hard. International Journal of Game Theory 27, 443–450.

Feltkamp, V., Tijs, S., Muto, S., 1994a. Minimum cost spanning extension problems: the proportional rule and the decentralized rule. Discussion Paper No. 1994-96, Tilburg University, CentER.
URL `http://arno.uvt.nl/show.cgi?fid=3002`

Feltkamp, V., Tijs, S., Muto, S., 1994b. On the irreducible core and the equal remaining obligations rule of minimum cost spanning extension problems. Discussion Paper No. 1994-106, Tilburg University, CentER.
URL `http://arno.uvt.nl/show.cgi?fid=3020`

Fragnelli, V., García-Jurado, I., Méndez-Naya, L., 2000. On shortest path games. Mathematical Methods of Operational Research 52(2), 251–264.

Fromen, B., 1997. Reducing the number of linear programs needed for solving the nucleolus problem of n-person game theory. European Journal of Operational Research 98, 626–636.

Gebreiter, F., 2011. Making medicine calculable: Hospital costing between the art and science of medicine. Ph.D. thesis, London School of Economics and Political Science, Department of Accounting.

Gillies, D. B., 1953. Some theorems on n-person games. Ph.D. thesis, Princeton University, Department of Mathematics.

Granot, D., Granot, F., 1992. Computational complexity of a cost allocation approach to a fixed cost spanning forest problem. Mathematics of Operations Research 17, 765–780.

Granot, D., Granot, F., Zhu, W. R., 1998. Characterisation sets for the nucleolus. International Journal of Game Theory 27, 359–374.

Granot, D., Granot, F., Zhu, W. R., 2000. Naturally submodular digraphs and forbidden digraph configurations. Discrete Applied Mathematics 100(1-2), 67–84.

Granot, D., Hamers, H., 2004. On the equivalence between some local and global chinese postman and traveling salesman graphs. Discrete Applied Mathematics 134(1-3), 67–76.

Granot, D., Hamers, H., Tijs, S., 1999. On some balanced, totally balanced and submodular delivery games. Mathematical Programming 86(2), 355–366.

Granot, D., Huberman, G., 1981. Minimum cost spanning tree games. Mathematical Programming 21(1), 1–18.

Granot, D., Huberman, G., 1984. On the core and the nucleolus of the minimum cost spanning tree games. Mathematical Programming 29(3), 323–347.

Granot, D., Maschler, M., 1998. Spanning network games. International Journal of Game Theory 27, 467–500.

Granot, D., Maschler, M., Owen, G., Zhu, W. R., 1996. The kernel/nucleolus of a standard tree game. International Journal of Game Theory 25, 219–244.

Grötschel, M., Lovász, L., Schrijver, A., 1988. Geometric algorithms and combinatorial optimization. Springer-Verlag.

Hallefjord, Å., Helming, R., Jørnsten, K., 1995. Computing the nucleolus when the characteristic function is given implicitly: A constraint generation approach. International Journal of Game Theory 24(4), 357–372.

Hamers, H., Borm, P., van de Leensel, R., Tijs, S., 1999. Cost allocation in the chinese postman problem. European Journal of Operational Research 118(1), 153–163.

Hamers, H., Miquel, S., Norde, H., 2011. Monotonic stable solutions for minimum coloring games. Discussion Paper No. 2011-016, Tilburg University, CentER.
URL `http://arno.uvt.nl/show.cgi?fid=113917`

Herer, Y. T., Penn, M., 1995. Characterizations of natural submodular graphs: A polynomially solvable class of the TSP. Proceedings of the American Mathematical Society 123(3), pp. 673–679.

Ichiishi, T., 1981. Super-modularity: Applications to convex games and to the greedy algorithm for lp. Journal of Economic Theory 25(2), 283–286.

Kannai, Y., 1992. Chapter 12 the core and balancedness. In: Aumann, R., Hart, S. (Eds.), Handbook of Game Theory with Economic Applications. Vol. 1. Elsevier, pp. 355 – 395.

Kopelowitz, A., 1967. Computation of the kernels of simple games and nucleolus of n-person games. Tech. Rep. RM-31, The Hebrew University of Jerusalem, Math. Dept.

Koster, M., Molina, E., Sprumont, Y., Tijs, S., 2001. Sharing the cost of a network: core and core allocations. International Journal of Game Theory 30, 567–599.

Kruskal, J., 1956. On the shortest path spanning subtree of a graph and the travelling salesman problem. Proceedings of the American Mathematical Society 7, 48–50.

Kwan, M. K., 1963. Graphic programming using odd or even points. Chinese Mathematics 118(1), 273–277.

Lovász, L., 1972. Normal hypergraphs and the perfect graph conjecture. Discrete Mathematics 2(3), 253 – 267.

Makki, S., Li, X., Pissinou, N., 2008. Sensor and ad hoc networks: theoretical and algorithmic aspects. Springer.

