Matchings and games on networks

by

Linda Farczadi

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Combinatorics and Optimization

Waterloo, Ontario, Canada, 2015

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

We investigate computational aspects of popular solution concepts for different models of network games. In chapter 3 we study balanced solutions for network bargaining games with general capacities, where agents can participate in a fixed but arbitrary number of contracts. We fully characterize the existence of balanced solutions and provide the first polynomial time algorithm for their computation. Our methods use a new idea of reducing an instance with general capacities to an instance with unit capacities defined on an auxiliary graph. This chapter is an extended version of the conference paper [32].

In chapter 4 we propose a generalization of the classical stable marriage problem. In our model the preferences on one side of the partition are given in terms of arbitrary binary relations, that need not be transitive nor acyclic. This generalization is practically well-motivated, and as we show, encompasses the well studied hard variant of stable marriage where preferences are allowed to have ties and to be incomplete. Our main result shows that deciding the existence of a stable matching in our model is NP-complete. We then use our model to study a long standing open problem about cyclic 3D stable matchings. In particular, we prove that deciding whether a fixed 2D perfect matching can be extended to a 3D stable matching is NP-complete. This chapter is an extended version of the conference paper [33].

In chapter 5 we study a long standing open problem of whether the nucleolus of matching games can be computed efficiently. Our approach follows previous techniques that rely on obtaining a polynomial sized characterization of the least core as both the initial and crucial step in establishing an efficient algorithm. As a preliminary result, we introduce a generalisation of the least core and show that for node-weighted matching games existing polynomial sized characterizations of the least core can be extended to this generalised version. We then use this result to show that for a certain class games with general weights one can identify a node-weighted subgraph such that the least core of the original matching game is equal to the generalised least core of this subgraph. This allows us to obtain a polynomial time algorithm for computing the nucleolus for this class of matching games with general weights.

Acknowledgements

This thesis would not have been possible without the guidance and support of my advisor Jochen Könemann. I am grateful for his constant confidence in my abilities and for allowing me to work on problems that I found both exciting and challenging. His optimism and enthusiasm for research have been contagious over the years. I am also indebted to Costis Georgiou, Umang Bhaskar, Bernhard von Stengel, Laura Sanità, Chaitanya Swamy and Kate Larson. Their input has been invaluable to the success of this thesis.

Throughout my studies I was fortunate to encounter several people whose support I have tremendously benefited from. I am grateful to Navin Kashyap for introducing to research and encouraging me to pursue graduate studies. I would also like to thank Luc Devroye for all his guidance during my masters studies that has greatly contributed to my development as a researcher.

My time in Waterloo has been enriched by the wonderful friendships that I have formed here. I am forever thankful to all my friends that have shared this journey with me, and have made the last four years truly memorable. I would also like to thank the wonderful C&O staff for all their help along the way.

This research has been supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

To my parents Gina and Lajos

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

This thesis studies the computational complexity of some central solution concepts in network games. Networks are prevalent in every aspect of our lives. For example, in the last decades social networks have proven to be essential to understanding human behaviour. Perhaps not surprisingly, a vast majority of decisions that people make from which products to buy, to which political candidate to support, are closely tied to the choices of their friends. As a result, social interactions and the spread of information have fundamental effects on the outcomes of these decisions. In turn, these interactions are often governed by the underlying structure of a network, a system of bilateral relationships between participants. The study of such systems is therefore a crucial step in better understanding the outcomes of human interactions.

In the game theoretic approach to networks, the inputs to the problem are assumed to be controlled by rational, self-interested agents. This gives rise to what we refer to as network games: situations where agents who are connected via a network interact with one another to achieve an outcome of mutual benefit. In its simplest form, a network game is described by a graph, where the vertices of the graph represent the agents and the edges are the bilateral links among them. These links capture the way that the agents can interact to derive some mutual benefit. As one would expect, the benefit derived by a given agent is heavily dependent on those of his or her neighbours in the network. Hence, the underlying graph-theoretical properties of the network become instrumental in determining exactly what type of outcomes one can expect.

Network games can also be viewed as a special case of cooperative games. Cooperative game theory studies situations where agents can cooperate with one another by entering in some binding agreements with each other to derive mutual benefit. In this thesis we are particularly interested in transferable utility (TU) games, a type of cooperative games where the value derived from cooperation can be transferred among the agents without the use of a third entity. The main goals of such games is to determine exactly which groups, or coalitions, of agents decide to cooperate and how they divide the total value achieved by cooperation. In particular, a fundamental question is to determine how the agents will divide the value fairly. In order to answer these questions, game theorists have devised several solution concepts. A solution concept identifies a subset of rational outcomes which are most likely to occur. One of the most widely accepted solution concepts for cooperative games is that of stability. An outcome is said to be stable if no member of the forming coalition has an incentive to deviate from it.

In addition to capturing rational outcomes, it is essential for solution concepts to also have desirable computational properties. In particular, it is important to determine whether a certain game always possesses a rational outcome according to a given solution concept, and whether such an outcome can always be computed efficiently. This thesis is concerned with answering exactly these type of questions. In the remainder of this chapter we present the three main type of network games considered in this thesis together with an overview of our main contributions for each game. We conclude with an outline of the remaining chapters of the thesis.

1.1 Network bargaining

Exchanges in networks have been studied for a long time in both sociology and economics. In sociology, they appear under the name of network exchange theory, a field which studies the behaviour of agents who interact across a network to form bilateral relationships of mutual benefit. The goal is to determine how an agent's location in the network influences its ability to negotiate for resources [15]. In economics, they are known as cooperative games [69] and have been used for studying the distribution of resources across a network, for example in the case of two-sided markets [75] [71].

From a theoretical perspective the most commonly used framework for studying such exchanges is that of network bargaining games. As a simple example consider the case of two agents A and B who can choose to form a contract of value w dollars. Given no other information about the agents,and assuming equal utilities, we would expect them to form the contract and divide its value equally among themselves so that each agents receives $w/2$ dollars as depicted in Figure 1.1.

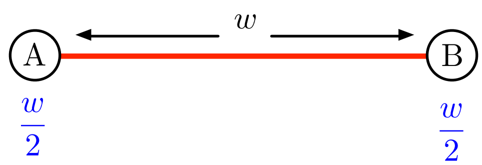

Figure 1.1: Splitting a contract with no outside options.

But now suppose that we are told the agents have the following backup options: if they decide not to form the contract, then agent A can receive a total value of α_A dollars while agent B can receive a total value of α_B dollars. How will this effect the negotiations between the two agents? John Nash first proposed a solution to this scenario in his paper [67] that became the foundation of modern bargaining theory. His idea, referred to as the Nash bargaining solution, states that the contract is formed if and only if its value is at least as much as $\alpha_A + \alpha_B$, the sum of the backup options of the two agents. Furthermore, if the contract is formed then its value is divided according to the following rule: each agent receives an amount equal to its backup option, and any leftover amount is split equally between the two agents. The resulting split is depicted in Figure 1.2.

The idea of this two person Nash bargaining solution can be generalized to any number of agents, where each agent can engage in a fixed number of contracts specified by its capacity. This general model is described in terms of an undirected graph $G = (V, E)$ with edge weights $w : E(G) \to \mathbb{R}_+$ and vertex capacities $c : V(G) \to \mathbb{Z}_+$. The vertices represent the agents, and the edges represent pairwise contracts that the agents can form.

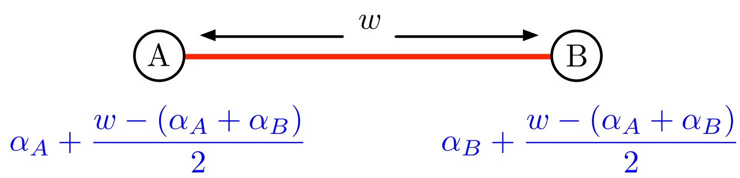

Figure 1.2: Splitting a contract with existing outside options.

The weight of each edge represents the value of the corresponding contract. As in the two person scenario, if a contract is formed between two agents, its value is divided between them, whereas if the contract is not formed neither agent receives any profit from this specific contract. The capacity of each agent limits the number of contracts it can form. This constraint, together with an agent's position in the network determine its bargaining power. The main questions that network bargaining tries to answer is which contracts are formed and how their value is divided among agents forming the contract.

A solution for the network bargaining model specifies the set of contracts which are formed, and how each contract is divided. Specifically, a solution consists of a pair (M, z), where M is a c-matching of the underlying graph G, and z is a vector which assigns each edge (u, v) two values $z_{uv}, z_{vu} \geq 0$ corresponding to the profit that agent u, respectively agent v, earn from the contract (u, v). To be a valid solution, the two values z_{uv} and z_{vu} must add up to the value of the contract whenever the edge (u, v) belongs to the c-matching M, and must be zero otherwise.

Solutions to network bargaining games are classified according to two main concepts: stability and balance. A solution is stable if the profit an agent earns from any formed contract is at least as much as its outside option: an agent's outside option, in this context, refers to the maximum profit that the agent can rationally receive by forming a new contract with one of its neighbours, under the condition that the newly formed contract would benefit both parties. The notion of balance, first introduced in [14], [71], is a generalization of the Nash bargaining solution to the network setting. Specifically, in a balanced solution the value of each contract is split according to the following rule: both endpoints must earn at least their outside options, and any surplus is to be divided equally among them. Balanced solutions have been shown to agree with experimental evidence, even to the point of picking up on subtle differences in bargaining power among agents [80, 11]. This is an affirmation of the fact that these solutions are natural and represent an important area of study.

As an example, consider the graph shown in Figure 1.3. There are five agents in the graph denote by the vertices A, B, C, D and E. Each agent has capacity one and can engage in at most one contract and the edges are all unit weight. One might immediately notice that in order to maximize their total profit, the agents should choose to form contracts corresponding to a maximum weight matching, that is either $\{AB, DE\}$ or $\{BC, DE\}$. This intuition turns out to be true, as any stable solution in a network bargaining game must always occur on a maximum weight matching. The solution shown in Figure 1.3

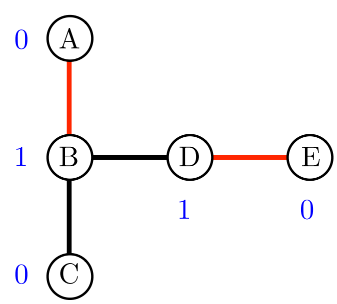

Figure 1.3: A stable but not balanced solution.

awards the entire weight of the contract AB to the vertex B, and the entire weight of the contract DE to the vertex D. This turns out to a stable solution since all vertices that receive a value of 0 always have an outside option of 0, so no stability conditions are violated. However, this solution is not balanced, since both vertices D and E have an outside option of 0, and the Nash bargaining solution would dictate that they split the value of their contract equally among them. The unique balanced solution to this game is shown in Figure 1.4. Intuitively, vertex B receives the full value of its contract with A due to its central location in the network. As for the contract between D and E, even though E is an endpoint vertex and has no bargaining power, D's only other connection is with a very powerful vertex and thus its bargaining power is no greater then E's.

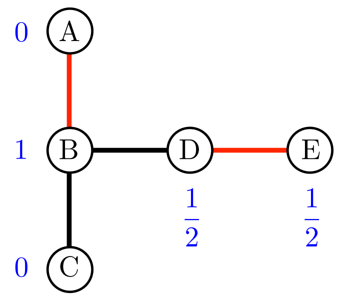

Figure 1.4: A stable and balanced solution.

There is a close connection between network bargaining games and cooperative games. Specifically given a solution (M, z) to the network bargaining game (G, w, c) we can define a corresponding payoff vector x, where x_u is just the total profit earned by vertex u from all its contracts in the solution (M, z). Then this vector x can be seen as a solution to a corresponding cooperative game (N, v) defined as follows: we let $N = V(G)$ denote the set of players, and for every subset $S \subseteq N$ of players, we define its value $v(S)$ as the weight of the maximum weight c-matching of $G[S]$. This game is also known as the matching game of Shapley and Shubik [75]. The subsets $S \subseteq N$ are referred to as coalitions, and the value $v(S)$ of each coalition is interpreted as the collective payoff that the players in S would receive if they decide to cooperate. The players are assumed to be able choose which coalitions to form, and their objective is to maximize their pay-offs.

Cooperative games are often assumed to be super-additive, meaning that the value of a union of disjoint coalitions is no less than the sum of the coalitions' separate values. The underlying assumption in such games is that the grand coalition N will form, and the question becomes how to distribute the payoff $v(N)$ among the players. A vector x describing such a distribution is referred to as an allocation. Given an allocation x, the excess of a coalition S is defined as $v(S) - x(S)$. The power of player u over player v with respect to the allocation x is the maximum excess achieved by a coalition that includes u but excludes v. Two important concepts from cooperative games are those of the core and prekernel. The core consists of allocations for which no coalition has a negative excess, whereas the prekernel consists of all allocations with symmetric powers.

1.1.1 Contributions

Our main result is providing the first polynomial time algorithm for computing balanced solutions for network bargaining games with general capacities and fully characterizing the existence of balanced solutions for these games. Specifically we show the following results.

Result 1. *There exists a polynomial time algorithm which, given an instance of a network bargaining game with general capacities and a maximum weight c-matching M, computes a balanced solution (M, z) whenever one exists.*

Result 2. *A network bargaining game with general capacities has a balanced solution if and only if it has a stable one.*

Our method relies on a new approach of reducing a general capacity instance to a network bargaining game with unit capacities defined on an auxiliary graph. This allows us to use existing algorithms for obtaining balanced solutions for unit capacities games, which we can then transform to balanced solutions of our original instance. This represents a departure from previous approaches of Bateni et al. [6] who were trying to establish an equivalence between the set of balance solutions and allocations in the intersection of the core and prekernel of the corresponding matching game. We show that such an approach cannot work in general, since this equivalence does not extend to all instances of general capacity games:

Result 3. *There exists an instance of a network bargaining game with general capacities for which we can find an allocation in the intersection of the core and prekernel such that there is no corresponding balanced solution for this allocation.*

Despite this result, we provide two necessary conditions which ensure that the correspondence between the set of balanced solutions and allocations in the intersection of the core and prekernel is maintained. Using the definition of gadgets we have the following result:

Result 4. *If the network bargaining game has no gadgets and the maximum c-matching M is acyclic, the set of balanced solutions corresponds to the intersection of the core and prekernel.*

1.1.2 Related work

Kleinberg and Tardos [55] introduced network bargaining games to the theoretical computer science community. They studied games with unit capacities and developed a polynomial time algorithm for computing the entire set of balanced solutions. They also showed that such games have a balanced solution whenever they have a stable one and that a stable solution exists if and only if the linear programming relaxation for the maximum weight matching of the underlying graph has an integral optimal solution. Here we are referring to the linear program containing only the degree constraints and not the odd set constraints.

Bateni et al. [6] were the first to generalize network bargaining games to arbitrary capacities. They characterized the existence of stable solutions by providing a parallel result to the case of unit weights: stable solutions exist if and only if the linear programming relaxation for the maximum weight c-matching (again without the odd set constraints) of the underlying graph has an integral optimal solution. For balanced solutions, they focused on a special case of bipartite graphs where one side of the partition has unit capacities but the other side can have arbitrary capacities. They proved an equivalence between the set of balanced solutions of this special class of graphs and the intersection of the core and prekernel of the corresponding matching game. This equivalence implies that balanced solutions can be computed using an existing polynomial time algorithm of Faigle [31] for finding allocations in the prekernel.

Network bargaining games with general capacities have also been studied by Kanoria et al. [51] who focused on the existence of a natural local dynamics for this game. They also provided a partial characterization of the existence of balanced solutions by proving that if the associated linear programming relaxation has a unique optimum solution that is integral, then a balanced solution is guaranteed to exist. They provide an algorithm for computing balanced solutions in this case which uses local dynamics but whose running time is exponential.

Local dynamics in network bargaining games with unit capacities are also studied by Azar et al. [3] who focus on a different local dynamics that assumes the maximum weight matching M is fixed and proceeds by iteratively fixing edges that do not satisfy the Nash bargaining solution. They show that this edge balancing dynamics converges to a balanced outcome whenever one exists. However the rate of convergence might be exponential. In [10] the authors analyse the same local dynamics using a new technique that relies on a connection to random turn games. They are able to improve the previous exponential bound on convergence to a polynomial one for the case of network bargaining games with unit weights. Independently from these works, Draief et al. [21] showed quadratic convergence of the edge balancing dynamics for the following graphs: a path, a cycle, a blossom and a bicycle.

For general cooperative games, Maschler proposed a simple transfer scheme to approximate points in the prekernel. Stearns [78] was the first to prove convergence of this scheme, and a simpler proof was provided by Faigle et al. [31]. In the same paper, Faigle et al. provide a polynomial time algorithm for finding a point in the intersection of the least core and prekernel that uses two components: a transfer scheme similar to Maschler's and a linear programming based update that guarantees the polynomial running time.

1.2 Stable matchings

The Stable Marriage (SM) problem is a classical bipartite matching problem first introduced by Gale and Shapley [36]. An instance of the problem consists of a set n of men, and a set of n women. Each man (woman) has a preference list that is a total order over the entire set of women (men). The goal is to find a stable matching between the men and women, meaning that there is no (man, woman) pair so that both prefer each other to their current partners in the matching. Since its introduction, the stable marriage problem has become one of the most popular combinatorial problems with several books being dedicated to its study [42, 56, 72] and more recently [61]. The popularity of this model arises not only from its nice theoretical properties but also from its many applications. In particular, a wide array of allocation problems from many diverse fields can be analysed within its context. Some well known examples include the labour market for medical interns, auction markets, the college admissions market, the organ donor-recipient pair market, and many more [72].

In their seminal work Gale and Shapley showed that every instance of SM admits a solution and such a solution can be computed efficiently using the so-called Gale-Shapley (or man-proposing) algorithm. Among the many new variants of this classical problem, two extensions have received most of the attention: incomplete preference lists and ties in the preferences. Introducing either one of these extensions on its own does not pose any new challenges, meaning that solutions are still guaranteed to exist, all solutions have the same size, and they can be computed efficiently using a modification of the original Gale-Shapley algorithm [37, 42]. However, the same cannot be said about the Stable Marriage problem with Ties and Incomplete Lists (SMTI) that incorporates both extensions. In this variant, stable matchings no longer need to be of the same size, even though they are still guaranteed to exist. To see this consider the example depicted in Figure 1.5. The edges in the graph denote all acceptable pairs and the numbers on the edges near a vertex denote the ranking of each partner with respect to that vertex. Man b_1 and woman c_2 only have one acceptable partner, whose ranking is by default 1. Woman c_1 finds both b_1 and b_2 acceptable and she is indifferent between them, since they both have rank 1. Man b_2 finds both women c_1 and c_2 acceptable but prefers woman c_1 over woman c_2. In the figure on the left the red edges b_1c_1 and b_2c_2 denote a stable matching of size two, while in the figure on the right the red edge b_2c_1 denotes a stable matching of size one.

Since SMTI instances poses solutions of different sizes, the natural question to ask is whether one can efficiently compute a stable matching of maximum size. The answer to this question is no, since deciding whether a given instance admits a stable matching of a given size is NP-hard [62], even in the case where ties occur only on one side of the partition. When ties occur in both the preferences of the men and women the problem is NP-hard to approximate to within $33/29 (> 1.1379)$ and UGC-hard to approximate to within $4/3$ [?]. The current best approximation ratio is $3/2$ [54]. When ties occur only in the preference lists of the women, the problem is NP-hard to approximate to within $21/19 (> 1.1052)$

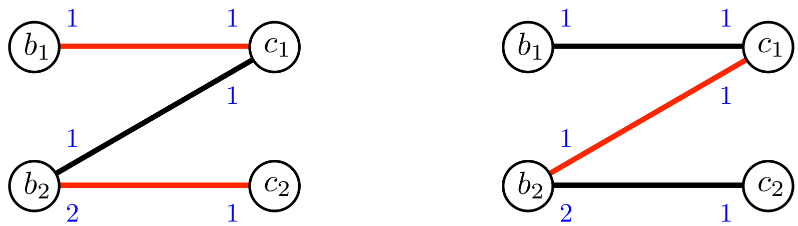

Figure 1.5: An SMTI instance with stable matchings of different size

and UGC-hard to approximate to within 5/4 [48]. The current best approximation ratio is 25/17 [49]. We refer the reading to [61] for an excellent account of recent work on the approximation version of this problem.

A central assumption in most variants of SM is that agents' preferences are transitive (i.e., if x is preferred to y, and y is preferred to z then x is also preferred to z). However, there are several studies [7, 35, 50, 64] that suggest that non-transitive, and even cyclic preferences arise naturally. Cyclic preferences, for example, may be introduced in the context of multi-attribute comparisons [34]; e.g., consider the following study from [64] where 62 college students were asked to make binary comparisons between three potential marriage partners x,y and z according to the following three criteria: intelligence, looks and wealth. The candidates had the following attributes: candidate x was very intelligent, plain, and well off; candidate y was intelligent, very good looking, and poor; and candidate z was fairly intelligent, good looking, and rich. From the 62 participants, 17 displayed the following cyclic preference: x was preferred to y, y was preferred to z, and z was preferred to x. In order to better capture such situations there is a need for a model that allows for more general preferences.

Addressing this need we propose the Stable Marriage with General Preferences (SMG) problem. As in SM, in an instance of SMG we are given n men, and n women, and the preferences of men are complete total orders over the set of women. The preferences of women, on the other hand, are given in terms of arbitrary binary relations over the men. Each of these binary relations will be represented by a set $\mathcal{R}$ containing ordered pairs of men. We allow the set $\mathcal{R}$ to be empty. We say that a woman prefers man x at least as much as man y if the ordered pair (x, y) is part of her preference set. A matching is then stable as long as for every unmatched (man, woman) pair at least one member prefers her mate in the matching at least as much as the other member of the pair.

It is easy to see that any stable matching for an SMG instance must be a perfect matching. This contrasts the case of SMTI where a given instance can have stable matchings of different size. A second difference is that unlike in SMTI, not all instances of SMG admit a stable matching. As a simple example consider the instance of SMG depicted in Figure 1.6. This instance consists of two men b_1, b_2 and two women c_1, c_2. Both men prefer woman c_1 to woman c_2. However the set of ordered pairs representing woman c_1's preferences is empty, meaning that she does not prefer b_1 at least as much as b_2, nor does she prefer b_2 at least as much as b_1. Then given any perfect matching, woman c_1 will always form a blocking pair with the man that she is not matched to. Hence this instance does not admit a stable matching.

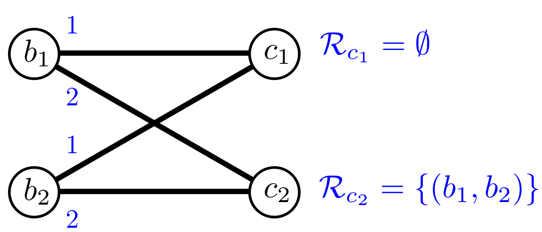

Figure 1.6: An SMG instance with no stable matching

Our SMG model can also be used to study a three-dimensional stable matching problem first proposed by Knuth [56]. In particular we consider the Cyclic 3-Dimensional Stable Matching problem (c3DSM), where we are given a set of n men, a set of n women and in addition a set of n dogs. The preferences of the men are complete total orders over the set of women. Similarly the women have preferences over the dogs, and the dogs have preferences over the men. A 3D matching is said to be stable if there is no (man, woman, dog) triple such that each member of the triple prefers this triple over the one assigned to them in the matching. A prominent open question is whether every instance of c3DSM admits a stable matching, and whether such a matching can be computed efficiently.

A natural avenue for attacking c3DSM is to solve the following problem which we refer to as Stable Extension (SE): suppose we fix a perfect matching M on the sets of dogs and men, can we efficiently determine whether M is extendible to a 3D stable matching? More specifically, we say that M is extendible to a 3D stable matching if there is a way of assigning to each man-dog pair in M a woman such that the resulting 3D matching is stable. Recall that women have preferences over dogs only, but note that the given matching M induces preferences over their male owners as well. In essence, this allows us to state the SE problem as a special case of the SMG model.

1.2.1 Contributions

Introducing non-transitive preferences, even when restricted to just one side of the partition, changes the properties of the model drastically. Like in SM any solution must be a perfect matching. However, solutions are no longer guaranteed to exist. We show that non-transitive preferences generalize both incomplete lists and ties by reducing the SMTI to SMG. In doing so, we prove that the SMG problem is also NP-hard.

Result 5. *The decision problem of whether an instance of* SMG *admits a solution is* NP*-complete.*

We then identify a significant class of instances that are solvable in polynomial time: those where the preferences are asymmetric, meaning that for every pair of men x, y, each woman prefers at most one to the other. We then prove the following result.

Result 6. *For instances of* SMG *with asymmetric preferences, there exists a polynomial time algorithm that finds a solution if and only if one exists.*

We provide two different proofs. The first employs an adaptation of the classical Gale-Shapley man-proposing algorithm. The second relies on a polyhedral characterization: we define a polytope that is non-empty if and only if the instance admits a stable matching. We also develop an efficient rounding algorithm for its fractional points.

Despite displaying stronger structural properties than SMG, we show that SE remains hard to solve.

Result 7. *The decision problem of whether an instance of* SE *admits a solution is* NP-*complete*

At a high level, the strategy for proving result 7 resembles that of the proof of result 5. The details are however significantly more intricate, mainly due to the fact that SE instances correspond to SMG instances in which preferences are induced by a given 3D matching instance. As an interesting consequence for the c3DSM, result 7 rules out the natural algorithmic strategy of fixing and extending a 2D perfect matching on two of the input sets. In addition our results for SMG instances with complete preferences provide a sufficient condition for when a perfect matching is extendable to a stable three dimensional matching.

1.2.2 Related work

To the best of our knowledge, the stable marriage problem with preferences given in terms of arbitrary binary relations has not been studied before. In [1] the authors do consider a version of SM with non-transitive preferences, however unlike in our model, the preference relations are required to be acyclic. The authors do not study this problem directly but instead use it as a tool for developing a reduction between the stable room-mates problem and the stable marriage problem.

There is a rich literature about SM and its variants. In particular, there has been significant work concerning the approximation variant of the SMTI problem, where the goal is to find a maximum size stable matching. When ties are allowed on both sides, the problem is NP-hard to approximate within 33/29 [**?**] and the currently best known ration is 3/2 [65]. When ties are only allowed on the side of the women the problem is NP-hard to approximate within 21/19 [43] and the currently best known ratio is 25/17 [49].

A related model, known as Stable Marriage with Indifference [45, 60], allows for preferences to be given in the form of partial orders that are not necessarily expressible as a single list involving ties. That is, the indifference relation need not be transitive. This model allows for several definitions of stability, and depending on which definition is used, solutions might not always exist. In particular, under strong stability, stable matchings might not always exists and it is show in [46] that the problem of deciding whether a strongly stable matching exists, given an instance of the stable marriage problem with partially ordered preferences, is NP-complete.

For c3DSM, it is known that every instance admits a stable matching for $n \leq 4$ [27]. The authors conjectured that this result can be extended to general instances. In [8] it was shown that if we allow unacceptable partners, the existence of a stable matching becomes NP-complete. In the same paper, and also independently in [44], it was shown that the c3DSM problem under a different notion of stability known as strong stability is also NP-complete.

1.3 Matching games

While the core is a very important solution concept for cooperative games, unfortunately many games have an empty core. This is particularly true for matching games, where the core is empty whenever the weight of the maximum weight fractional matching is strictly greater than the weight of the maximum weight integral matching. As an example consider the matching game displayed in Figure 1.7.

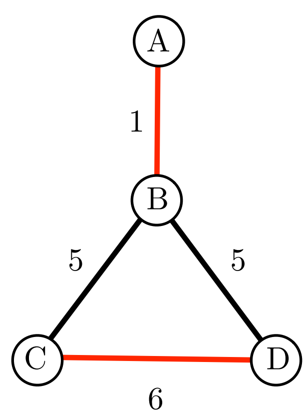

Figure 1.7: A matching game with empty core.

The red edges indicate the unique maximum weight matching with total weight 7. However, the maximum weight fractional matching achieves a total weight of 8 by assigning a value of $1/2$ to each edge of the triangle. It can be easily checked that there is no way to divide the 7 units among the five vertices while ensuring that the endpoints of each edge receive a total value equal to at least the weight of the edge. Indeed, any allocation x that satisfies $x(e) \geq w(e)$ for every edge e of our graph would need to have $x_A + x_B + x_C \geq 8$. Hence the core of this matching game is empty.

Even if the core is non-empty, it might be a very large set, meaning that the question of which outcome is the most fair remains largely unanswered. To address these issues Schmeidler [73] introduced the concept of the nucleolus as a single point solution concept for cooperative games. The idea behind the nucleolus is that it recursively maximizes the pay-off of the least satisfied coalition. For a more formal definition, we need the notion of excess. Given a coalition S, and an allocation $x : 2^N \to \mathbb{R}$, the excess of S with respect to x is defined as $x(S) - v(S)$. This difference captures the incentive of a coalition to deviate: if the excess is positive, then the coalition is fairly satisfied and would have no incentive to deviate; while if the excess is negative then the coalition would have an incentive to deviate since its agents can derive more value by leaving the grand coalition and cooperating among themselves. The nucleolus can then be defined as the allocation that lexicographically maximizes the vector of non-decreasingly ordered excesses over the set of all allocations.

Figure 1.8 displays the nucleolus for the same matching game that we considered in Figure 1.7. Since the core is empty as we observed, there have to be some coalitions that receive negative excess. These are exactly the coalitions corresponding to the triangle edges, and each one of them has an excess of $-2/3$.

While its definition might seem hard to digest, the nucleolus turns out to have several very nice properties. For instance, it always exists and it is part of the core whenever the core is non-empty. But perhaps most importantly, the nucleolus is always unique, meaning that it specifies a unique way to divide the value of the grand coalition among all the agents. Due to these properties, applications of the nucleolus have arisen in many fields such as cost allocations problems, surplus sharing problems, and in insurance and bankruptcy policies [58].

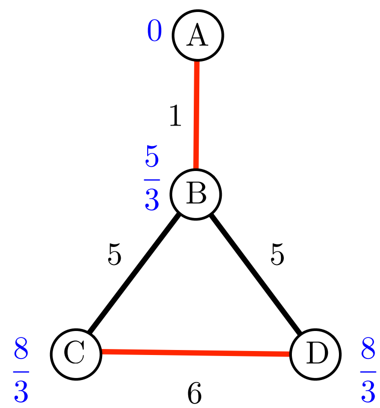

Figure 1.8: The nucleolus of a matching game with empty core.

The nucleolus has also been used to explain a rather curious division problem from the Babylonian Talmud, whose solution has puzzled academics for thousands of years [2]. The problem concerns a man that dies with outstanding debts to three creditors. Creditor 1 is owed 100, creditor 2 is owed 200, and creditor 3 is owed 300. It is decided that in order to make up for his debts, the man's estate will be divided among the three creditors. The question is, what proportion of the estate should each creditor get? The Talmud offers answers for three different scenarios: when the estate size is 100, 200, and 300. The answers are portrayed in Table 1.1 below.

Table 1.1: Mysterious division of a man's estate

Estate size	Creditor 1	Creditor 2	Creditor 3
100	100/3	100/3	100/3
200	50	75	75
300	50	100	150

When the size of the estate is 100, the suggested division gives equal amounts to each creditor, while when the size of the estate is 300 the amounts are proportional to the amount of money owed. Perhaps even more puzzling is the suggested division for the 200 case, where the second and third creditor receive equal amounts, while the first receives a smaller amount. Not being able to explain the pattern behind this decisions, many scholars have suggested that 200 case an error introduced during transcription. However, Aumann and Maschler [2] showed that the divisions given in the Talmud correspond to the nucleolus of a properly defined cooperative game. Intuitively, any subset of creditors $S \subseteq \{1, 2, 3\}$ can guarantee themselves an amount equal to the maximum between zero and the remainder of the estate after every creditor outside S has been paid in full. This characteristic function defines a cooperative game whose nucleolus coincides with the value given in the table. For example for the case where the estate size is 200, each coalition $S \subseteq \{1, 2, 3\} \setminus \{2, 3\}$ has value $v(S) = 0$ while $v(S) = 100$ for $S = \{2, 3\}$. The excess vector corresponding corresponding to the suggested division of $(50, 75, 75)$ is $(50, 50, 75, 75, 125, 125)$ and this is indeed the unique lexicographically maximum over the set of all allocations.

Due to its many applications and useful properties, the question of whether the nucleolus can be computed efficiently is a major problem in cooperative games. This is particularly true for matching games. For the case of matching games with non-empty core, and matching games with unit weights, polynomial time algorithms for computing the nucleolus have been known to exist for a while. However, for matching games with empty core and general weights, the question remains to be answered and it is considered one of the major open problems in the area.

1.3.1 Contributions

We study the open question of whether the nucleolus can be computed efficiently for matching games with an empty core and general weights. We follow previous techniques that compute the nucleolus using a sequence of linear programs that iteratively refine the underlying feasible region until it consists of a single allocation that is the nucleolus of the game. The main challenge in turning this procedure into a polynomial time algorithm is describing each linear program using only a polynomial number of constraints. The first linear program in this sequence defines the least core, which is the set of allocations that maximize the minimum excess of a coalition. Thus, obtaining a polynomial sized characterization of the least core is the initial step in establishing an efficient algorithm. In fact, this also turns out to be the crucial step since similar characterizations can be obtained for all subsequent linear programs by induction.

As part of our results we require a generalisation of the least core obtained by relaxing the requirement that the total value of an allocation equals the maximum weight of a matching. Instead we allow a total value of Δ, where Δ is a fixed constant that is at least the maximum weight of a matching and strictly less than the maximum weight of a fractional matching. Our first result shows that existing characterizations of the least core be extended to this generalised least core for matching games that are node-weighted, meaning that the weight of each edge can be obtained as the sum of non-negative weights of the endpoints.

Result 8. *There exists a polynomial sized characterization of the generalised least core of any matching game that is node-weighted.*

We then use this result as follows: given a weighted graph (G, w) that has an empty core and is not node-weighted we first obtain an optimal dual solution y to the maximum weight fractional matching problem for (G, w). We then look at the subgraph $G[y]$ of G that is induced by the edges of G that are tight with respect to y, and consider the generalised least core of $G[y]$ obtained by setting Δ to be equal to the maximum weight of a matching in (G, w). If this is equal to the weight of a maximum weight matching in $(G[y], w)$, then this is just the regular least core of $G[y]$. However, since this might not always be the case, we really require the generalisation of the least core. Given this generalised least core, we check whether it contains any extendable allocations, where extendable allocations are defined as having the same minimum excess in both G and $G[y]$. The following is then our main result.

Result 9. *If the generalised least core of $G[y]$ contains at least one extendable allocation, then the least core of G is equal to the set of all extendable allocations. Moreover, the nucleolus of the matching game (G, w) can be computed in polynomial time.*

We conclude the chapter by showing that the existence of an extendable allocation is independent of our original choice of optimal dual solution.

Result 10. *Let y_1 and y_2 be two optimal solutions to the dual of the maximum weight fractional matching linear program of (G, w). Then the generalised least core of $G[y_1]$ contains an extendable allocation if and only if the generalised least core of $G[y_2]$ contains an extendable allocation.*

1.3.2 Related work

This computational complexity of the nucleolus for cooperative games has been studied by Maschler et al. [63] who devised an approach, often referred to as Maschler's scheme, that is based on solving a sequence of linear programs that refine the core until it becomes a singleton set containing exactly the nucleolus. While the number of linear programs in this scheme is no more then the number of agents in the game, each linear program contains an exponential number of constraints, one for each coalition. In general it is unclear whether these linear programs can be solved efficiently [18].

On the positive side, there are several polynomial time algorithms for computing the nucleolus for certain classes of cooperative games. Solymosi and Raghavan [76] provide a combinatorial algorithm for computing the nucleolus of matching games on bipartite graphs, also known as assignment games. An important property of this class of games is that the core is always non-empty. Biro et al. extend this result to non-bipartite graphs but maintaining the assumption that the core is non-empty. Chen et al. [13] provide an algorithm for fractional matching games, whose core is also non-empty.

Kern and Paulusma [53] were the first to study matching games with an empty core in the case where all edges have unit weight. They provide a polynomial time algorithm for computing the nucleolus that relies on initially obtaining a polynomial sized description of the least core, which is the first linear program that has to be solved in Maschler's scheme. In Paulusma's thesis [68] this result is extended to node-weighted games, matching games where there exists an assignment of non-negative weights to the nodes of the graph such that the weight of each edges is equal to the sum of the node weights of its endpoints.

Faigle et al. [29] propose a different solution concept called the nucleon, which differs from the nucleolus by defining the excess of a coalition as $x(S)/v(S)$ as supposed to $x(S) - v(S)$. They provide an efficient algorithm for computing the nucleon of any matching game.

Outside of matching games, efficient algorithms for computing the nucleolus have also been found for standard tree games [66, 40], convex games [57], flow games [19] , cyclic-permutation games [77], airport profit games [9] and spanning connectivity games [4] among others.

On the negative side, several hardness results also exist. For the case of minimum cost spanning tree games Faigle et al. showed that testing core membership [28] as well as computing the nucleolus [30] are NP-hard. A similar result is shown by Elkind et al. [25] for weighted threshold games. In a follow up paper Elkind et al. [26] provide a pseudo-polynomial time algorithm that computes the nucleolus for k-vector weighted voting games.

For an excellent treatise of general computational issues arising in cooperative game theory we refer the reader to [12].

1.4 Outline of the thesis

In chapter 2 we introduce the main terminology and background results requires for this thesis. Section 2.1 introduces the main matching theory concepts. This includes the maximum weight matching and maximum weight fractional matching problems and corresponding linear programming formulations in 2.1.1, the Edmonds-Gallai decomposition of a graph and its properties in 2.1.2, and c-matchings in 2.1.3. The major concepts of cooperative games are all defined in the section 2.2: the core 2.2.1, the least core 2.2.2, the nucleolus 2.2.3, and the prekernel 2.2.4. Section 2.3 introduces the special type of cooperative games studied in this thesis: matching games, and their generalisation c-matching games 2.3.6. We review some famous results specific to matching games regarding the 2.3.1, the least core 2.3.2 and a linear programming scheme for computing the nucleolus 2.3.3. We also introduce existing results for computing the nucleolus when the core is non-empty 2.3.4, and when the graph is node-weighted 2.3.5. In section 2.4 we introduce network bargaining games and review existing results regarding unit capacities 2.4.1 and connections to cooperative games 2.4.2. In section 2.5 we introduce stable matchings including the classical stable marriage model in 2.5.1 together with the Gale-Shapley algorithm, the generalisation known as stable marriage with ties and incomplete lists in 2.5.2 and extensions to three dimensions in 2.5.3.

In chapter 3 we present our result on network bargaining games with general capacities. Section 3.1 contains the results 3 and 4, while section 3.2 contains results 1 and 2. In chapter 4 we have the results for stable matchings, in particular result 5 in section 4.1, result 6 in section 4.2 and result 7 in section 4.3. In chapter 5 we present out results on matching games with empty core and general weights. Result 8 is presented in section 5.2, result 9 in sections 5.3 and 5.5, and result 10 in section 5.4. Finally in chapter 6 we conclude with some directions for future work.

Chapter 2

Background

2.1 Matching theory

In this chapter we review some fundamental results from matching theory that will be used throughout the thesis. For a more in depth take on these topics we refer the reader to [59].

2.1.1 Maximum weight matchings

We let G denote an undirected simple graph with vertex set $V(G)$ and edge set $E(G)$. The edges of G have weights given by a vector $w \in \mathbb{R}^{|E(G)|}$. For the purposes of this thesis we can assume that the weight vector is non-negative, meaning that $w_e \geq 0$ for all $e \in E(G)$. For a given vertex $v \in V(G)$ we let $\delta(v)$ denote the set of edges in $E(G)$ that are incident with v. Given a subset of vertices $S \subseteq V(G)$ we denote by $\gamma(S)$ the set of edges in $E(G)$ with both endpoints in S and let $G[S]$ denote the subgraph of G with vertex set S and edge set $\gamma(S)$. Given a vector x defined over a finite set N and $S \subseteq N$ we use the standard notation $x(S)$ to denote $\sum_{i \in S} x_i$. In particular if $x \in \mathbb{R}^{|V(G)|}$ and e is an edge with endpoints u and v then $x(e)$ denotes $x_u + x_v$.

We start by introducing the main underlying notion of this thesis: a matching.

Definition 2.1.1. *A* matching *of G is a subset of edges $M \subseteq E(G)$ no two of which share an endpoint. The weight of M is $w(M) = \sum_{e \in M} w_e$. The maximum weight of a matching of G is denoted by $\nu(G)$ and defined as*

$$\nu(G) := \max \{ w(M) : M \text{ is a matching } \} .$$

A matching M is said to *cover* a vertex $v \in V$ if $|M \cap \delta(v)| = 1$, and otherwise M is said to *expose* v. A matching M is *perfect* if it does not expose any vertex of the graph, and *near-perfect* if it exposes exactly one vertex of the graph.

A graph G is *factor critical* if G has no perfect matching but for each vertex v of G there exists a near-perfect matching of G exposing v.

Given two disjoint sets of vertices A and B we say that a matching M matches A *completely* into B if each vertex of A is matched by M to a vertex of B.