Maschler, M., Peleg, B., Shapley, L. S., 1979. Geometric properties of the kernel, nucleolus, and related solution concepts. Mathematics of Operations Research 4(4), 303–338.

Maschler, M., Potters, J., Reijnierse, H., 1995. Monotonicity properties of the nucleolus of a standard tree game. Report 9556, University of Nijmegen, Department of Mathematics.

McDiarmid, C., Reed, B., 2000. Channel assignment and weighted coloring. Networks 36(2), 114–117.

Megiddo, N., 1978. Computational complexity of the game theory approach to cost allocation for a tree. Games and Economic Behavior 3(3), 189–196.

Meinhardt, H. I., 2002. Cooperative decision making in common pool situations. Springer.

Miquel, S., van Velzen, S., Hamers, H. J. M., Norde, H. W., 2006. Fixed tree games with repeated players. Networks 47(2), 93–101.

Moretti, S., Tijs, S., Branzei, R., Norde, H., 2005. Cost monotonic 'construct and charge' rules for connection situations. Discussion Paper No. 2005-104, Tilburg University, CentER.
URL `http://arno.uvt.nl/show.cgi?fid=53801`

Mosquera, M. A., 2007. Essays on operations research games and cautious behaviour. Ph.D. thesis, Universidade de Santiago de Compostela.

Moulin, H., 1988. Axioms of cooperative decision making. Cambridge University Press.

Nemhauser, G. L., Wolsey, L. A., 1988. Integer and combinatorial optimization. Wiley-Interscience Series in Discrete Mathematics and Optimization.

Norde, H., Moretti, S., Tijs, S., 2004. Minimum cost spanning tree games and population monotonic allocation schemes. European Journal of Operational Research 154, 84–97.

Okamoto, Y., 2003. Submodularity of some classes of the combinatorial optimization games. Mathematical Methods of Operations Research 58(1), 131–139.

Prim, R., 1957. Shortest connection networks and some generalizations. Bell Systems Technical Journal 36, 1389–1401.

Quant, M., Borm, P., Reijnierse, H., 2006. Congestion network problems and related games. European Journal of Operational Research 172, 919–930.

Schmeidler, D., 1969. The nucleolus of a characteristic function game. SIAM Journal of Applied Mathematics 17, 1163–1170.

Schrijver, A., 2003. Combinatorial Optimization. Springer-Verlag.

Shapley, L. S., 1953. A value for n-person games. Annals of Mathematics Studies 28, 307–317.

Shapley, L. S., 1967. On balanced sets and cores. Naval Research Logistics Quarterly 14, 453–460.

Shapley, L. S., 1971. Cores of convex games. International Journal of Game Theory 1(1), 11–26.

Solymosi, T., Raghavan, T. E. S., 1994. An algorithm for finding the nucleolus of assignment games. International Journal of Game Theory 23, 119–143.

Spira, P. M., Pan, A., 1973. On finding and updating shortest paths and spanning trees. In: IEEE Conference Record of 14th Annual Symposium on Switching and Automata Theory. pp. 82 – 84.

Sprumont, Y., 1990. Population monotonic allocation schemes for cooperative games with transferable utility. Games and Economic Behavior 2(4), 378–394.

Thomson, W., 2007. Fair allocation rules. Working Paper No. 539, University of Rochester, Rochester Center for Economic Research.
URL `http://rcer.econ.rochester.edu/RCERPAPERS/rcer_539.pdf`

Tijs, S., 1981. Bounds for the core and the value. In: Moeschlin, O., Pallaschke, D. (Eds.), Game Theory and Mathematical Economics. North-Holland, pp. 123–132.

Tijs, S., Branzei, R., Moretti, S., Norde, H., 2006. Obligation rules for minimum cost spanning tree situations and their monotonicity properties. European Journal of Operational Research 175(1), 121–134.

Tijs, S. H., Driessen, T. S. H., 1986. Game theory and cost allocation problems. Management Science 32(8), 1015–1028.

van Gellekom, J. R. G., Potters, J. A. M., 1999. Consistent solution rules for standard tree enterprises. Report 9919, University of Nijmegen, Department of Mathematics. URL `http://repository.ubn.ru.nl/bitstream/2066/18725/1/18725.pdf`

Voorneveld, M., Grahn, S., 2002. Cost allocation in shortest path games. Mathematical Methods of Operational Research 56, 323–340.

Wölk, E. S., 1965. A note on "the comparability graph of a tree". Proceedings of the American Mathematical Society 16, 17–20.

Wolsey, L. A., 1998. Integer programming. Wiley-Interscience Series in Discrete Mathematics and Optimization.

Yan, J., Chen, J., Chang, G. J., 1996. Quasi-threshold graphs. Discrete Applied Mathematics 69, 247–255.

Young, H. (Ed.), 1985. Cost Allocation: Methods, Principles, Applications. North-Holland.

Index

List of Figures

List of Tables