We say that a matching M is a *maximum weight matching* if $w(M) = \nu(G)$ and a *maximum cardinality matching* if $|M| \geq |N|$ for any other matching N. We define the linear program

$$\begin{aligned} (P_{FM}) \quad \max \quad & \sum_{e \in E(G)} w_e x_e & & (2.1) \\ x(\delta(u)) &\leq 1 & \text{for all } u \in V(G) \\ x_e &\geq 0 & \text{for all } e \in E(G). \end{aligned}$$

The dual linear program to P_{FM} is

$$\begin{aligned} (D_{FM}) \quad \min \quad & \sum_{u \in V(G)} y_u & & (2.2) \\ y_u + y_v &\geq w_e & \text{for all } e = (u,v) \in E(G) \\ y_u &\geq 0 & \text{for all } u \in V(G). \end{aligned}$$

This brings us to the definition of a fractional matching.

Definition 2.1.2. *A* fractional matching *is a vector x that is a feasible solution to P_{FM}. The weight of a fractional matching x is $\sum_{e \in E(G)} x_e w_e$, the objective value achieved by the solution x. The optimal value of P_{FM} is denoted by $\nu_f(G)$.*

We say that a fractional matching x is a *maximum weight fractional matching* if $\sum_{e \in E} x_e w_e = \nu_f(G)$. Since the incidence vector of any matching is a feasible solution to P_{FM} we immediately have $\nu(G) \leq \nu_f(G)$. The following is a famous result of Egerváry [24].

Theorem 2.1.3. *If G is bipartite then $\nu_f(G) = \nu(G)$.*

For non-bipartite graphs this is no longer true since the optimal solution to P_{FM} might be fractional. However the following result states that any optimal solution to P_{FM} is half-integral [5]

Theorem 2.1.4. *Let x^* be a basic solution to P_{FM}. Then $x^*_e \in \left\{0, \frac{1}{2}, 1\right\}$ for all $e \in E(G)$ and the edges $\left\{e \in E(G) : x^*_e = \frac{1}{2}\right\}$ induce vertex disjoint odd cycles.*

We say that a basic fractional matching x^* is *perfect* if $\sum_{e \in \delta(v)} x^*_e = 1$ for all vertices $v \in V(G)$. We will require the following result from [17], also found in [70], about fractional matching of factor critical graphs.

Theorem 2.1.5. *Let G be a factor critical graph with $|V(G)| > 1$. For each vertex $v \in V(G)$ there exists a basic perfect fractional matching x^* of G such that $\left\{e \in E(G) : x^*_e = \frac{1}{2}\right\}$ contains a single cycle C that passes through v.*

The complementary slackness conditions imply that any maximum weight fractional matching x^* and optimal dual solution y^* to D_{FM} must satisfy

$$y^*_u + y^*_v = w_e \quad \text{for all } e = (u, v) \in E(G) \text{ such that } x^*_e > 0 \tag{2.3}$$

$$x^*(\delta(v)) = 1 \quad \text{for all } v \in V(G) \text{ such that } y^*_v > 0 \tag{2.4}$$

Let $\mathcal{V}_{\text{odd}}$ be the set of all odd sized subsets of V of size at least 3. We define the linear program

$$\begin{aligned} (P_M) \quad \max \quad & \sum_{e \in E(G)} w_e x_e && \qquad (2.5)\\ x(\delta(u)) &\leq 1 && \text{for all } u \in V(G) \\ x(\gamma(B)) &\leq (|B| - 1)/2 && \text{for all } B \in \mathcal{V}_{\text{odd}} \\ x_e &\geq 0 && \text{for all } e \in E(G). \end{aligned}$$

The dual linear program to P_M is given by

$$(D_M) \quad \min \quad \sum_{u \in V} y_u + \sum_{B \in \mathcal{V}_{\text{odd}}} \frac{|B|-1}{2} z_B \tag{2.6}$$

$$y_u + y_v + \sum_{B \in \mathcal{V}_{\text{odd}}:\, e \in \gamma(B)} z_B \geq w_e \qquad \text{for all } e = (u,v) \in E(G) \tag{2.7}$$

$$y_u \geq 0 \qquad \text{for all } u \in V(G)$$

$$z_B \geq 0 \qquad \text{for all } B \in \mathcal{V}_{\text{odd}}.$$

Note that any incidence vector of a matching is a feasible solution to P_M. Hence the optimal value of P_M is clearly an upper bound on $\nu(G)$. The following classic result of Edmonds [22] states that this bound is exact.

Theorem 2.1.6. *The optimal value of P_M is equal to $\nu(G)$.*

Despite the exponential number of constraints of the linear program P_M, a celebrated result of Edmonds [22] provides a polynomial time combinatorial algorithm that constructs a primal-dual optimal pair to P_M and D_M. This algorithm is known as Edmonds blossom algorithm, and for a detailed description of it we refer the reader to [16]. For the purposes of this thesis it suffices to use the fact that a maximum weight matching of G can be computed in polynomial time.

2.1.2 Edmonds-Gallai decomposition

For our results we will need the Edmonds-Gallai decomposition of a graph. In order to describe this decomposition we first need to introduce some more notation.

Definition 2.1.7. *Let $A \subseteq V(G)$. We let $\mathcal{C} = \mathcal{C}(A)$ denote the set of even sized components of $G \backslash A$ and $\mathcal{D} = \mathcal{D}(A)$ the set of odd sized components of $G \backslash A$. The subset A is called a* Tutte set *if every maximum cardinality matching M in G can be decomposed as*

$$M = M_{\mathcal{C}} \cup M_{A,\mathcal{D}} \cup M_{\mathcal{D}}$$

where $M_{\mathcal{C}}$ is a perfect matching in the union of all even components $\mathcal{C}$; $M_{\mathcal{D}}$ induces a near-perfect matching in all odd components $D \in \mathcal{D}$; and $M_{A,\mathcal{D}}$ is a matching that matches A completely into $\cup\mathcal{D}$ the union of odd components.

The following theorem proven by both Gallai [38, 39] and Edmonds [23] guarantees that we can compute such a set efficiently and lists some of its useful properties.

Theorem 2.1.8. *Given a graph G, one can construct a Tutte set $A \subseteq V(G)$ in polynomial time such that*

1. *Each odd component $D \in \mathcal{D}$ is factor critical.*
2. *For each $D \in \mathcal{D}$ there exists a maximum cardinality matching in G that exposes a vertex of D.*

The triple $(A(G), \mathcal{D}(G), \mathcal{C}(G))$ is called the Edmonds-Gallai decomposition of G. The components in $\mathcal{D}(G)$ are referred to as *odd* components, while the ones in $\mathcal{C}(G)$ are referred to as *even* components.

2.1.3 Vertex capacities

If in addition to edge weights we also have vertex capacities we use the following generalised definition of a matching.

Definition 2.1.9. *Let G be a graph with vertex capacities $c_v \in \mathbb{Z}_+$ for all $v \in V(G)$. A* c-matching *of G is a subset of edges $M \subseteq E(G)$ satisfying*

$$|M \cap \delta(v)| \leq c_v \quad \text{for all } v \in V(G).$$

We let $\nu^c(G)$ denote the maximum weight of a c-matching in G.

We define the linear program

$$\begin{aligned} (P^c_{FM}) \quad \max \quad & \sum_{e \in E} w_e x_e && \quad (2.8) \\ & x(\delta(u)) \leq c_u && \text{for all } u \in V(G) \\ & 0 \leq x_e \leq 1 && \text{for all } e \in E(G). \end{aligned}$$

We let $\nu_f^c(G)$ denote the optimal value of P_{FM}^c.

2.2 Cooperative games

In this section we introduce the main concepts from cooperative game theory that are used in this thesis.

Definition 2.2.1. *A* cooperative game *is a pair* (N, v) *where* $N = \{1, \cdots, n\}$ *is a finite set and* $v : 2^N \to \mathbb{R}_+$ *is a function satisfying* $v(\emptyset) = 0$. *The set* N *is said to represent the agents of the game, and the function* v *is called the* characteristic function *of the game.*

A subset of agents $S \subseteq N$ is called a *coalition* and the set N is called the *grand coalition*. The function v assigns to each coalition an outcome $v(S)$ called the *value* of coalition S. This value represents the total benefit that the players in S can derive by choosing to cooperate and form the coalition S. However, $v(S)$ does not specify anything about how this value should be split among the agents in S.

Our definition of cooperative games implicitly assumes that the value $v(S)$ can be divided amongst the agents in S in any way that they choose. Such games are also referred to as *transferable utility* games or *TU*-games for short.

All cooperative games considered in this thesis have the property that it is always profitable for two different coalitions to join together. This is formally known as super additivity.

Definition 2.2.2. *A cooperative game* (N, v) *is* super additive *if it satisfies* $v(S \cup T) \geq v(S) + v(T)$ *for all pairs of disjoint coalitions* $S, T \subseteq N$.

In super additive games there is no benefit for the agents to form disjoint coalitions, since they can always derive at least as much profit by joining in the grand coalition. It is thus assumed for such games that the grand coalition will always form, and the remaining question is to decide how the agents divide the total value $v(N)$ amongst themselves. A vector that specifies such a division is called an *allocation*.

Definition 2.2.3. *An* allocation *of the cooperative game* (N, ν) *is a vector* $x \in \mathbb{R}_+^N$ *that satisfies* $x(N) = v(S)$*. The set of all allocations of* (N, v) *is denoted by* $I(N, v)$*.*

If in addition to being an allocation, a vector x satisfies $x(\{i\}) \geq v(\{i\})$ for all $i \in N$ then x is called an *imputation.* Intuitively this correspond to individual rationality since it ensures that no one is worse off by cooperating. The type of cooperative games considered in this thesis all have the property that $v(\{i\}) = 0$ for any $i \in N$, therefore we do not need to make a distinction between allocations and imputations.

2.2.1 The core

The *excess* of a coalition $S \subseteq N$ with respect to an allocation x is defined as

$$e(S, x) := x(S) - v(S)$$

The core represents the set of all allocations under which no coalition has negative excess.

Definition 2.2.4. *The* core *of a cooperative game* (N, v) *is defined as*

$$\textit{core}(N, v) = \{x \in I(N, v) : x(S) \geq v(S) \ \textit{ for all } S \subseteq N\}$$

2.2.2 The least core

While the core is a very important solution concept, it can often be empty. After all, the requirement of satisfying every single coalition is rather strong. As an alternative, one often works with a relaxation of the core that is guaranteed to be non-empty. To define this relaxation, we first introduce the concept of minimum excess. Given an allocation x we define the *minimum excess* of a coalition with respect to x as

$$e_{\min}(x) := \min \{e(S, x) : \emptyset \neq S \neq N\}$$

We then maximize this value over the set of allocations to obtain

$$\epsilon_1 := \max_{x \in I(N,v)} e_{\min}(x).$$

Note that $\epsilon_1 \geq 0$ if and only if the core is non-empty.

Definition 2.2.5. *The* least core *of a cooperative game* (N, v) *is defined as*

$$\textit{leastcore}(N, v) = \{x \in I(N, v) : x(S) \geq v(S) + \epsilon_1 \ \textit{for all}\ S \subseteq N\}$$

Unlike the core, the least core is always non-empty. Furthermore, the least core is a subset of the core whenever the latter is non-empty.

2.2.3 The nucleolus

One can refine the concept of the least core by extending the idea of maximizing the minimum excess. Given any allocation x, we define the *excess vector* $\theta(x)$, as the 2^{N-2} dimensional vector whose components are the excesses $e(S, x)$, arranged in non-decreasing order, of all the coalitions $\emptyset \neq S \subset N$. A vector $x \in \mathbb{R}^m$ is said to be *lexicographically smaller than or equal to* $y \in \mathbb{R}^m$, denoted by $x \preceq y$, if $x = y$ or there exists an index j, $1 \leq j < n$, such that $x_i = y_i$ for all $i \leq j$ and $x_{j+1} < y_{j+1}$. We now arrive at the definition of the nucleolus.

Definition 2.2.6. *The* nucleolus *of a cooperative game* (N, v) *is defined as*

$$\eta(N, v) = \{x \in I(N, v) : \theta(y) \preceq \theta(x) \ \textit{for all}\ y \in I(N, v)\}$$

That is, the nucleolus is the set of allocations that lexicographically maximize $\theta(x)$ over the set of allocations $I(N, v)$. If follows immediately from definition that the nucleolus always exists and is a subset of the least core. It is not immediately clear however that the nucleolus defines a unique point. The result was first shown in [73].

Theorem 2.2.7. *The set* $\eta(N, v)$ *consists of exactly one point.*

2.2.4 The prekernel

Another important solution concept for cooperative games is the prekernel. In order to define it this we first need to introduce the notion of power. Let (N, v) be a cooperative game, and $i, j \in N$ with $i \neq j$. The power of i over j with respect to the allocation x is defined as

$$s_{ij}(x) := \max_{S \subseteq N : i \in S, j \notin S} v(S) - x(S). \tag{2.9}$$

The prekernel is a solution concept based on the idea that these powers should be equalized.

Definition 2.2.8. *The* prekernel *of a cooperative game* (N, v) *is defined as*

$$\textit{prekernel}(N, v) := \{x \in I(N, v) : s_{ij}(x) = s_{ji}(x) \text{ for all } i, j \in N, i \neq j\}$$

While its definition is not very intuitive, the prekernel does have some nice connections to other solution concepts. The prekernel is always non-empty and satisfies the following property [31]

Theorem 2.2.9. *Let* (N, v) *be a cooperative game. If core*$(N, v) \cap$ *prekernel*(N, v) *consists of a single point* $x^* \in \mathbb{R}^{|N|}$, *then* x^* *is the nucleolus of the game.*

2.3 Matching games

Matching games are a special class of cooperative games derived from an underlying maximum matching problem.

Definition 2.3.1. *Given an undirected simple graph* $G = (V, E)$ *with edge weights* $w : E \to \mathbb{R}_+$ *we define the corresponding* matching game *on* G *as the cooperative game* (N, v) *where* $N = V$ *and*

$$v(S) := \nu(G[S]) \quad \text{for all } S \subseteq N$$

where $\nu(G[S]) = \max\{w(M) : M \text{ is a matching in } G[S]\}$

Matching games are super additive. A matching game on a bipartite graph is also called an *assignment game*.

2.3.1 The core

It turns out that for matching games, the core can be characterized using only the constraints corresponding to coalitions of size one and two [68].

Theorem 2.3.2. *Let $G = (V, E)$ be a graph with edge weights $w : E \to \mathbb{R}_+$ and (N, v) the corresponding matching game on G. Then*

$$core(N, v) = \{x \in \mathbb{R}_+ : x(N) = v(N) \text{ and } x(e) \geq w_e \text{ for all } e \in E\}.$$

Recall our definition of the linear program P_{FM} for the maximum weight matching problem

$$(P_{FM}) \quad \max \sum_{e \in E} w_e x_e$$
$$x(\delta(u)) \leq 1 \quad (u \in V)$$
$$x_e \geq 0 \quad (e \in E)$$

and its dual

$$(D_{FM}) \quad \min \sum_{u \in V} y_u$$
$$y_u + y_v \geq w_e \quad (e = uv \in E)$$
$$y_u \geq 0 \quad (u \in V).$$

Using our new characterization of the core we can see that y is a core allocation if and only if y is a feasible solution to D_{FM} with objective value $\nu(G)$. We therefore have the following useful observation, also found in [20].

Theorem 2.3.3. *Let $G = (V, E)$ be a graph with edge weights $w : E \to \mathbb{R}_+$ and (N, v) the corresponding matching game on G. Then $core(N, v) \neq \emptyset$ if and only if $\nu_f(G) = \nu(G)$.*

Furthermore, from complementary slackness we have that for any $x \in \text{core}(N, v)$

- $x_i = 0$ for all $i \in V$ such that i is exposed by a maximum weight matching
- $x(e) = w_e$ for all $e \in E$ such that e belongs to a maximum weight matching

2.3.2 The least core

In addition to the core having a nice representation for matching games, the description of the least core can also be simplified using the following observation from [68].

Theorem 2.3.4. *Let $G = (V, E)$ be a graph with edge weights $w : E \to \mathbb{R}_+$ and (N, v) the corresponding matching game on G. Then core$(N, v) \subseteq \mathbb{R}^N_+$.*

This allows us to work with the following equivalent characterization of the core. Let $\mathcal{M}$ be the set of all matchings in G. Define the linear program

$$\begin{aligned} (P_1) \quad & \text{maximize } \epsilon \\ \text{subject to } & x(M) \geq w(M) + \epsilon \quad M \in \mathcal{M} \\ & x(V) = \nu(G) \\ & x \geq 0. \end{aligned} \tag{2.10}$$

For any ϵ let $P_1(\epsilon)$ be the set of vectors x so that (x, ϵ) is a feasible solution to P_1, and let ϵ_1 is the optimum value of P_1. Theorem 2.3.4 implies that leastcore$(N, v) = P_1(\epsilon_1)$.

2.3.3 The nucleolus: Maschler's scheme

In this subsection we review the standard procedure used for computing the nucleolus found in [63]. This procedure applies to any cooperative game (N, v) but we assume here that (N, v) is a matching game. We start by solving the linear program

$$\begin{aligned} (Q_1) \quad & \text{maximize } \epsilon \\ \text{s.t.} \quad & x(S) \geq v(S) + \epsilon \quad \text{for all } S \notin \mathcal{S}_0 \\ & x(N) = v(N). \end{aligned}$$

where $\mathcal{S}_0 := \{\emptyset, N\}$.

Let ϵ_1 be the optimum value for Q_1 and for a given ϵ define $Q_1(\epsilon)$ to be the set of all x such that (x, ϵ) is feasible for Q_1. Note that under these definitions, we have core$(N, v) = Q_1(0)$ and leastcore$(N, v) = Q_1(\epsilon_1)$.

Now if $Q_1(\epsilon_1)$ does not consist of a single point we proceed by defining $\mathcal{S}_1$ to be the set of coalitions that are fixed by $Q_1(\epsilon_1)$

$$\mathcal{S}_1 := \{S \subseteq N : x(S) = v(S) + \epsilon_1 \text{ for all } x \in Q_1(\epsilon_1)\}$$

And we solve the second linear program

$$\begin{aligned} (Q_2) \quad & \text{maximize } \epsilon \\ \text{s.t.} \quad & x(S) = v(S) + \epsilon_1 \quad \text{for all } S \in \mathcal{S}_1 \\ & x(S) \geq v(S) + \epsilon \quad \text{for all } S \notin \mathcal{S}_0 \cup \mathcal{S}_1 \\ & x(N) = v(N) \end{aligned}$$

defining in a similar way ϵ_2 be the optimum value for Q_2 , $Q_2(\epsilon_2)$ to be the set of all x such that (x, ϵ_2) is feasible to Q_2 and

$$\mathcal{S}_2 := \{S \subseteq N : x(S) = v(S) + \epsilon_2 \text{ for all } x \in Q_2(\epsilon_2)\}$$

We continue in this way until we reach

$$\begin{aligned} (Q_r) \quad & \text{maximize } \epsilon \\ \text{s.t.} \quad & x(S) = v(S) + \epsilon_1 \quad \text{for all } S \in \mathcal{S}_1 \\ & x(S) = v(S) + \epsilon_2 \quad \text{for all } S \in \mathcal{S}_2 \\ & \vdots \\ & x(S) = v(S) + \epsilon_{r-1} \quad \text{for all } S \in \mathcal{S}_{r-1} \\ & x(S) \geq v(S) + \epsilon \quad \text{for all } S \notin \cup_{i=0}^{r-1} \mathcal{S}_i \\ & x(N) = v(N) \end{aligned}$$

such that $Q_r(\epsilon_r)$ consists of a single point $\eta(N, v)$, the nucleolus of the game.

It follows from the definition of this procedure that $\epsilon_1 < \epsilon_2 < \cdots < \epsilon_r$ and $\eta(N, v) = Q_r(\epsilon_r) \subset Q_{r-1}(\epsilon_{r-1}) \subset \cdots \subset Q_2(\epsilon_2) \subset Q_1(\epsilon_1)$. Since in each iteration at least one equality constraints is added that is independent from the previous equality constraints, it follows that the dimension of feasible region of the above sequence of LPs decreases by at least 1 with each iteration. Hence $r \leq |N|$ and the number of LP's is polynomial in the size of the input.

Despite the number of iterations being polynomial, there remain two main difficulties for turning this procedure into an efficient algorithm. The first is computing the sets of coalitions that are fixed at every step, and the second is dealing with the exponential number of constraints. One way of circumventing the latter is via a polynomial-time separation oracle, an algorithm that, given a candidate feasible solution, either confirms that it is feasible or outputs a violated constraint. Given such an oracle, one could solve each of the LP's efficiently using the ellipsoid method [41, 74]. Our approach in this thesis will be different. We focus instead on obtaining an equivalent characterization for each LP that contains only a polynomial number of constraints.

2.3.4 The nucleolus when the core is non-empty

In the case of matching games with a non-empty core, Maschler's scheme can be turned into a polynomial time algorithm by including the constraints corresponding only to coalitions of size one and two.

Let $G = (V, E)$ be a graph with edge weights $w : E \to \mathbb{R}_+$ and (N, v) the corresponding matching game on G. Suppose that $\text{core}(N, v) \neq \emptyset$ or equivalently $\nu(G) = \nu_f(G)$. The key observation is that if the core is non-empty then the optimal value ϵ_k of each LP Q_k in Maschler's scheme is non-negative. Hence the constraints

$$\begin{aligned} x(e) &\geq w_e + \epsilon_k && \text{for all } \ e \in E \\ x_i &\geq \epsilon_k && \text{for all } \ e \in N \end{aligned}$$

imply all the constraints $x(S) \geq v(S) + \epsilon_k$ for all $S \subseteq N$. Therefore we can replace the LP

Q_k with the equivalent program

$$
\begin{array}{lll}
(Q_k^+) & \text{maximize } \epsilon & \\
\text{s.t.} & x(e) = w_e + \epsilon_1 & \text{for all } \; e \in E_1 \\
& x_i = \epsilon_1 & \text{for all } \; e \in N_1 \\
& x(e) = w_e + \epsilon_2 & \text{for all } \; e \in E_2 \\
& x_i = \epsilon_2 & \text{for all } \; i \in N_2 \\
& \vdots & \\
& x(e) = w_e + \epsilon_{k-1} & \text{for all } \; e \in E_{k-1} \\
& x_i = \epsilon_{k-1} & \text{for all } \; i \in N_{k-1} \\
& x(e) \geq w_e + \epsilon & \text{for all } \; e \notin \cup_{j=0}^{k-1} E_j \\
& x_i \geq \epsilon & \text{for all } \; i \notin \cup_{j=0}^{k-1} N_j \\
& x(N) = v(N) &
\end{array}
$$

where E_j is the set of all edges for which $x(e) = w_e + \epsilon_j$ for all $x \in Q_j^+$ and N_j is the set of all vertices $i \in N$ for which $x_i = \epsilon_j$ for all $x \in Q_j^+$ and N_j. Since each Q_K^+ has a polynomial number of constraints, each linear program in Maschler's scheme can now be solved efficiently, and thus we have the following result also stated in [68].

Theorem 2.3.5. *Let $G = (V, E)$ be a graph with edge weights $w : E \to \mathbb{R}_+$ and (N, v) the corresponding matching game on G. If core$(N, v) \neq \emptyset$ then the nucleolus of (N, v) can be computed in polynomial time.*

2.3.5 The nucleolus for node weighted matching games

The approach in the previous section that relied on replacing Q_k with the equivalent program Q_k^+ fails whenever $\epsilon_k < 0$. Therefore a new technique is needed to reduce the constraints of each LP to a polynomial number.

In this section we review the result of Paulusma [68] that achieves this for the case of matching games that have an empty core, but whose weights satisfy the following property.

Definition 2.3.6. *Let $G = (V, E)$ be a graph with edge weights $w : E \to \mathbb{R}_+$. We say that G is* node-weighted *if there exists a vector $y \in \mathbb{R}_+^V$ such that for all $e = (u, v) \in E$ we have $w_e = y_u + y_v$.*

The main idea behind Paulusma's result is obtaining an equivalent characterization of the least core, that unlike the linear program P_1 2.10, contains only a polynomial number of constraints. Once this characterization is obtained, it is not too difficult to prove by induction that each subsequent linear program in Maschler's scheme can also be written with only a polynomial number of constraints.

We present here Paulusma's characterization of the least core. Let G be a node-weighted graph where the node weights are given by the vector $y \in \mathbb{R}^{|V|}$. Recall that the least core is the set of all allocations that maximize the minimum excess. Using the representation of the least core given by the linear program P_1, this essentially means that the least core is the set of all allocation x that maximize the minimum excess of a matching, where the excess of a matching M is given by

$$x(M) - w(M) = \sum_{e \in M} (x(e) - w_e) = \sum_{e \in M} (x(e) - y(e)).$$

Since we are assuming that the core is empty it follows that $\nu(G) < y(V)$, since otherwise y would be a core allocation. Therefore for any allocation x we must have $x(V) < y(V)$. Hence there will be at least one edge e with negative excess, meaning that $x(e) < y(e)$. A crucial observation of [68] is that the set of edges that receive negative excess are the same for any least core allocation. More specifically, it is shown in [68] that there exists a partition of the edge set E into two sets $E_{\geq 0}$ and $E_{\leq 0}$ such that for any allocation $x \in \text{leastcore}(N, v)$ we have

$$x(e) - w_e \leq 0 \quad \text{for all } e \in E_{\leq 0}$$
$$x(e) - w_e \geq 0 \quad \text{for all } e \in E_{\geq 0}$$

Note that this also implies that any matching with minimum excess will be contained in the set $E_{\leq 0}$. In fact, the set of matchings with minimum excess will be the same for any allocation $x \in \text{leastcore}(N, v)$.

This partition of the edge set into edges receiving non-positive, respectively non-negative, excess in any least core allocation is based on the Edmonds-Gallai decomposition of the graph. Let $(A, \mathcal{D}, \mathcal{C})$ be the Edmonds-Gallai decomposition of G, as defined in 2.1.8. It is not too difficult to see that the emptyness of the core implies that there must exist at least one non-trivial component $D \in \mathcal{D}$ with $|D| > 1$ [68]. It is these non-trivial odd sized components that account for the negative excess in the graph. That is, the partition of E is obtained by letting

$$E_{\leq 0} = \bigcup_{D \in \mathcal{D}} E(D))$$

$$E_{\geq 0} = E \backslash E_{\leq 0}$$

where $E(D)$ is the edge set of the component D for all $D \in \mathcal{D}$. However, this observation by itself is not sufficient to obtain a polynomial sized characterization of the least core. In fact, something stronger turns out to be true: for each component $D \in \mathcal{D}$ the excess across all edges in $E(D)$ is equal. This implies that for any least core allocation, every maximum cardinality matching contained in the union of the odd components must have minimum excess. Hence it suffices to include only one constraint of the type $x(M) \geq w(M) + \epsilon$ for a fixed maximum cardinality matching on the odd components.

We state Paulusma's main result [68] below.

Theorem 2.3.7. *Let G be a node-weighted graph with respect to the non-negative node weights $y \in \mathbb{R}_+^{|V(G)|}$. Let $\mathcal{D}(G)$ denote the set of odd components in the Edmonds-Gallai decomposition of G and fix M_1 to be an arbitrary maximum cardinality matching contained* *in the union of these odd components. Define the linear program $\tilde{P} = \tilde{P}(G, y)$ as*

$$
\begin{array}{llll}
(\tilde{P}) & \textit{maximize} & \epsilon & \\
& \textit{subject to} & x(e) \geq w(e) & \textit{for all } e \in E \setminus \left(\bigcup_{D \in \mathcal{D}} E(D) \right) \\
& & x_i \leq y_i & \textit{for all } i \in \bigcup_{D \in \mathcal{D}} V(D) \\
& & x_i - y_i = x_j - y_j & \textit{for all } i, j \in V(D),\ D \in \mathcal{D}(G) \\
& & x(M_1) \geq y(M_1) + \epsilon & \\
& & x(V(G)) = \nu(G, y) & \\
& & x \geq 0 &
\end{array}
$$

and let $\tilde{\epsilon}_1 = \tilde{\epsilon}_1(G, y)$ denote its optimum value. Then $\tilde{\epsilon}_1(G, y) = \epsilon_1(G, y)$ and

$$\textit{leastcore}(G, y) = \tilde{P}(\epsilon_1).$$

Paulusma showed that this result leads to an efficient computation of the nucleolus by proving by induction that equivalent polynomial sized characterization can be obtained for each subsequent linear program in Maschler's scheme.

2.3.6 Vertex capacities

In the case where we also have capacities on the vertices we can define a generalised c-matching game as follows.

Definition 2.3.8. *Given an undirected simple graph $G = (V, E)$ with edge weights $w : E \to \mathbb{R}_+$ and vertex capacities $c : V \to \mathbb{Z}_+$ we define the corresponding c-matching game as the cooperative game (N, v) where $N = V$ and*

$$v(S) = \max \{ w(M) : \ M \textit{ is a c-matching in } G[S] \}$$

for all $S \subseteq N$.

The following result from [52] is an analogue of theorem 2.3.3 for general capacities

Theorem 2.3.9. *Let $G = (V, E)$ be a graph with edge weights $w : E \to \mathbb{R}_+$ and vertex capacities $c : V \to \mathbb{Z}_+$, and let (N, v) be the corresponding c-matching game. Then core$(N, v) \neq \emptyset$ if and only if the weight of the maximum weight c-matching is equal to the optimal value of the linear program P^c_{FM} given in section 2.1.*

2.4 Network bargaining

In this section we review all the basic definition and notations regarding network bargaining games.

Definition 2.4.1. *An instance of a* network bargaining game with general capacities *is a triple (G, w, c) where G is a undirected graph, $w \in \mathbb{R}^{|E|}_+$ is a vector of edge weights, and $c \in \mathbb{Z}^{|V|}_+$ is a vector of vertex capacities. If we omit the capacity vector c then (G, w) defines an instance of a* network bargaining game with unit capacities.

Given a c-matching M, we let d_u denote the degree of vertex u in M. We say that vertex u is *saturated* in M if $d_u = c_u$. The following definition specifies exactly what we mean by a solution to such a game.

Definition 2.4.2. *A* solution *to a network bargaining game (G, w, c) is a pair (M, z) where M is a c-matching of G and z is a vector belonging to $\mathbb{R}^{2|E|}_+$ that assigns each edge $e = (u, v) \in E$ a pair of values z_{uv}, z_{vu} such that $z_{uv} + z_{vu} = w_{uv}$ if $(u, v) \in M$ and $z_{uv} = z_{vu} = 0$ otherwise.*

The *allocation* associated with the solution (M, z) is the vector $x \in \mathbb{R}^{|V|}$ where x_u represents the total payoff of vertex u in the solution (M, z)

$$x_u = \sum_{(u,v)\in\delta(u)} z_{uv}, \tag{2.11}$$

An important concept for network bargaining games is that of an outside option.

Definition 2.4.3. *The* outside option *of vertex u with respect to a solution (M, z) is defined as*

$$\alpha_u(M, z) := \max\left(0, \max_{(u,v)\in E(G)\setminus M}\left(w_{uv} - \boldsymbol{1}_{[d_v=c_v]}\min_{vw\in M} z_{vw}\right)\right),$$

where $\boldsymbol{1}_E$ is the indicator function for the event E, which takes value one whenever the event holds, and zero otherwise. If $\{v : uv \in E(G)\setminus M\} = \emptyset$ then we set $\alpha_u(M, z) = 0$. We write α_u instead of $\alpha_u(M, z)$ whenever the context is clear.

The outside option is meant to capture the maximum payoff that a vertex could earn by breaking one of its current contracts in M and making a new contract. The idea behind the definition is that in order for a vertex u to make a new contract with a neighbour v, v must not be worse off after forming the contract with u. Now if v is unsaturated in M it can just form a new contract with u and even by earning zero profit from this contract v cannot be worse off. Therefore in this case u is assumed to get the full value of the contract given by w_{uv}. On the other hand if v is saturated, then it must first give up one of its current contracts before it can form a contract with u. Hence in order for v to not be worse off, it must now earn from the contract with u at least as much as it was earning from the contract that was dropped. Hence the maximum profit that u can earn in this case is the value of the contract uv minus the minimum profit that v earns in any of its current contracts.

We can now introduce the two main types of solutions that we are interested in: stable and balanced.

Definition 2.4.4. *A solution (M, z) is* stable *if for all $uv \in M$ we have $z_{uv} \geq \alpha_u(M, z)$, and for all unsaturated vertices u we have $\alpha_u(M, z) = 0$. A solution (M, z) is* balanced *if it is stable and in addition for all $uv \in M$ we have $z_{uv} - \alpha_u(M, z) = z_{vu} - \alpha_v(M, z)$.*

The following theorem of [51] characterizes the existence of stable solutions for network bargaining games with general capacities.

Theorem 2.4.5. *Let (G, w, c) be an instance of a network bargaining game with general capacities. Then (G, w, c) admits a stable solution if and only if $\nu_f^c(G) = \nu^c(G)$.*

Recall that $\nu_f^c(G)$ was defined as the optimal value of the linear program

$$
\begin{aligned}
(P_{FM}^c) \quad \max \quad & \sum_{e \in E} w_e x_e \\
& x(\delta(u)) \leq c_u \quad \text{for all } u \in V(G) \\
& 0 \leq x_e \leq 1 \quad\ \ \text{for all } e \in E(G).
\end{aligned}
$$

2.4.1 Unit capacities

The definitions of a network bargaining game simplify in the case where all vertices have unit capacities. Specifically a *solution* to the unit capacity game (G, w) is a pair (M, x) where M is now a matching of G and $x \in \mathbb{R}_+^{|V(G)|}$ assigns a value to each vertex such that for all edges $uv \in M$ we have $x_u + x_v = w_{uv}$ and for all $u \in V(G)$ not covered by M we have $x_u = 0$. Since each vertex has at most one unique contract, the vector x from the solution (M, x) is also the allocation vector in this case.

The *outside option* of vertex u can now be expressed as

$$\alpha_u(M, x) := \max_{(u,v) \in E(G) \setminus M} (w_{uv} - x_v),$$

where as before we set $\alpha_u(M, x) = 0$ whenever $\{v : uv \in E(G) \setminus M\} = \emptyset$.

A solution (M, x) is *stable* if for all $u \in V(G)$ we have $x_u \geq \alpha_u(M, x)$ and *balanced* if it is stable and in addition $x_u - \alpha_u(M, x) = x_v - \alpha_v(M, x)$ for all $uv \in M$. The following theorem of [55] characterizes the existence of stable solutions for network bargaining games with unit capacities. This is a special case of Theorem 2.4.5 that was proven earlier.

Theorem 2.4.6. *Let (G, w) be an instance of a network bargaining game. Then (G, w) admits a stable solution if and only if $\nu_f(G) = \nu(G)$.*

More interestingly, it is shown in [55] that this condition is always sufficient to characterize the existence of balanced solutions, and such solutions can be computed efficiently whenever they exist.

Theorem 2.4.7. *Let (G, w) be an instance of a network bargaining game. Then (G, w) admits a balanced solution if and only if it admits a stable one. Furthermore, the set of all balanced solutions can be computed in polynomial time.*

2.4.2 Connection to cooperative games

In this section we explore the close connections between the network bargaining games and cooperative games. Given a graph $G = (V, E)$ with edge weights $w \in \mathbb{R}_+^{|E(G)|}$ and vertex capacities $c \in \mathbb{Z}_+^{|V(G)|}$ we define both an instance (G, w, c) of a network bargaining game and a corresponding instance (N, v) of a c-matching game. The following theorem found in [6] establishes a connection between solution concepts of the two games. Recall that given a solution (M, z) to the network bargaining game (G, w, c) we define the the corresponding allocation x for the c-matching game (N, v) as

$$x_u = \sum_{(u,v)\in\delta(u)} z_{uv},$$

for all $u \in V$.

The following result is shown in [6].

Theorem 2.4.8. *Let (M, z) be a solution to the network bargaining game (G, w, c) and x the corresponding allocation for the c-matching game (N, v). Then*

(i) (M, z) *stable* $\Rightarrow x \in$ *core*(N, v)

(ii) (M, z) *balanced* $\Rightarrow x \in$ *core*$(N, v) \cap$ *prekernel*(N, v)

In addition if we consider the unit capacity network bargaining game (G, w) and the corresponding matching game (N, v) then the converse is also true [6].

Theorem 2.4.9. *Let (M, z) be a solution to the network bargaining game with unit capacities (G, w) and x the corresponding allocation for the matching game (N, v). Then*

(i) (M, z) *stable* $\Leftrightarrow x \in$ *core*(N, v)

(ii) (M, z) *balanced* $\Leftrightarrow x \in$ *core*$(N, v) \cap$ *prekernel*(N, v)

Theorem 2.4.9 implies that in order to compute a balanced solution to a network bargaining game with unit capacities, one can use the existing algorithm of Faigle et al. [31] for computing allocations in the intersection of the core and prekernel of the corresponding matching game. This approach offers an alternative proof of the results for unit capacity network bargaining games found in [55].

2.5 Stable matchings

In this thesis we study both two-dimensional (2D) and three-dimensional (3D) stable matchings. In the 2D setting we have two sets of agents that we denote by B and C. The set B will be referred to as the set of men and the set C as the set of women. We assume throughout this thesis that we have the same number of men and women, usually denoted by n.

Definition 2.5.1. *A* 2D matching *is a set of ordered pairs from $N \subseteq B \times C$ such that each $q \in B \cup C$ appears in at most one pair in N. If each $q \in B \cup C$ appears in exactly one pair in N then N is said to be a* perfect matching*. For any $b \in B$ we let $N(b)$ denote the woman in C that b is matched to in N, and similarly for any $c \in C$ we let $N(c)$ denote the man in B that c is matched to in N.*

In stable matching problems, each agent q has preferences over some subset of agents S. In the 2D setting the men have preferences over the women and the women over the men. The preferences of an agent q over a S can be expressed using a *preference list* $P(q)$ that lists the agents in S in non-increasing order of preference according to q. We say that $P(q)$ is a *strict ordering* if it does not contain any ties. Given $s, s' \in S$ we write $s \succ_q s'$ to denote that q strictly prefers s to s' and $s \succeq_q s'$ to denote that q prefers s at least as much as s'.

2.5.1 Stable Marriage

We introduce here the classical stable marriage problem.

Definition 2.5.2. *An instance $\mathcal{I}$ of the* SM *problem consists of a set B of n men and a set C of n women. Each man $b \in B$ has a preference list $P(b)$ that is a strict ordering over the set of women C. Similarly each woman $c \in C$ has a preference list $P(c)$ that is a strict ordering over the set of men B. A pair (b, c) is blocking with respect to a matching N if*

$$c \succ_b N(b) \text{ and } b \succ_c N(c).$$

A matching N is stable if there are no blocking pairs with respect to N. In that case, we also say that N is a solution to $\mathcal{I}$.

For the stable marriage problem, any stable matching must be a perfect matching, since any pair of unmatched agents (b, c) would form a blocking pair. The following is a famous result of Gale and Shapley [37].

Theorem 2.5.3. *There exists a stable matching for any instance of* SM *and such a matching can be computed in polynomial time.*

The proof of this result is via the well-known Gale-Shapley algorithm that is described in 2.5.1.

Algorithm 1 The Gale-Shapley algorithm for the Stable Marriage Problem

```
Initially all men and all women are single
while there exists a single man who has not proposed to every woman do
  choose one such man b
  let c be the most preferred woman on b's list to whom b has not proposed
  b proposes to c
  if c is single then
    (b, c) become engaged
  else
    if c is engaged to b′ and she prefers b to b′ then
      (b, c) become engaged
      c rejects b′
      b′ becomes single
  else
    c rejects b
    b remains single
return the set of engaged pairs
```

The algorithm terminates since a man can propose to the same woman at most once. It is not too hard to see that the set of pairs returned by the algorithm forms a perfect matching N. To see that N is indeed stable suppose by contradiction that we have a blocking pair (b, c). This means that b prefers woman c over $N(b)$. Since men propose to women in order of preference, b must have proposed to c before $N(b)$. Hence c must have rejected b at some point in the algorithm. This would have only happened if c received a proposal from some man c' that she prefers to c. But since each woman ends up engaged with the highest ranked man that ever proposed to her, c must prefer $N(c)$ over b. Hence (b, c) cannot be a blocking pair.

An alternative way to proof Theorem 2.5.3 is using a polyhedral approach. Vande Vate [79] initiated the study of the stable marriage problem from the mathematical programming perspective by introducing the following integer programming formulation.

$$
\begin{aligned}
(I_{SM}) \qquad \sum_{c} x_{bc} &= 1 && \text{for all } b \in B \\
\sum_{b} x_{bc} &= 1 && \text{for all } c \in C \\
x_{bc} + \sum_{c' \succ_b c} x_{bc'} + \sum_{b' \succ_c b} x_{b'c} &\geq 1 && \text{for all } b \in B,\ c \in C \\
x_{bc} &\in \{0, 1\} && \text{for all } b \in B,\ c \in C.
\end{aligned}
$$

The interpretation of the variables is that $x_{bc} = 1$ if b is matched to c and $x_{bc} = 0$ otherwise. The first two constraints ensure that the result is perfect matching. The third constraint can be interpreted as follows: whenever $x_{bc} = 0$, meaning that b is not matched to c, at least one of the two sums in the constraint has to be equal to one. Now if this is the first sum then b is matched to someone he prefers over c, while if it is the second sum then c is matched to someone she prefers over b. Hence the result is a stable matching. Now consider the linear programming formulation obtained by relaxing the integer constraints.

$$
\begin{aligned}
(P_{SM}) \qquad \sum_{c} x_{bc} &= 1 && \text{for all } b \in B \\
\sum_{b} x_{bc} &= 1 && \text{for all } c \in C \\
x_{bc} + \sum_{c' \succ_b c} x_{bc'} + \sum_{b' \succ_c b} x_{b'c} &\geq 1 && \text{for all } b \in B,\ c \in C \\
x_{bc} &\geq 0 && \text{for all } b \in B,\ c \in C.
\end{aligned}
$$

Vande Vate [79] proved the following result.

Theorem 2.5.4. *The polytope P_{SM} is the convex hull of the stable marriage solutions.*

This result is particularly interesting because it means that one can efficiently optimize over the set of stable matchings.

2.5.2 Stable marriage with ties and incomplete lists

The most popular variant of the classical stable marriage problem is obtained by allowing ties in the preference lists and unacceptable partners.

Definition 2.5.5. *An instance $\mathcal{I}$ of* SMTI *consists of a set B of n men and a set C of n women. Each man $b \in B$ has a preference list $P(b)$ that is an ordering over a subset of C and is allowed to contain ties. Similarly each woman $c \in C$ has a preference list $P(c)$ that is an ordering over a subset of B and is also allowed to contain ties. A pair (b, c) is said to be acceptable if b appears in $P(c)$ and c appears in $P(b)$. A pair (b, c) is blocking with respect to a matching N if (b, c) is an acceptable pair and it satisfies the usual blocking conditions*

$$c \succ_b N(b) \text{ and } b \succ_c N(c).$$

A matching N is stable if every pair in N is acceptable and there are no blocking pairs with respect to N. In that case, we also say that N is a solution to $\mathcal{I}$.

We follows the usual assumption is that a woman c is acceptable to a man b if and only if man b is acceptable to woman c. One can modify the Gale-Shapley algorithm slightly to account for ties and incomplete lists. Specifically, all ties are broken arbitrarily and men propose only to acceptable women. With this modification, the algorithm always produces a stable matching [47], hence we have the following result.

Theorem 2.5.6. *There exists a stable matching for any instance* SMTI *and such a matching can be computed in polynomial time.*

However, unlike the case of SMinstances, an instance of SMTIcan admit stable solutions of different size. We will need the following result from [62].

Theorem 2.5.7. *Let $\mathcal{I}$ be an instance of* SMTI *where ties occur only in the preference lists of the women. Deciding whether $\mathcal{I}$ admits a perfect stable matching is NP-hard.*

2.5.3 Three-dimensional stable matchings

In the 3D setting, we have an additional set of n agents that we denote by A, usually referred to as dogs.

Definition 2.5.8. *A* 3D matching *$\mathcal{M}$ is a set of ordered triples from $A \times B \times C$ such that each $q \in A \cup B \cup C$ appears in at most one triple in $\mathcal{M}$. If each $q \in A \cup B \cup C$ appears in exactly one triple then $\mathcal{M}$ is said to be a* perfect matching. *For every dog $a \in A$ we denote by $\mathcal{M}(a)$ the man that a is matched to in $\mathcal{M}$. Similarly for every man $b \in B$, $\mathcal{M}(b)$ denotes the woman that b is matched to in $\mathcal{M}$, and for every woman $c \in C$, $\mathcal{M}(c)$ denotes the dog that c is matched to in $\mathcal{M}$*

A 3D perfect matching $\mathcal{M}$ can be induced by fixing two perfect matchings on any of the following sets $A \times B$, $B \times C$ or $C \times A$.

Definition 2.5.9. *Let M be a perfect matching on $A \times B$ and N a perfect matching on $B \times C$. Then $M \circ N$ is the 3D matching defined as: $(a, b, c) \in M \circ N$ if and only if $(a, b) \in M$ and $(b, c) \in N$. For each $q \in A \cup B$ we denote by $M(q)$ the partner of q in M, and similarly for each $q \in B \cup C$ we denote by $N(q)$ the partner of q in N.*

We can now introduce the 3D stable matching problem that we focus on in this thesis.

Definition 2.5.10. *An instance of the Three-Dimensional Stable Matching with Cyclic Preferences problem (Cyclic 3DSM) consists of three mutually disjoint sets A, B, C of n elements each. Each $a \in A$ has preferences over the set B, each $b \in B$ has preferences over the set C, and each $c \in C$ has preferences over the set A. Preferences are expressed in terms of a strict total order over all the elements of the corresponding set. A triple (a, b, c) is said to be* blocking *with respect to a matching $\mathcal{M}$ if $(a, b, c) \notin \mathcal{M}$ and a prefers b to $\mathcal{M}(a)$, b prefers c to $\mathcal{M}(b)$ and c prefers a to $\mathcal{M}(c)$. A matching which has no blocking triples is said to be* stable.

We point out that our definition of a blocking triple implies that in order for (a, b, c) to be blocking with respect to a matching $\mathcal{M}$, no two agents from this triple can be matched together in $\mathcal{M}$. Our definition of stability is also referred to as *weak stability*. If instead we defined a triple to be blocking by requiring that each member of the triple prefers this triple *at least as much* as the one assigned to them in the matching and at least one member prefers it strictly, this would give rise to a notion of stability known as *strong stability*. The following result appears both in [8] and [44].

Theorem 2.5.11. *Deciding whether an instance of* c3DSM *admits a strongly stable matching is NP-complete.*

The question of whether every instance of c3DSM admits a weakly stable matching is still open. Since in this thesis we only consider weakly stable matchings, we refer to them from now as stable matchings.

Chapter 3

Network bargaining

Let (G, w, c) be an instance of a network bargaining game with general capacities defined in section 2.4. We denote by (N, v) the corresponding c-matching game defined in section 2.3.6. In this chapter we study two problems regarding balanced solutions for (G, w, c). First, we want to characterize when an instance admits a balanced solution, and second we want to develop an efficient algorithm for computing such a solution.

3.1 Connection to cooperative games

Our first attempt relies on exploring a connection between the solution concepts of network bargaining games and cooperative games. For the special class of unit capacity and constrained bipartite games, Bateni et al. show that the set of stable solutions corresponds to the core, and the set of balanced solutions corresponds to the intersection of the core and prekernel of the associated matching game. This implies that efficient algorithms, such as the one of [31], can be used to compute points in the intersection of the core and prekernel from which a balanced solution can be uniquely obtained. Hence, a natural question to answer is whether such a connection can be established in general. For general capacities we already know from Theorem 2.4.8 that one direction of the equivalence still holds: every balanced solution gives rise to an allocation in the intersection of the core and prekernel.

Hence, the remaining question is whether the converse is also true: can every allocation in the intersection of the core and prekernel be obtained from a balanced solution? The following lemma proves that this is not always the case.

Lemma 3.1.1. *There exists an instance* (G, w, c) *of the network bargaining game and a vector* $x \in core(N, v) \cap prekernel(N, v)$ *such that there exists no balanced solution* (M, z) *satisfying*

$$x_u = \sum_{v:(u,v)\in E} z_{uv}$$

for all $v \in V$.

Proof. Consider the graph shown in Figure 3.1 where every vertex has capacity 2 and the edge weights are given above each edge. The red edges denote the unique optimal 2-matching of weight 120.

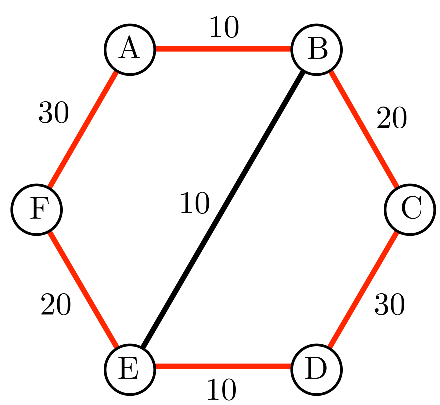

Figure 3.1: An instance of a network bargaining game with general capacities.

Let x be the allocation that assigns a value of 20 to each vertex in the graph. This is a valid allocation since the total value assigned to all the vertices is equal to 120, the weight of the maximum weight 2-matching of the graph. We now show that the vector x is in the intersection of the core and prekernel but there exists no balanced solution (M, z) corresponding to x. We start by showing that x is in the core. To do this, we must check that $x(S) \geq v(S)$ for every subset of vertices S, where $v(S)$ is the weight of the maximum weight 2-matching in $G[S]$. Note that for any subset S of vertices we have $x(S) = 20|S|$. On the other hand it can easily checked that for any subset of vertices S the weight of the maximum weight 2-matching in $G[S]$ is at most $20|S|$. Hence $x \in \text{core}(N, v)$.

Next we show that $x \in \text{prekernel}(N, v)$. To do this we must compute the powers

$$s_{uv}(x) = \max_{T: u \in T, v \notin T} v(T) - x(T)$$

for all pairs of vertices $u, v \in V(G)$. Let C_1 denote the outer cycle on vertices A, B, C, D, E, F and let C_2 and C_3 denote the inner cycles on vertices B, C, D, E and E, F, A, B respectively. For the pair A, B we have:

$$s_{AB} = v(\{A, F\}) - x(\{A, F\}) = 30 - 40 = -10$$
$$s_{BA} = v(\{C_2\}) - x(\{C_2\}) = 70 - 80 = -10$$

Similarly for the pair B, C we have:

$$s_{BC} = v(\{C_3\}) - x(\{C_3\}) = 70 - 80 = -10$$
$$s_{CB} = v(\{C, D\}) - x(\{C, D\}) = 30 - 40 = -10.$$

And for the pair C, D:

$$s_{CD} = v(\{C\}) - x(\{C\}) = -20$$
$$s_{DC} = v(\{D\}) - x(\{D\}) = -20.$$

Hence the pairs $(A, B), (B, C)$ and (C, D) satisfy the prekernel condition. By symmetry so do $(D, E), (E, F)$ and (F, A). Now for any pair u, v of non-adjacent vertices, one of the two cycles C_2 or C_3 will contain u but not v, and vice versa. Therefore $s_{uv} = s_{vu} = -10$ for all non-adjacent pairs u, v. This proves that x is in the prekernel.

We now show that there is no vector z such that (M, z) is a balanced solution corresponding to the vector x. Since any stable, and therefore balanced, solution must occur on a maximum weight 2-matching [6] of the graph, any balanced solution (M, z) will have $M = E(C_1)$. First note that vertices A, F, C and D have an outside option of zero in any solution, since there are no edges in $E \backslash M$ incident with these vertices. Hence the contracts (C, D) and (A, F) have to be split evenly in any balanced solution. Since each vertex must have a total profit of 20 from its two contracts in M in order for (M, z) to correspond to x, this uniquely determines all values of the vector z, as are shown in Figure 3.2.

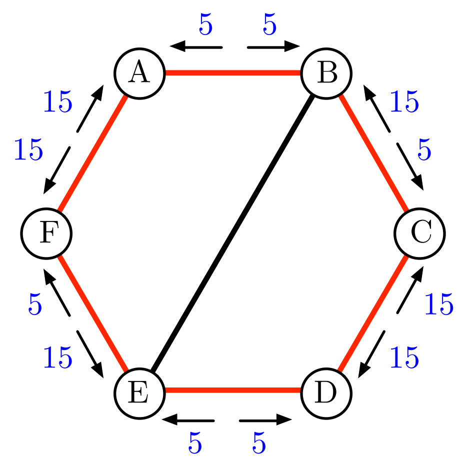

Figure 3.2: An allocation in $\text{core}(N, v) \cap \text{prekernel}(N, v)$ with no corresponding balanced solution.

The minimum contract of both B and E is 5 and therefore $\alpha_B = \alpha_E = 10 - 5 = 5$. However, the edge (B, C) (and also the edge (E, F) by symmetry) violates the balance condition since $z_{BC} - \alpha_B = 15 - 5 = 10$ while $z_{CB} - \alpha_C = 5 - 0 = 5$. This proves that there is no balanced solution corresponding to the allocation x. □

We remark that the example used in the above lemma does possess a balanced solution, as shown in Figure 3.3. Here the outside option of both B and E is 10/3 and all edges in the matching satisfy the balance condition. The allocation corresponding to this balanced solution is

$$x'_u = \begin{cases} \frac{55}{3} & u \in \{A, B, D, E\} \\ \frac{70}{3} & u \in \{C, F\} \end{cases}$$

Theorem 2.4.8 implies that x' should be in $\text{core}(N, v) \cap \text{prekernel}(N, v)$ and this is indeed the case.

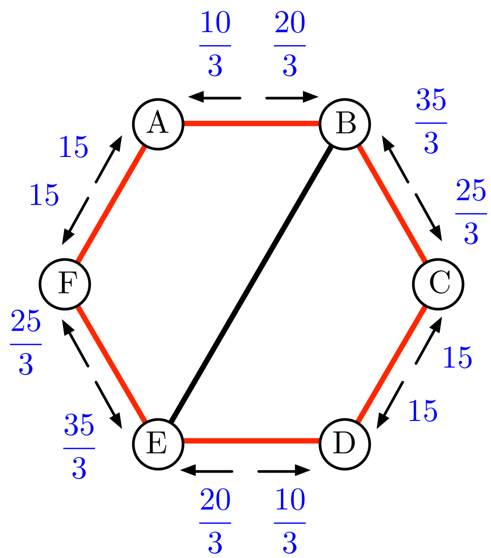

Figure 3.3: A balanced solution.

In view of lemma 3.1.1, we cannot hope to extend the correspondence between balanced solutions and allocations in the intersection of the core and prekernel to all network bargaining games. However we can generalize the results of [6] by characterizing a larger class of network bargaining games, including unit capacity and constrained bipartite games, for which this correspondence holds. We achieve this by defining a certain gadget whose absence, together with the fact that the c-matching M is acyclic, will be sufficient for the correspondence to hold.

3.1.1 Acyclic c-matchings and gadgets

In this section we characterize some sufficient conditions under which every allocation in the intersection of the core and prekernel corresponds to a balanced outcome. Our first condition requires that the maximum weight c-matching M is acyclic.

We can now introduce the definition of a gadget.

Definition 3.1.2. *Let (M, z) be a solution and let u be a vertex covered by M such that $\alpha_u > 0$. Let v be a neighbour of u in M and let v' be the vertex corresponding to u's best outside option. If v' is saturated in M, let u' be its weakest contract. A* gadget *is set of edges $P \subseteq M$ satisfying one of the following two conditions:*

(i) P is a path from v to v' that does pass through u

(ii) P is a path from u to u' that does not pass through v'.

. If there exists such a gadget with respect to vertex u in the solution (M, z) we say that u is a bad *vertex this solution.*

Figure 3.4 shows the two possible types of gadgets. The red edges denote edges in M and the black edges denote edges in $E \backslash M$.

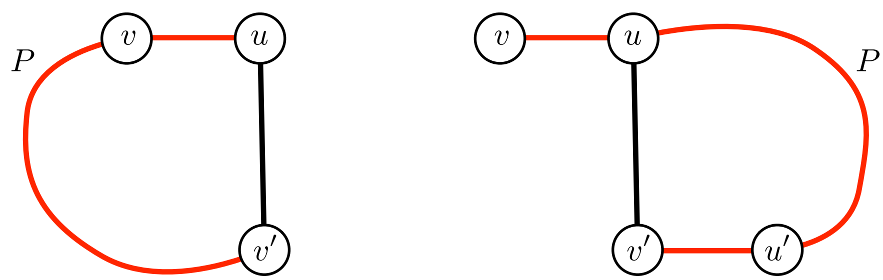

Figure 3.4: The two types of gadgets.

We can now state our main theorem of this section.

Theorem 3.1.3. *Let (G, w, c) be an instance of the network bargaining game and (N, v) the corresponding c-matching game. Let $x \in core(N, v)$ be a core allocation and (M, z) be a corresponding stable solution so that*

$$x_u = \sum_{v:(u,v) \in E} z_{uv}$$

for all $u \in V(G)$. If the following two conditions are satisfied

1. *M is acyclic,*
2. *there are no bad vertices in the solution (M, z),*

then (M, z) is a balanced solution if and only if $x \in$ prekernel(N, v).

Proof. Fix an edge $(u, v) \in M$. Note that it suffices to show $s_{uv} = -z_{uv} + \alpha_u$, since this would imply that $s_{uv} = s_{vu}$ if and only if $z_{uv} - \alpha_u = z_{vu} - \alpha_v$. Our strategy is to first show that s_{uv} is upper bounded by $-z_{uv} + \alpha_u$, after which it will be sufficient to find a set T so that $v(T) - x(T)$ achieves this upper bound. We start with the following lemma.

Lemma 3.1.4. *$s_{uv} \leq -z_{uv} + \alpha_u$.*

Proof. Let $T \subset N$ such that $u \in T$ and $v \notin T$. Let M' be a maximum weight c-matching in $G[T]$. Then

$$\begin{aligned} v(T) - x(T) &= w(M') - \sum_{a \in T} x_a \\ &= w(M') - \sum_{a \in T, ab \in M} z_{ab} \\ &= \left(w(M' \cap M) - \sum_{a \in T,\, ab \in M \cap M'} z_{ab} \right) + \left(w(M' \backslash M) - \sum_{a \in T,\, ab \in M \backslash M'} z_{ab} \right) \\ &= w(M' \backslash M) - \sum_{a \in T,\, ab \in M \backslash M'} z_{ab}. \end{aligned}$$

Define the set of ordered pairs

$$\mathcal{S} := \{(a, b) : a \in T,\, b \in V(G),\, ab \in M \backslash M'\},$$

so that

$$v(T) - x(T) = w(M' \backslash M) - \sum_{(a,b) \in \mathcal{S}} z_{ab}. \tag{3.1}$$

Since (M, z) is a stable solution it follows that M is a c-matching of maximum weight. Hence any edge in $M' \backslash M$ must have at least one saturated endpoint. Let $\{a_1, \cdots, a_\ell\}$ be the set of vertices in T that are saturated in M. For each $i \in [\ell]$ define the sets of ordered pairs

$$\mathcal{E}_i := \{(a_i, b) : a_i b \in M' \backslash M\}$$
$$\mathcal{F}_i := \{(a_i, c) : a_i c \in M \backslash M'\}.$$

Note that all these sets are pairwise disjoint, and $\mathcal{F}_i \subset \mathcal{S}$ for all $i \in [\ell]$. Now since each a_i is saturated in M it follows that $|\mathcal{E}_i| \leq |\mathcal{F}_i|$. Therefore we can fix an arbitrary mapping

$$\phi_i : \mathcal{E}_i \to \mathcal{F}_i$$

so that each element of $\mathcal{E}_i$ is mapped to a distinct element of $\mathcal{F}_i$.

We now assign to each edge $e \in M' \backslash M$ a set of ordered elements $\mathcal{S}_e \subset \mathcal{S}$. As previously observed, each such edge e must have at least one saturated endpoint. Hence let $e = (a_i, y)$ and define:

$$\mathcal{S}_e = \begin{cases} \{\phi_i(a_i, a_j), \phi_j(a_j, a_i)\} & \text{if } y = a_j \text{ for some } j \in [\ell] \backslash \{i\}, \\ \{\phi_i(a_i, y)\} & \text{otherwise.} \end{cases}$$

It follows from the definition of the sets $\mathcal{E}_i$ and the choice of the mapping ϕ_i that the sets $\mathcal{S}_e$ are well defined, are pairwise disjoint, and are all subsets of $\mathcal{S}$. Let

$$\mathcal{S}' := \mathcal{S} \backslash \left(\cup_{e \in M' \backslash M} \mathcal{S}_e \right),$$

Then from equation (3.1) and the fact that each ordered pair of $\mathcal{S}$ belongs to exactly one $\mathcal{S}_e$ set we obtain

$$v(T) - x(T) = \sum_{e \in M' \backslash M} \left(w_e - \sum_{(a,b) \in \mathcal{S}_e} z_{ab} \right) - \sum_{(a,b) \in \mathcal{S}'} z_{ab}. \tag{3.2}$$

Now it follows from stability that for all $e \in M' \backslash M$ we have

$$w_e \leq \sum_{(a,b) \in \mathcal{S}_e} z_{ab}. \tag{3.3}$$

To see this, consider $e = (a_i, y) \in M'\backslash M$. If $y = a_j$ for some $j \in [\ell]\backslash\{i\}$ then $\mathcal{S}_e := \{\phi_i(a_i, a_j), \phi_j(a_j, a_i)\}$. Now $\phi_i(a_i, a_j)$ represents the profit that a_i gets from one of his contracts in M, and similarly $\phi_j(a_j, a_i)$ represents the profit that a_j earns from one of his contracts in M. Since the edge (a_i, a_j) is not in M, by stability we must have $\phi_i(a_i, a_j) + \phi_j(a_j, a_i) \geq w_{a_i y}$. In the other case where y is not a saturated vertex in T, we have $\mathcal{S}_e = \{\phi_i((a_i, y))\}$. Since y is an outside option for a_i and $\phi_i(a_i, y)$ represents the profit that a_i gets from one of his contracts in M, stability for vertex a_i implies that $\phi_i(a_i, y) \geq w_{a_i y}$ as required.

Now suppose that $(u, v) \notin \cup_{e \in M'\backslash M} \mathcal{S}_e$. Then from (3.2) and (3.3) we have

$$v(T) - x(T) \leq -z_{uv}.$$

Since $\alpha_u \geq 0$, this proves the lemma in this case. If on the other hand, there exists a set $\mathcal{S}_{e^*}$ such that $(u, v) \in \mathcal{S}_{e^*}$. Then using (3.2) and (3.3) again we have

$$v(T) - x(T) \leq -z_{uv} + w_{e^*} - \sum_{(a,b) \in \mathcal{S}_{e^*}\backslash\{(u,v)\}} z_{uv},$$

Now $(u, v) \in \mathcal{S}_{e^*}$ implies that e^* must be an edge in $M'\backslash M$ that is incident to vertex u. Hence $e^* = (u, w)$ where w is an outside option for vertex u. If w is not saturated then the set $\mathcal{S}_{e^*}\backslash\{(uv)\}$ is empty. Otherwise if w is saturated, this set contains a unique ordered pair (w, k) such that $(w, k) \in M\backslash M'$. Therefore it follows from the definition of α_u that

$$v(T) - x(T) \leq -z_{uv} + \alpha_u,$$

as desired. □

Now that we have established this upper bound, it suffices to find a set T that achieves this bound with equality. That is, we want to find a set $T \subseteq V(G)$ such that $u \in T$, $v \notin T$ and show that $v(T) - x(T) \geq -z_{uv} + \alpha_u$. Given a set of vertices S we let M_S denote the edges of M that have both endpoints in S. Note that for any set of vertices S we have

$$w(M_S) - x(S) = - \sum_{ab \in M: a \in S, b \notin S} z_{ab}. \tag{3.4}$$

We define $\mathcal{C}$ to be the set of components of G induced by the edges in M. Since u and v are neighbours in M they will be in the same component, call it C. Now suppose we remove the edge (u, v) from C. Since M is acyclic, this disconnects C into two components C_u and C_v, containing vertices u and v respectively. Now M_{C_u} is a valid c-matching of C_u hence applying equation (3.4) to the vertex set of the component C_u we obtain

$$v(C_u) - x(C_u) \geq w(M_{C_u}) - x(C_u) = -z_{uv}.$$

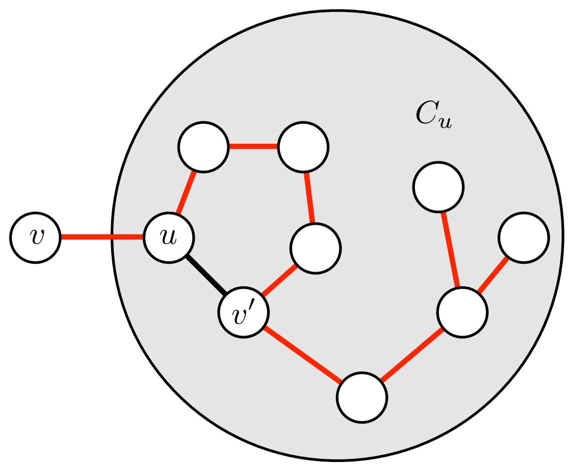

Figure 3.5: Outside option is in the same component and is not saturated

If $\alpha_u = 0$ then setting T to be the vertex set of component C_u completes the proof for this case. Hence it remains to consider the case where $\alpha_u > 0$. Then by stability of the solution (M, z) vertex u must be saturated in M. Let v' be vertex u's best outside option.

Case 1: $v' \in C_u$ and v' is not saturated in M (see Figure 3.5). Since $uv \notin C_u$ and v' is not saturated in M the set of edges $M_{C_u} \cup \{(u, v')\}$ is a valid c-matching of C_u and therefore

$$\begin{aligned} v(C_u) - x(C_u) &\geq w(M_{C_u} \cup \{(u, v')\}) - x(C_u) \\ &= w(M_{C_u}) - x(C_u) + w_{uv'} \\ &= -z_{uv} + w_{uv'} && \text{applying (3.4) to } C_u. \end{aligned}$$

Case 2: $v' \in C_u$ and v' is saturated in M (see Figure 3.6). Let u' be the weakest contract

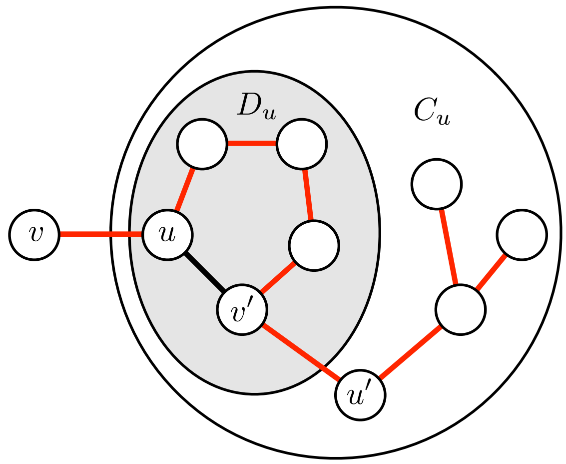

Figure 3.6: Outside option is in the same component and is saturated

of v' and suppose we remove the edge (v', u') from C_u. Since M is acyclic, this disconnects C_u into two components. From condition (2) we know that u' is not on the $u - v'$ path in M. Hence u and v' are in the same component of $C_u \setminus \{(v', u')\}$. Denote this component by D_u. Now $M_{D_u} \cup \{(u, v')\}$ is a c-matching of D_u and thus

$$\begin{aligned} v(D_u) - x(D_u) &\geq w\left(M_{D_u} \cup \{(u, v')\}\right) - x(D_u) \\ &= w(M_{D_u}) - x(D_u) + +w_{uv'} \\ &= -z_{uv} - z_{v'u''} + +w_{uv'} && \text{applying (3.4) to } D_u \\ &= -z_{uv} + \alpha_u && \text{by choice of } v' \text{ and } u'. \end{aligned}$$

Case 3: $v' \notin C_u$ and v' is not saturated in M (see Figure 3.7). Condition (2) implies that there is no $v - v'$ path in M, hence the fact that $v' \notin C_u$ implies that $v' \notin C$. Let $C_{v'}$ be the component in $\mathcal{C}$ that contains vertex v'. Now $M_{C_u} \cup M_{C_{v'}} \cup \{(u, v')\}$ is a c-matching

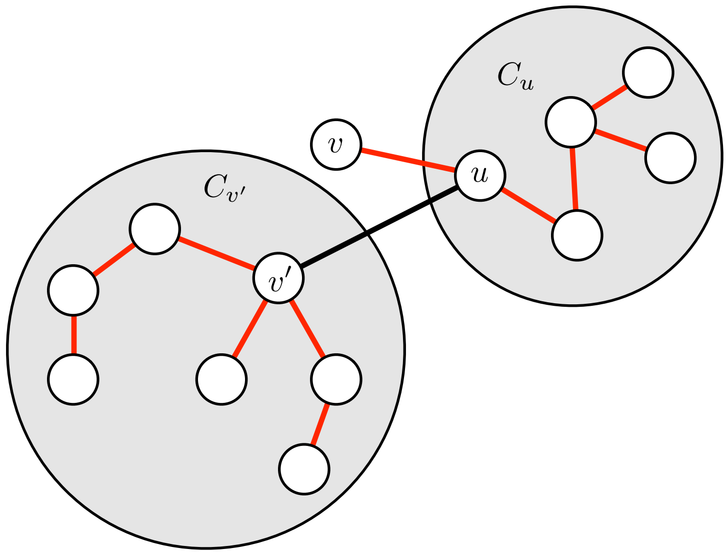

Figure 3.7: Outside option is not in the same component and is not saturated

of $C_u \cup C_{v'}$ and therefore

$$\begin{aligned} v(C_u \cup C_{j'}) - x(C_u \cup C_{v'}) &\geq w(M_{C_u} \cup M_{C_{v'}} \cup \{(u, v')\}) - x(C_u \cup C_{v'}) \\ &= (w(M_{C_u}) - x(C_u)) + \big(w(M_{C_{v'}}) - x(C_{v'})\big) + w_{uv'} \\ &= -z_{uv} + w_{uv'} \qquad\qquad \text{applying (3.4) to } C_u,\ C_{v'} \\ &= -z_{uv} + \alpha_u \qquad\qquad \text{by choice of } v'. \end{aligned}$$

Case 4: $v' \notin C_u$ and v' is saturated in M (see Figure 3.8). As in Case 3 we let $C_{v'}$ be the component in $\mathcal{C}$ that contains vertex v'. Let u' be the weakest contract of v' and suppose we remove the edge (v', u') from $C_{v'}$. Since M is acyclic, this disconnects $C_{v'}$ into two components and we let $D_{v'}$ be the one that contains vertex v'. Now $M_{C_u} \cup M_{D_{v'}} \cup \{(u, v')\}$

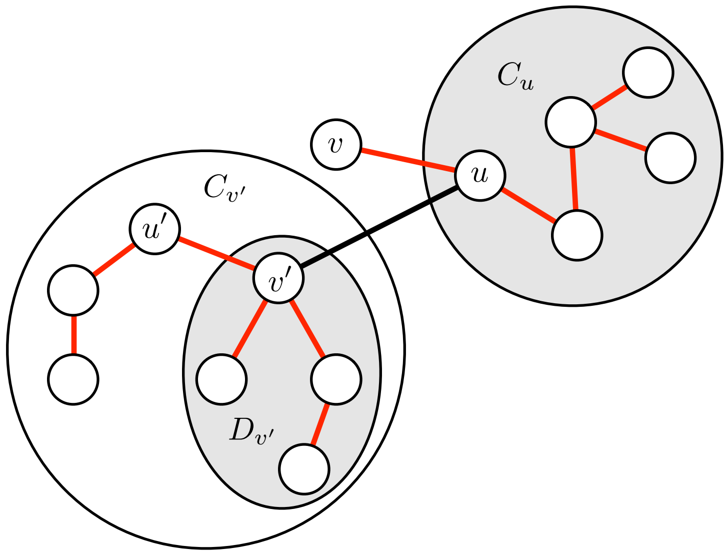

Figure 3.8: Outside option is not in the same component and is saturated

is a c-matching of $C_u \cup C_{v'}$ thus

$$\begin{aligned} v(C_u \cup D_{v'}) - x(C_u \cup D_{v'}) &\geq w\left(M_{C_u} \cup M_{D_{v'}} \cup \{(u, v')\}\right) - x\left(C_u \cup C_{v'}\right) \\ &= \left(w(M_{C_u}) - x(C_u)\right) + \left(w(M_{D_{v'}}) - x(D_{v'})\right) + w_{uv'} \\ &= -z_{uv} - z_{v'u'} + w_{uv'} \qquad \text{applying (3.4) to } C_u\text{ , } D_{v'} \\ &= -z_{uv} + \alpha_u \qquad\qquad\qquad \text{by choice of } v' \text{ and } u'. \end{aligned}$$

Hence in all cases we have $v(T) - x(T) \geq -z_{uv} + \alpha_u$ as required. This completes the proof of the theorem. □

We note that all network bargaining games studied in [6] satisfy conditions (1) and (2) of Theorem 3.1.3. In addition to these, Theorem 3.1.3 also covers the case of network bargaining games where the underlying graph is a tree, but where each vertex can have an arbitrary capacity. Hence one can use the polynomial time algorithm of [31] to compute a point in the intersection of the core and prekernel for these games from which we can obtain a corresponding solution (M, z) that is guaranteed to be balanced by Theorem 3.1.3.

3.2 Reduction to unit capacity games

While we were able to generalize the class of network bargaining games for which balanced solutions can be obtained by computing a point in the intersection of the core and prekernel, we were not able to apply this technique to all network bargaining games. In this section we show that balanced solutions can be obtained to any network bargaining game (G, w, c) by a reduction to a unit capacity game defined on an auxiliary graph.

Suppose we are given an instance (G, w, c) of the network bargaining game together with a c-matching M of G. Algorithm 3.2 describes how to obtain an instance (G', w') of the unit capacity game together with a matching M' of G.

Algorithm 2 Mapping from general capacities to unit capacities.

Input: An instance (G, w, c) with general capacities and a c-matching M
Output: An instance (G', w') with unit capacities and a matching M'
  for $u \in V(G)$ **do**
    Fix an arbitrary labelling $\sigma_u : \{v : uv \in M\} \to \{1, \cdots, d(u)\}$
    Create c_u copies $u_1, \cdots, u_{c_u}$ in $V(G')$
  for $(u, v) \in E(G)$ **do**
    if $(u, v) \in M$ **then**
      add the unique edge $(u_{\sigma_v(u)}, v_{\sigma_u(v)})$ to $E(G') \cap M'$ and set its weight to w_{uv}
    else
      add all edges (u_i, v_j) to $E(G')$ for $i \in [c_u]$ and $j \in [c_v]$, and set their weights to w_{uv}
  return (G', w', M')

Figure 3.9 illustrates an example of this construction. All edges have unit weight and the capacities are listed next to each vertex. The red edges denote the unique maximum cardinality c-matching M. The first step is make copies of each vertex according to its capacity. Hence we make four copies of u , two copies of x and y, and one copy of every other node. This defines the vertex set of G'. We then map each edge in M to a unique copy of itself in M'. The edge (u, x) is mapped to (u_1, x_1), the edge (x, y) to (x_2, y_1), the edge (u, y) to (u_2, y_2) and finally the edge (u, z) to (u_3, z_1). Note that M' has exactly the same number of edges as M and each edge in M' has weight equal to its corresponding

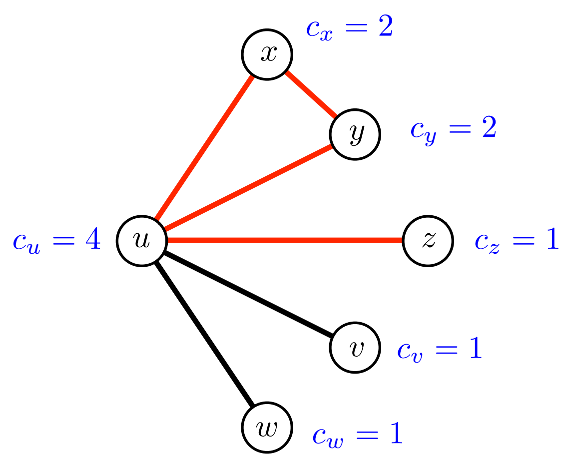

Figure 3.9: An instance with general capacities

edge in M. It remains to consider the edges (u, v) and(u, w) that are part of the original graph G but were not in the c-matching M. For these edges we create all the possible copies connecting their corresponding endpoints but do not include them in the matching M. Hence (u, v) maps to $c_u c_v$ copies of itself, and (u, w) to $c_u c_w$ copies of itself. Finally, each edge in G' has weight equal to its corresponding edge in G. The resulting graph is shown in Figure 3.10.

The important idea behind our reduction is the fact that we create only one copy of each edge that belongs to the original c-matching M and the resulting set of edges M' form a matching in the new graph. This is exactly what distinguishes our reduction from the standard reduction of a c-matching to a matching where $c_u c_v$ copies of each edge are created.

3.2.1 Mapping between the two solution sets

Suppose we are given an instance of the network bargaining game (G, w, c) with a c-matching M. Let $[(G', w'), M']$ be obtained using the construction described in algorithm 3.2. Note that M and M' have the same number of edges and each edge $(u, v) \in M$ is mapped to

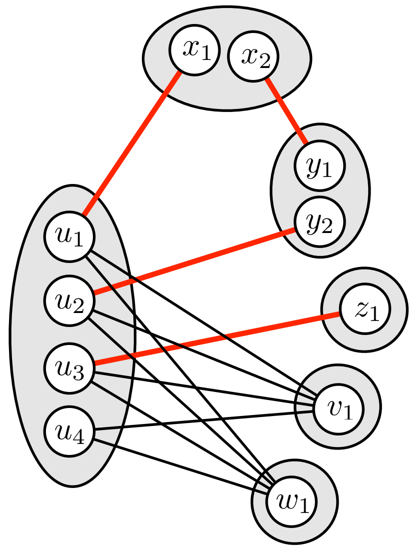

Figure 3.10: The corresponding instance with unit capacities.

the unique edge $(u_i, v_j) \in M'$ where $i = \sigma_v(u)$ and $j = \sigma_u(v)$. This allows us to go back and forth between solutions on M and M' by dividing the weight of each edge in the same way as its corresponding pair.

We define the two solution sets:

$$\mathcal{X} := \left\{ x \in \mathbb{R}^{|V(G')|} : \ (M', x) \text{ is a solution to } (G', w') \right\}$$
$$\mathcal{Z} := \left\{ z \in \mathbb{R}^{2|E(G)|} : (M, z) \text{ is a solution to } (G, w, c) \right\}.$$

And the two mappings:

1. $\phi : \mathcal{X} \to \mathcal{Z}$

   For all $(u, v) \in E$ define

$$(\phi(x)_{uv}, \phi(x)_{vu}) := \begin{cases} \left(x_{u_{\sigma_v(u)}}, x_{v_{\sigma_u(v)}} \right) & \text{if } uv \in M, \\ (0, 0) & \text{otherwise.} \end{cases} \tag{3.5}$$

2. $\phi^{-1} : \mathcal{Z} \to \mathcal{X}$.

For all $u_i \in V(G')$ define

$$\phi^{-1}(z)_{u_i} := \begin{cases} z_{uv} & \text{if } i = \sigma_u(v), \\ 0 & \text{otherwise.} \end{cases} \tag{3.6}$$

Note that $z = \phi(x)$ if and only if $x = \phi^{-1}(z)$. The following lemma implies that the mapping given by the function ϕ and its inverse ϕ^{-1} defines a bijection.

Lemma 3.2.1. *Let ϕ and ϕ^{-1} be defined as in 3.5 and 3.6. Then the following are true.*

1. *If $x \in \mathcal{X}$ and $z = \phi(x)$, then $z \in \mathcal{Z}$.*
2. *If $z \in \mathcal{Z}$ and $x = \phi^{-1}(z)$ then $x \in \mathcal{X}$.*

Proof. Let $x \in \mathcal{X}$ and $z = \phi(x)$. We show that $z \in \mathcal{Z}$. Take $(u, v) \in E(G) \cap M$. Suppose $i = \sigma_v(u)$ and $j = \sigma_u(v)$. Then from the construction of G' and M' it follows that $u_i v_j \in M'$. We have

$$\begin{aligned} z_{uv} + z_{vu} &= x_{u_i} + x_{v_j} && \text{from the definition of } \phi(x) \\ &= w_{u_i u_j} && \text{since } (M', x) \text{ is a solution} \\ &= w_{uv} && \text{from the construction of } (G', w'). \end{aligned}$$

Furthermore if $(u, v) \in E(G) \backslash M$ then from the definition of $\phi(x)$ we have $z_{uv} = z_{vu} = 0$.

Now let $z \in \mathcal{Z}$ and $x = \phi^{-1}(z)$. We show that $x \in \mathcal{X}$. Take $u_i v_j \in E(G') \cap M'$. From the construction of G' and M' there must exist an edge $(u, v) \in E(G) \cap M$ such that $i = \sigma_v(u)$ and $j = \sigma_u(v)$. We have

$$\begin{aligned} x_{u_i} + x_{v_j} &= z_{uv} + z_{vu} && \text{from the definition of } \phi^{-1}(z) \\ &= w_{uv} && \text{since } (M, z) \text{ is a solution} \\ &= w_{u_i v_j} && \text{from the construction of } (G', w'). \end{aligned}$$

Furthermore if u_i is uncovered in M' then $x_{u_i} = 0$ by definition. □

From now on we write $(M, z) \sim (M', x)$ whenever $z = \phi(x)$ or equivalently $x = \phi^{-1}(z)$. The next lemma is the key step in showing that certain properties of a solution are preserved under our mapping.

Lemma 3.2.2. *Let (G, w, c) be an instance of the network bargaining game and M a c-matching on G. Suppose the auxiliary instance (G', w') and the matching M' were obtained using the construction given in algorithm 3.2. Let (M, z) be a solution to (G, w, c) and (M', x') a solution to (G', w') such that $(M, z) \sim (M', x)$. Then for any $u \in V(G)$ and any $i \in [d_u]$ we have*

$$\alpha_u(M, z) = \alpha_{u_i}(M', x).$$

Proof. We first show that $\alpha_u(M, z) \leq \alpha_{u_i}(M', x)$. We may assume that $\alpha_u(M, z) > 0$ since otherwise there is nothing to show. Let v be vertex u's best outside option in (M, z). That is $(u, v) \in E(G) \backslash M$ and

$$\alpha_u(M, z) = w_{uv} - \mathbf{1}_{[d_v = c_v]} \min_{vw \in M} z_{vw}.$$

Since $(u, v) \in E(G) \backslash M$ we have $u_i v_j \in E(G') \backslash M'$ for all $j \in [c_v]$. We have two cases:

1. v is not saturated in M. Then the vertex v_{d_v+1} is in $V(G')$ and it is not covered by M'. Since $(u_i, v_{d_v+1}) \in E(G') \backslash M'$ we have

$$\begin{aligned}
\alpha_{u_i}(M', x) &\geq w_{u_i v_{d_v+1}} - x_{v_{d_v+1}} \\
&= w_{u_i v_{d_v+1}} && v_{d_v+1} \text{ is not covered by } M' \text{ so } x_{v_{d_v+1}} = 0 \\
&= w_{uv} && \text{from the definition of } (G', w') \\
&= \alpha_u(M, z) && \text{by choice of } v.
\end{aligned}$$

2. v is saturated in M. Let $w = \arg\min_{vw \in M} z_{vw}$. Suppose that $j = \sigma_w(v)$. Then v_j is covered in M' and $x_{v_j} = z_{vw}$. Since $(u_i, v_j) \in E(G') \backslash M'$ we have

$$\begin{aligned}
\alpha_{u_i}(M', x) &\geq w_{u_i v_j} - x_{v_j} \\
&= w_{uv} - z_{vw} && \text{from the definition of } (G', w') \\
&= \alpha_u(M, z) && \text{by choice of } v.
\end{aligned}$$

We now show that $\alpha_u(M, z) \geq \alpha_{u_i}(M', x)$. We may assume that $\alpha_{u_i}(M', x) > 0$. Let v_j be vertex u_i's best outside option in (M', x). That is, $v_j \in V(G')$ such that $(u_i, v_j) \in E(G')\backslash M'$ and

$$\alpha_{u_i}(M', x) = w_{u_i v_j} - x_{v_j}.$$

Since $(u_i, v_j) \in E(G')\backslash M'$ we must have $(u, v) \in E(G)\backslash M$. Again, we have two cases:

1. v_j is not covered in M'. Then the vertex v is not saturated in M and

$$\begin{aligned}
\alpha_u(M, z) &\geq w_{uv} & \\
&= w_{u_i v_j} & \text{from the definition of } (G', w') \\
&= w_{u_j v_j} - x_{v_j} & v_j \text{ is not covered in } M' \text{ so } x_{v_j} = 0 \\
&= \alpha_{u_i}(M', x) & \text{by choice of } v_j.
\end{aligned}$$

2. v_j is covered in M. Then there exists $w \in V(G)$ such that $(v, w) \in E(G) \cap M$ and $j = \sigma_w(v)$. We have

$$\begin{aligned}
\alpha_u(M, z) &\geq w_{uv} - z_{vw} & \\
&= w_{u_i v_j} - x_{v_j} & \text{from the definition of } (G', w') \\
&= \alpha_{u_i}(M', x) & \text{by choice of } v_j.
\end{aligned}$$

□

Lemma 3.2.2 shows that our mapping preserves outside options. This crucial property allows us to prove that stability and balance are also preserved. The following theorem is our main result.

Theorem 3.2.3. *Let (G, w, c) be an instance of the network bargaining game and M a c-matching on G. Suppose the auxiliary instance (G', w') and the matching M' were obtained using the construction given in algorithm 3.2. Let (M, z) be a solution to (G, w, c) and (M', x') a solution to (G', w') such that $(M, z) \sim (M', x)$. Then:*

1. (M, z) is stable if and only if (M', x) is stable.

2. (M, z) is balanced if and only if (M', x) is balanced.

Proof. Let $(u, v) \in M$. Suppose that $i = \sigma_v(u)$. Then $z_{uv} = x_{u_i}$ and using lemma 3.2.2 we have

$$z_{uv} \geq \alpha_u(M, z) \quad \text{if and only if} \quad x_{u_i} \geq \alpha_{u_{\sigma_v(u)}}(M', x)\,.$$

It remains to show that if (M', x) is stable then $\alpha_u(M, z) = 0$ for any unsaturated vertices u of G. Suppose u is such a vertex. Then the vertex u_{d_u+1} is not covered in M' and therefore $x'_{d_u+1} = 0$. If (M', x) is stable then $\alpha_{u_{d_u+1}} = 0$ and by lemma 3.2.2 we have $\alpha_u(M, z) = 0$ as desired. This completes the proof of the first statement.

To prove the second statement let $(u, v) \in M$ and suppose that $i = \sigma_v(u)$ and $j = \sigma_u(v)$. Then $z_{uv} = x_{u_i}$, $z_{vu} = x_{v_j}$ and by lemma 3.2.2 we have:

$$z_{uv} - \alpha_u(M, z) = z_{vu} - \alpha_v(M, z) \quad \Leftrightarrow \quad x_{u_i} - \alpha_{u_i}(M', x) = x_{v_j} - \alpha_{v_j}(M', x)\,.$$

This completes the proof. □

Using Theorem 3.2.3 we have the following algorithm for finding a balanced solution to the network bargaining game (G, w, c):

Algorithm 3 Computing balanced solutions for general capacities.

1: Find a maximum c-matching M in G
2: Obtain the unit capacity instance (G', w') with matching M' using algorithm 3.2
3: Find a balanced solution x on the matching M' in G'
4: Set $z = \phi(x)$.
5: **return** (M, z)

We note that step 3 of the algorithm can be implementing using the existing polynomial time algorithm of Kleinberg and Tardos [55]. Given any instance of a network bargaining game with unit capacities together with a maximum weight matching, their algorithm returns a balanced solution on the given matching, whenever one exists.

Theorem 3.2.3 also helps us characterize the existence of balanced solutions. Since any stable solution must occur on a c-matching, respectively matching, of maximum weight we have the following corollary.

Corollary 3.2.4. *Let (G, w, c) be an instance of the network bargaining game and M a c-matching on G. Suppose the auxiliary instance (G', w') and the matching M' were obtained using the given construction. Then*

1. *M is a maximum weight c-matching for (G, w, c) if and only if M' is a maximum weight matching for (G', w').*
2. *There exists a balanced solution for (G, w, c) on the c-matching M if and only if there exists a balanced solution for (G', w') on the matching M'.*

This leads us to the following full characterization of the existence of balanced solutions for network bargaining games with general capacities.

Theorem 3.2.5. *An instance (G, w, c) of the network bargaining game has a balanced solution if and only if it has a stable one.*

Chapter 4

Stable matchings

We start by introducing our model of stable marriage with general preferences.

Definition 4.0.6. *An instance $\mathcal{I}$ of* SMG *consists of a set B of n men and a set C of n women. Each man $b \in B$ has a preference list $P(b)$ that is complete total order over C. Each woman $c \in C$ has a preference relation given in terms of a set of ordered pairs $\mathcal{R}_c \subseteq B \times B$. A pair (b, c) is* blocking *with respect to a matching N if $(b, c) \notin N$, $c \succ_b N(b)$ and $(N(c), b) \notin \mathcal{R}_c$. A matching N is stable if it has no blocking pairs.*

Given two men $b, b' \in B$ and woman $c \in C$ we interpret $(b, b') \in \mathcal{R}_c$ as woman c preferring man b at least as much as man b'. Note that whether (b, b') is in $\mathcal{R}_c$ is completely independent of whether $(b', b) \in \mathcal{R}_c$. The definition of a blocking pair can be interpreted as saying that a pair (b, c) is blocking with respect to a matching N if neither agent of the pair prefers their current partner in N at least as much as the partner in the pair.

We use SMG to refer to the decision problem of whether a given instance admits a stable matching (which we also call a solution). The following definition introduces a special class of instances.

Definition 4.0.7. *An instance $\mathcal{I}$ of* SMG *is said to have asymmetric preferences if for every $b_1, b_2 \in B$ and $c \in C$ at most one of the following two conditions holds: $(b_1, b_2) \in \mathcal{R}_c$ or $(b_2, b_1) \in \mathcal{R}_c$.*

Note that we could have obtained an alternate definition of stability by saying that a pair (b, c) is blocking if $(b, c) \notin N$, $c \succ_b N(b)$ and $(b, N(c)) \in \mathcal{R}_c$. Note that then we would also have to chance the definition of asymmetric preferences accordingly. However, the two models are equivalent via the following correspondence: create a new instance $\mathcal{I}'$ with sets $\mathcal{R}'_c$ where $(b, b') \in \mathcal{R}'_c$ if and only if $(b', b) \notin \mathcal{R}_c$. Then the solutions that are stable for $\mathcal{I}$ under our definition of stability, are exactly those that are stable for $\mathcal{I}'$ using the alternate definition of stability. Hence, we can use our definition of stability without loss of generality.

It is easy to see that any stable matching for an SMG instance must be a perfect matching. This contrasts the case of SMTI where a given instance can have stable matchings of different size. A second difference is that unlike in SMTI, not all instances of SMG admit a stable matching as demonstrated by the example consider in chapter 1 Figure 1.6. We now introduce our second model of stable matching: the stable extension problem.

Definition 4.0.8. *An instance $\mathcal{I}$ of* SE *consists of a set of n dogs A, a set of n men B, and a set of n women C, together with a fixed perfect matching M on $A \times B$. The preferences are defined cyclically (A over B, B over C, and C over A) and are complete total orders over the corresponding sets. A triple (a, b, c) is blocking with respect to a 3D matching $\mathcal{M}$ if $(a, b, c) \notin \mathcal{M}$, $b \succ_a \mathcal{M}(a)$, $c \succ_b \mathcal{M}(b)$ and $a \succ_c \mathcal{M}(c)$. A 3D matching $\mathcal{M}$ is stable if it has no blocking pairs.*

We note that if (a, b, c) is a blocking triple then a, b and c must be part of three disjoint triples in $\mathcal{M}$. It follows from this definition that any stable 3D matching must be a perfect matching. We say that a perfect matching N on $B \times C$ is a *stable extension*, or a solution to $\mathcal{I}$, if $M \circ N$ is a 3D stable matching, and we use SE to refer to the decision problem of whether a given instance admits a stable extension.

Our first result demonstrates demonstrate how an instance $\mathcal{I}$ of SE can be reduced to an SMG instance. Let $\mathcal{I}$ be an instances of SE. We describe how to obtain a corresponding instance of SMG. First, for each man $b \in B$ we define A_b to be the set of dogs in A that prefer b to the man assigned to them in the fixed perfect matching M. That is

$$A_b = \{a \in A : b \succ_a M(a)\} \tag{4.1}$$

The set A_b contains exactly those dogs in A with whom man b can potentially be in a blocking triple when extending M to a 3D matching. It follows that if $A_b = \emptyset$ then man b cannot be in a blocking triple in any extension of M to a 3D matching. Now, for each pair (b, c) where $A_b \neq \emptyset$ we define $\alpha(b, c)$ to be the dog in the set A_b that woman c prefers the most. That is

$$\alpha(b, c) = \{a \in A_b : a \succ_c a' \text{ for all } a' \in A_b,\, a' \neq a\} \tag{4.2}$$

Since the preference list of c does not contain any ties, $\alpha(b, c)$ is well-defined. If $A_b = \emptyset$ then we let $\alpha(b, c)$ be the dog in the last position in woman c's preference list. We now define preferences for each woman $c \in C$

$$\mathcal{R}_c := \{(b, b') \,|\, b, b' \in B, b \neq b',\, M(b) \succeq_c \alpha(b', c)\}\,. \tag{4.3}$$

Note that if $(N(c), b) \in \mathcal{R}_c$ then in the 3D matching $M \circ N$ the woman c will be matched to a dog that she prefers at least as much as any dog in A_b, therefore guaranteeing that the man b and woman c will never be part of the same blocking triple. Hence, in order for $M \circ N$ to be a 3D stable matching it suffices to ensure that for all $(b, c) \notin N$ we either have b matched to someone better than c, meaning $N(b) \succ_b c$, or we have c matched to some $N(c)$ such that $(N(c), b) \in \mathcal{R}_c$. But this is exactly the definition of a stable matching for an instance of SMG. Hence we have the following theorem.

Theorem 4.0.9. SE *can be reduced in polynomial time to* SMG*.*

4.1 Hardness results for SMG

We note that Theorem 4.0.9, together with Theorem 4.3.1 imply NP-hardness of SMG, and hence prove Theorem 4.1.1 (modulo containment in NP which is straightforward). Nevertheless, we choose to present first the proof of Theorem 4.1.1 as a warm-up, as it shares many similarities with the more complex version of Theorem 4.3.1.

Theorem 4.1.1. SMG *is NP-complete.*

Proof. Containment in NP is straightforward, and is based on the observation that deciding whether an edge not in a perfect matching of an instance of SMG is blocking or not can be done in polynomial time in n. The rest of our argument focuses on hardness. Our proof uses a polynomial time reduction from SMTI where ties occur only in the preference lists of the women. This problem is known to be NP-complete [62]. Let $\mathcal{I}$ be an instance of SMTI where ties occur only on the side of the women. We let $B = \{b_1, \cdots, b_n\}$ denote the set of men and $C = \{c_1, \cdots, c_n\}$ denote the set of women for the instance $\mathcal{I}$. For each person $q \in B \cup C$ we let $P(q)$ denote their preference list.

We now describe how to construct an instance $\mathcal{J}$ of SMG. The set of men for our instance will be given by $B' = B \cup \{b_{n+1}\}$ and the set of women by $C' = C \cup \{c_{n+1}\}$. The preferences of the men are defined as follows: each original man $b \in B$ ranks the women in $P(b)$ first, in the same order as in $P(b)$, followed by the woman c_{n+1}, and the remaining women of C' ranked arbitrarily; the man b_{n+1} ranks the woman c_{n+1} first, and the remaining women of C' arbitrarily. Now, for each original woman $c \in C$ we define the binary relation $\mathcal{R}_c \subseteq B \times B$ as follows $\mathcal{R}_c := \{(b, b') \,|\, b, b' \in P(c), b \succeq_c b'\}$. That is, $\mathcal{R}_c$ contains the ordered pair (b, b') when both b and b' are acceptable to c under the instance $\mathcal{I}$ and c prefers b at least as much as b'. Finally, for the woman c_{n+1} we set $\mathcal{R}_{c_{n+1}} := \emptyset$. This completes the definition of the instance $\mathcal{J}$.

We use the introduction of the new agents to establish the following property

Lemma 4.1.2. *In any solution N to $\mathcal{J}$ every men $b \in B$ is matched to a woman from the set $P(b)$.*

Proof. Let N be a solution to $\mathcal{J}$. We first show that b_{n+1} is matched to c_{n+1} in N. Suppose by contradiction that $N(b_{n+1}) \neq c_{n+1}$. Then from the way we defined the preferences of b_{n+1} in $\mathcal{J}$ we can conclude that b_{n+1} prefers c_{n+1} to its partner in N. Hence, in order for the pair (b_{n+1}, c_{n+1}) to not be blocking with respect to the solution N we must have $(N(c_{n+1}), b_{n+1}) \in \mathcal{R}_{c_{n+1}}$, which gives us a contradiction since we defined $\mathcal{R}_{c_{n+1}} = \emptyset$. Hence $N(b_{n+1}) = c_{n+1}$.

We can now prove the lemma. Suppose by contradiction that there exists a man $b \in B$ such that $N(b) \notin P(b)$. Then b cannot be matched to c_{n+1}, since we showed that c_{n+1}

is always matched to b_{n+1}. From the way we defined the preferences of b in $\mathcal{J}$ we can conclude that b prefers c_{n+1} to its partner in N. Hence, in order for the pair (b, c_{n+1}) to not be blocking with respect to the solution N we must have $(N(c_{n+1}), b) \in \mathcal{R}_{c_{n+1}}$, which gives us a contradiction since we defined $\mathcal{R}_{c_{n+1}} = \emptyset$. □

The following lemma completes our proof.

Lemma 4.1.3. *$\mathcal{I}$ admits a perfect stable matching if and only if $\mathcal{J}$ admits a stable matching.*

Proof. Suppose that N is a perfect stable matching for $\mathcal{I}$. Then complete N to a perfect matching on $B' \cup C'$ by matching b_{n+1} to c_{n+1}. To see that this is a stable matching for $\mathcal{J}$, note that b_{n+1} is matched to its most preferred woman in C', hence it cannot be part of any blocking pairs. It remains to show that no man in B can be part of a blocking pair. Consider a man $b \in B$, and suppose that c is a woman that b strictly prefers to $N(b)$ according to the preferences in $\mathcal{J}$. Then it must be the case that $c \in P(b)$ and b also strictly prefers c to $N(b)$ in $\mathcal{I}$. Since N is a solution to $\mathcal{I}$, woman c prefers $N(c)$ at least as much as b in $\mathcal{I}$. Hence from the way we defined the set $\mathcal{R}_c$ we have $(N(c), b) \in \mathcal{R}_c$, implying that (b, c) is not blocking in $\mathcal{J}$ either.

To see the other direction suppose that $\mathcal{J}$ admits a stable matching, and let N be the part of this stable matching obtained by restricting it to the sets $B \cup C$. It follows from lemma 4.1.2 that N is a perfect matching and every man is matched to an acceptable woman. To see that there are no blocking pairs, consider any pair $(b, c) \notin N$ such that (b, c) is an acceptable pair, that is $b \in P(c)$ and $c \in P(b)$. Assume now that in $\mathcal{I}$ man b strictly prefers c to $N(b)$. Since $c \in P(b)$ it follows that b also strictly prefers c to $N(b)$ in $\mathcal{J}$. Hence, since N is a stable matching for $\mathcal{J}$, we must have $(N(c), b) \in \mathcal{R}_c$. From the way we defined $\mathcal{R}_c$ this implies that $N(c)$ is acceptable to c, and c prefers $N(c)$ at least as much as b in $\mathcal{I}$. Therefore N does not have any blocking pairs, and is a stable matching in $\mathcal{I}$. □

□

4.2 Algorithmic results for SMG

In this section we introduce a variant of the Gale-Shapley man-proposing algorithm for instances of SMG that have asymmetric preferences. Let $\mathcal{I}$ be an instance of SMG. Like in the classical algorithm, each man in B is originally declared single and is given a list containing all the women of C in order of preference. In each round, every man b that is still single proposes to its most preferred woman in C that is still in his list. If a woman c accepts a proposal from a man b then they become engaged, and b's status changes from single to engaged. On the other hand, if c rejects b's proposal then b removes woman c from his list and remains single. The difference from the original Gale-Shapley algorithm from section 2.5.1 is in the way that the women decide to accept or reject incoming proposals. A woman c accepts a proposal from a man b if and only if $(b, b') \in \mathcal{R}_c$ for all other men b' that have proposed to c up to that point in the algorithm. This will ensure that whenever a woman c rejects a proposal from a man b, c is guaranteed to be matched at the end of the algorithm to some b' such that $(b', b) \in \mathcal{R}_c$ therefore ensuring that (b, c) will not be a blocking pair. The description of the algorithm is given below.

Algorithm 4 A deferred acceptance algorithm for SMG

```
while there a single man in B with a non-empty list do
    for all b single with a non-empty list do
        b proposes to the top c in its list
    for all b, c such that b proposed to c do
        if (b, b') ∈ R_c for all b' ≠ b that proposed to c then
            c accepts b ( (b, c) become an engaged pair)
        else
            c rejects b
If the set of engaged pairs forms a perfect matching return this solution, else conclude
that I does not admit a stable matching.
```

It is easy to see that algorithm 4 terminates and runs in polynomial time since each man in B proposes to every woman in C at most once. The following lemma is also easy to establish.

Lemma 4.2.1. *Any solution returned by algorithm 4 is a stable matching.*

Proof. Let N be a solution returned by algorithm 4. It then follows that N is a perfect matching. Suppose there is a blocking pair (b, c). Then b must have proposed to c, and c rejected b's proposal. But, every time a woman c rejects a proposal from b, algorithm 4 ensures that c will only become engaged to some man b' such that $(b', b) \in \mathcal{R}_c$. Therefore we cannot have any blocking pairs. □

We now present our main result of this section.

Theorem 4.2.2. *For instances of* SMG *with asymmetric preferences, there exists a polynomial time algorithm that finds a solution if and only if one exists.*

Proof. Using lemma 4.2.1 it suffices to show that if algorithm 4 does not find a solution then the given instance does not admit a stable matching. Suppose by contradiction that there exists a stable matching N but algorithm 4 does not find a solution. First note that since the preferences are asymmetric, no woman $c \in C$ accepts more then one proposal at any point in the algorithm. Hence every woman is engaged to at most one man. Now since Algorithm 1 does not find a solution there is a man b that is rejected by every woman in C. In particular, b is rejected by $N(b)$. Among all pairs $(b, c) \in N$ such that b proposed to c and c rejected b, let (b_0, c_0) be the one that corresponds to the earliest rejection. As observed earlier, no man is rejected because of an arbitrary choice. Therefore, if b_0 was rejected at c_0 then there must be some man b_1 that also proposed to c_0 and $(b_0, b_1) \notin \mathcal{R}_{c_0}$. Let c_1 be the partner of b_1 in N. We must have $c_1 \succ_{b_1} c_0$, since otherwise (b_1, c_0) would be a blocking pair for the stable matching N. Thus b_1 proposed to c_1 before proposing to c_0 and c_1 rejected b_1 before c_0 rejected b_0. But this contradicts our choice of (b_0, c_0). This concludes the proof of the theorem. □

We now provide a polyhedral description for SMG, that is an analogue of the well studied stable marriage polytope, first introduced in [79]. It is well known that the latter polytope is integral, meaning that the optimization version of SM can be solved in polynomial time. For our setting, we show that our polytope can be used to efficiently decide the feasibility of an SMG instance with asymmetric preferences, thus giving an alternative proof of Theorem 4.2.2. We remark however that our polytope is not integral for this class of instances. Indeed, one can easily find instances with asymmetric preferences for which our polytope has fractional extreme points.

Given an instance $\mathcal{I}$ of SMG we associated with each pair $(b,c) \in B \times C$ a variable x_{bc}, with the intended meaning that $x_{bc} = 1$ if b and c are matched to each other and $x_{bc} = 0$ otherwise. We then consider the following relaxation of the problem and let $P(\mathcal{I})$ denote the set of all vectors satisfying the constraints below

$$\sum_c x_{bc} = 1 \quad \text{for all } b \in B \tag{4.4}$$

$$\sum_b x_{bc} = 1 \quad \text{for all } c \in C \tag{4.5}$$

$$x_{bc} + \sum_{c' \succ_b c} x_{bc'} + \sum_{(b',b) \in \mathcal{R}_c} x_{b'c} \geq 1 \quad \text{for all } b \in B, c \in C \tag{4.6}$$

$$x_{bc} \geq 0 \quad \text{for all } b \in B, c \in C \tag{4.7}$$

It is easy to check that x is the incidence vector of a stable matching for $\mathcal{I}$ if and only if x is an integer vector in $P(\mathcal{I})$. Our main result is the following.

Theorem 4.2.3. *Let $\mathcal{I}$ be an instance of* SMG *with asymmetric preferences. Then $P(\mathcal{I}) \neq \emptyset$ if and only if $\mathcal{I}$ admits a stable matching. Furthermore any fractional point $x \in P(\mathcal{I})$ can be efficiently rounded to a stable matching solution for $\mathcal{I}$.*

Proof. The first direction is trivial since if $\mathcal{I}$ admits a stable matching then the incidence vector corresponding to this stable matching is clearly in P. Now assume $P \neq \emptyset$ and let x be any point in P. We will show how to efficiently round x to a stable matching, thus completing the proof of the theorem. For each $b \in B$ let $f(b)$ be b's most preferred woman in the set $\{c \in C : x_{bc} > 0\}$. Define $N = \{(b, f(b)) : b \in B\}$. We first show that N is a perfect matching. Since each man selects exactly one woman it suffices to show that no two men select the same woman. Suppose by contradiction that $f(b_1) = f(b_2) = c$ for some $b_1 \neq b_2$. Note that

$$\begin{aligned} f(b) = c \quad &\Rightarrow \sum_{c' \succ_b c} x_{bc'} = 0 \quad \text{from the definition of } f(b) \\ &\Rightarrow x_{bc} + \sum_{(b',b) \in \mathcal{R}_c} x_{b'c} \geq 1 \quad \text{from the stability constraint for } (b,c) \\ &\Rightarrow x_{bc} + \sum_{(b',b) \in \mathcal{R}_c} x_{b'c} = 1 \quad \text{from the matching constraint for } c \\ &\Rightarrow \sum_{(b',b) \notin \mathcal{R}_c} x_{b'c} = 0. \end{aligned}$$

Therefore $f(b_1) = f(b_2) = c$ implies that $(b_1, b_2) \in \mathcal{R}_c$ and $(b_2, b_1) \in \mathcal{R}_c$. But this contradicts the assumption that the preferences are asymmetric. Hence N must be a perfect matching. To see that N satisfies the stability constraints consider any pair (b, c). If $f(b) = c$ or $f(b) \succ_b c$ then (b, c) cannot be blocking, since b will be matched in N to someone he prefers at least as much as c. Hence it suffices to consider the case where $c \succ_b f(b)$. But then we must have $x_{bc} + \sum_{c' \succ_b c} x_{bc'} = 0$ and since $x \in P(\mathcal{I})$ this implies that $\sum_{(b',b) \in \mathcal{R}_c} x_{b'c} = 1$. Now, since N uses only edges in the support of x, it follows that $(N(c), b) \in \mathcal{R}_c$ and hence (b, c) is not blocking. Therefore N is a stable matching for $\mathcal{I}$. □

4.3 Hardness results for SE

In this section we present a hardness result for SE similar to the one we proved for SMG. At a high level, our proof adopts a similar strategy to the one used for SMG. In particular, we will provide a polynomial-time reduction for SMTI to SE. However, due to the additional structural properties of SE instances this reduction becomes significantly more intricate.

Theorem 4.3.1. SE *is NP-complete.*

Proof. Containment of SE in NP follows by observing that deciding whether any triple not in a 3D matching of any instance (extending the given 2D matching) is blocking or not, can be done in polynomial time in n. The rest of the argument focuses on hardness. Let $\mathcal{I}$ be an instance of SMTI where ties occur only on the side of the women. We let

$B = \{b_1, \cdots, b_n\}$ denote the set of men and $C = \{c_1, \cdots, c_n\}$ denote the set of women for the instance $\mathcal{I}$. For each person $q \in B \cup C$ let $P(q)$ denote their preference list. For a given woman $c_i \in C$ the preference list $P(c_i)$ might have several men tied at a certain position. We let t_i be the number of positions in c_i's preference list, so that $1 \leq t_i \leq n$. Now for each $j \in \{1, \cdots, t_i\}$ we let $P(c_i)_j$ denote the men that are tied in position j in c_i's preference list. Furthermore, for each $b \in P(c_i)$ we let $r_{c_i}(b)$ denote the position that b occupies in $P(c_i)$.

We now describe how to construct an instance $\mathcal{J}$ of SE. Our instance will consist of three sets A', B' and C'. We will include all the elements of B in B', and all the elements of C in C'. We will also define some extra elements as follows

- For all $i \in \{1, 2, 3\}$ we create the new elements a_{n+i}, b_{n+i} and c_{n+i}.
- For all $i \in [n]$ we create t_i new elements $B_i = \{b_{i,1}, \cdots, b_{i,t_i}\}$, together with the corresponding sets $A_i = \{a_{i,1}, \cdots, a_{i,t_i}\}$, and $C_i = \{c_{i,1}, \cdots, c_{i,t_i}\}$.
- We create a set of n new elements $A = \{a_1, \cdots, a_n\}$, corresponding to the original sets B and C.

We then set

$$A' = A \cup \bigcup_{i=1}^{n} A_i \cup \{a_{n+1}, a_{n+2}, a_{n+3}\}$$
$$B' = B \cup \bigcup_{i=1}^{n} B_i \cup \{b_{n+1}, b_{n+2}, b_{n+3}\}$$
$$C' = C \cup \bigcup_{i=1}^{n} C_i \cup \{c_{n+1}, c_{n+2}, c_{n+3}\}$$

We now fix the perfect matching M on $A' \times B'$ by matching each dog of A' to its corresponding man in B'. That is a_i is matched to b_i for all $i \in [n]$, $a_{i,j}$ is matched to $b_{i,j}$ for all $i \in [n]$ and $j \in [t_i]$, and a_{n+i} is matched to b_{n+i} for all $i \in \{1, 2, 3\}$.

Next, we create the preferences lists of each agent. For each $q \in A' \cup B' \cup C'$ we let $P'(q)$ denote the preference list of agent q for the instance $\mathcal{J}$. Recall that for each $a \in A'$ the preference list $P'(a)$ must be a complete and strict ordering of the set B'. Similarly for each

for all $i \in [n]$	$P'(a_i) = b_i, [\cdots]$ $P(b_i), c_{n+1}, [\cdots]$ $P'(c_i) = [M(P(c_i)_1)], a_{i,1}, \cdots, [M(P(c_i)_{t_i})], a_{i,t_i}, [\cdots]$
for all $i \in [n]$, for all $j \in [t_i]$	$P'(a_{i,j}) = [P(c_i)_j], b_{i,j}, [\cdots]$ $P'(b_{i,j}) = c_{i,j}, [\cdots]$ $P'(c_{i,j}) = a_{n+2}, [\cdots]$
	$P'(a_{n+1}) = b_{n+1}, [\cdots]$ $P'(a_{n+2}) = [\cdots], b_{n+2}$ $P'(a_{n+3}) = b_{n+2}, b_{n+3}, [\cdots]$
for all $i \in [3]$	$P'(b_{n+i}) = c_{n+i}, [\cdots]$
	$P'(c_{n+1}) = a_{n+2}, [\cdots]$ $P'(c_{n+2}) = a_{n+3}, [\cdots]$ $P'(c_{n+1}) = a_{n+2}, [\cdots]$

Table 4.1: The preferences for the instance $\mathcal{J}$.

$b \in B'$ the list $P'(b)$ will be a complete and strict ordering of the set C', and for each $c \in C'$ the list $P'(c)$ will be a complete and strict ordering of the set A'. Table 4.1 summarizes the preferences of each agent. When listing the preference list $P'(q)$ of an agent q, we use the notation $[Q]$ to denote that the agents of the set Q appear in consecutive positions in $P'(q)$ in any arbitrary order among them. The notation $[\cdots]$ is used to denote that the remaining agents that have not been listed in $P'(q)$ appear in consecutive positions in $P'(q)$ in any arbitrary order among them.

The preferences of the dogs in A' over the set of men B' are defined as follows

- For all $i \in [n]$: a_i ranks b_i first, and the remaining men of B' arbitrarily.
- For all $i \in [n]$ and for all $j \in [t_i]$: $a_{i,j}$ ranks the men that are tied in position j of women c_i's list (in the instance $\mathcal{I}$) at the top of its list in any arbitrary order among them, followed by the man $b_{i,j}$. The remaining men of B' are ranked arbitrarily.
- a_{n+1} ranks b_{n+1} first, a_{n+2} ranks b_{n+2} last, and a_{n+3} ranks b_{n+2} first followed by b_{n+3} second. The rest of their lists are arbitrary.

The preferences of the men in B' over the set of women C' are defined as follows

– For all $i \in [n]$: b_i ranks the women in $P(b_i)$ first, in the same order as in $P(b_i)$, followed by the woman c_{n+1}, and the remaining women of C' ranked arbitrarily.

– For all $i \in [n]$ and for all $j \in [t_i]$: $b_{i,j}$ ranks $c_{i,j}$ first, and the remaining women of C' arbitrarily.

– For all $i \in \{1, 2, 3\}$: b_{n+i} ranks c_{n+i} first and the remaining women of C' arbitrarily.

The preferences of the women in C' over the set of dogs A' are defined as follows

– For all $i \in [n]$: c_i ranks the dogs in the set $M(P(c_i)_1)$ at the top of its list in an arbitrary order, followed by the dog $a_{i,1}$, then the dogs in the set $M(P(c_i)_2)$ again in an arbitrary order, followed by the dog $a_{i,2}$, and so on until the dogs in the set $M(P(c_i)_{t_i})$ followed by the a_{i,t_i}. The rest of c_i's preference list is arbitrary.

– For all $i \in [n]$ and for all $j \in [t_i]$: $c_{i,j}$ ranks a_{n+2} first, and the remaining dogs in A' arbitrarily.

– c_{n+1} and c_{n+3} rank a_{n+2} first, and the remaining dogs in A' arbitrarily, while c_{n+2} ranks b_{n+3} first and the remaining dogs in A' arbitrarily.

This concludes the description of the instance $\mathcal{J}$. Note that since we already fixed the perfect matching M, the sets $\{A_b : b \in B'\}$ are now fully determined as follows

$$A_b = \begin{cases} \left\{a_{i,r_{c_i}(b)} : i \in [n], b \in P(c_i)\right\} \cup \{a_{n+2}\} & \text{if } b \in B, \\ \{a_{n+2}\} & \text{if } b \in B' \setminus (B \cup \{b_{n+2}\}), \\ \{a_{n+3}\} & \text{if } b = b_{n+2}. \end{cases} \tag{4.8}$$

This also fixes the sets $\mathcal{R}_c$ for all $c \in C$ using the definition given in equation (4.3). In particular, the sets $\mathcal{R}_c$ satisfy the following property.

Lemma 4.3.2. *For every $c \in C$ and $b, b' \in B$ where b is an acceptable partner of c in $\mathcal{I}$, we have $(b', b) \in \mathcal{R}_c$ if and only if b' is also an acceptable partner of c in $\mathcal{I}$ and c prefers b' at least as much as b.*

Proof. Let $(b, c) \in B \times C$ such that (b, c) is an acceptable pair for the instance $\mathcal{I}$. Then b must appear in the preference list $P(c)$ of woman c. Recall that $r_c(b)$ denotes the position that b occupies in $P(c)$. Then it follows from the description of A_b in (4.8) that the only dog from the set $\{a_{i,1}, \cdots, a_{i,t_i}\}$ that is contained in A_b is $a_{i,r_c(b)}$. It then follows from the way we defined the preference list $P'(c_i)$ that $\alpha(b, c_i) = a_{i,r_c(b)}$. The lemma then follows form the definition of the sets $\mathcal{R}_c$. □

The proof of the Theorem is completed via the following lemmas.

Lemma 4.3.3. *In any solution to $\mathcal{J}$ b_{n+2} and b_{n+3} are matched to c_{n+2} and c_{n+3}.*

Proof. Let N be a solution to $\mathcal{J}$. Suppose by contradiction that this is not true. Then at least one of the men b_{n+2} and b_{n+3} must be matched to a woman that is neither c_{n+2} nor c_{n+3}. Assume that this is b_{n+2}. The case with b_{n+3} follows from a symmetric argument. Then since c_{n+2} was the most preferred woman of b_{n+2} it follows that c_{n+2} must be matched to some b' such that $(b', b_{n+2}) \in \mathcal{R}_{c_{n+2}}$, since otherwise the pair (b_{n+2}, c_{n+2}) would be blocking. Now from (4.8) we have that $A_{b_{n+2}} = \{a_{n+3}\}$ hence $\alpha(b_{n+2}, c_{n+2}) = a_{n+3}$ and therefore it follows from the preference list $P'(c_{n+2})$ that $(b', b_{n+2}) \in \mathcal{R}_{c_{n+2}}$ if and only if $b = b_{n+3}$. Hence c_{n+2} must be matched to b_{n+3}. But now we can argue that since c_{n+3} was the most preferred woman of b_{n+3}, c_{n+3} must be matched to some b' such that $(b', b_{n+3}) \in \mathcal{R}_{c_{n+3}}$. Again from (4.8) we have that $A_{b_{n+3}} = \{a_{n+3}\}$ hence $\alpha(b_{n+3}, c_{n+3}) = a_{n+2}$ and therefore it follows from the preference list $P'(c_{n+3})$ that $(b', b_{n+3}) \in \mathcal{R}_{c_{n+3}}$ if and only if $b' = b_{n+2}$. But this means that c_{n+3} must be matched to b_{n+2}, which contradicts our assumption that b_{n+2} is not matched to neither c_{n+2} nor c_{n+3}. □

Lemma 4.3.4. *In any solution to $\mathcal{J}$ b_{n+1} is matched to c_{n+1}.*

Proof. Let N be a solution to $\mathcal{J}$. Suppose by contradiction that b_{n+1} is not matched to c_{n+1}. Then since c_{n+1} was the most preferred woman of b_{n+1} it follows that c_{n+1} must be matched to some b' such that $(b', b_{n+1}) \in \mathcal{R}_{c_{n+1}}$. Now from (4.8) we have that $A_{b_{n+1}} = \{a_{n+2}\}$ which implies that $\alpha(b_{n+1}, c_{n+1}) = a_{n+2}$. Therefore it follows from the preference list $P'(c_{n+1})$ that $(b', b_{n+1}) \in \mathcal{R}_{c_{n+1}}$ if and only if $b' = b_{n+2}$. But from lemma 4.3.3 we know that b_{n+2} is always matched to either c_{n+2} or c_{n+3}. Therefore by contradiction, b_{n+1} must be matched to c_{n+1}. □

Lemma 4.3.5. *In any solution to $\mathcal{J}$ every man $b \in B$ is matched to a woman from the set $P(b)$.*

Proof. Consider a man $b \in B$ and suppose by contradiction that b is not matched to someone in $P(b)$. It follows from lemma 4.3.4 that b cannot be matched to c_{n+1}, since c_{n+1} is matched to b_{n+1}. From the way we defined the preferences of b in $\mathcal{J}$ we can conclude that b prefers c_{n+1} to its match in N. Furthermore we have $a_{n+2} \in A_b$ from (4.8). Now since c_{n+1} ranks a_{n+2} at the top of its list it follows that $\alpha(b, c_{n+1}) = a_{n+2}$ and thus $(b', b) \in \mathcal{R}_{c_{n+1}}$ if and only if $b = b_{n+2}$. But then the pair (b, c_{n+1}) is blocking, thus contradicting the fact that N is a solution to $\mathcal{J}$. □

Lemma 4.3.6. *$\mathcal{I}$ admits a complete weakly stable matching if and only if $\mathcal{J}$ admits a stable extension.*

Proof. Suppose that N is a complete stable matching for $\mathcal{I}$. Then complete N to a perfect matching on $B' \cup C'$ by matching $b_{i,j}$ to $c_{i,j}$ for every $i \in [n]$ and $j \in [t_i]$ and matching b_{n+i} to c_{n+i} for every $i \in \{1, 2, 3\}$. Note that every man in $B' \backslash B$ is matched to the woman that it prefers the most hence no man in B' can be part of a block. From lemma 4.3.5 we have that every man in B is matched to a partner that is acceptable in $\mathcal{I}$. Hence for any $b \in B$, if c is a woman that b strictly prefers to $N(b)$ according to the preferences in $\mathcal{J}$, then it must be the case that $c \in P(b)$ and b also strictly prefers c to $N(b)$ in $\mathcal{I}$. Since N is a solution to $\mathcal{I}$ it follows that c prefers $N(c)$ at least as much as b. We can now use lemma 4.3.2 to conclude that $(N(c), b) \in \mathcal{R}_c$. Therefore (b, c) cannot be a blocking pair in $\mathcal{J}$. Hence the perfect matching that we defined from N is solution to the instance $\mathcal{J}$.

Conversely, suppose that $\mathcal{J}$ admits a stable extension, and let N be the part of this stable extension obtained by restricting it to the sets $B \cup C$. It follows from lemma 4.3.5 that N is a perfect matching and every agent is matched to an acceptable partner. To see that there are no blocking pairs consider any pair $(b, c) \notin N$ such that (b, c) is an acceptable pair, that is $b \in P(c)$ and $c \in P(b)$. Assume now that in $\mathcal{I}$ man b strictly prefers c to $N(b)$. Since $c \in P(b)$ it follows that b also strictly prefers c to $N(b)$ in $\mathcal{J}$. Since N is a solution to $\mathcal{J}$ we must have $(N(c), b) \in \mathcal{R}_c$. Again using lemma 4.3.2 we can conclude that $N(c)$ is a man in $P(c)$ that c prefers at least as much as b in $\mathcal{I}$. Therefore N does not have any blocking pairs, and is a weakly stable matching in $\mathcal{I}$. $\square$

This completes the proof of the theorem. $\square$

Chapter 5

The nucleolus of matching games with empty core

Let (G, w) be a weighted graph and (N, v) the corresponding matching game. We assume throughout this chapter that the edge weights are always non-negative. We denote by $\nu(G, w)$ the maximum weight of matching of G with respect to the weights w. We recall the primal dual linear programs defined in section 2.1.1

$$
\begin{aligned}
(P_{FM}) \quad \max \quad & \sum_{e \in E} w(e) x_e \\
x(\delta(u)) &\leq 1 && \text{for all } u \in V(G) \\
x_e &\geq 0 && \text{for all } e \in E(G).
\end{aligned}
\tag{5.1}
$$

$$
\begin{aligned}
(D_{FM}) \quad \min \quad & \sum_{u \in V(G)} y_u \\
y_u + y_v &\geq w(e) && \text{for all } e = (u, v) \in E(G) \\
y_u &\geq 0 && \text{for all } u \in V(G).
\end{aligned}
\tag{5.2}
$$

We let $\nu_f(G, w)$ denote the optimal value of P_{FM} and D_{FM}. We are interested in the case where core$(N, v) = \emptyset$, hence we assume throughout this chapter that $\nu_f(G, w) > \nu(G, w)$ (see Theorem 2.3.3).

Our approach for computing the nucleolus of (N, v) will be similar to the one used by [53] and [68]. That is, we will focus on first obtaining a polynomial sized characterization of the least core. We start by recalling the definition of the least core.

Definition 5.0.7. *Given a weighted graph (G, w) we define the linear program $P = P(G, w)$:*

$$
\begin{aligned}
(P(G,w)) \quad & \max \epsilon \\
& s.t.\ x(M) \geq w(M) + \epsilon \quad \text{for all } M \in \mathcal{M}(G) \\
& \qquad x(V(G)) = \nu(G, w) \\
& \qquad x \geq 0
\end{aligned}
$$

where $\mathcal{M}(G)$ denotes the set of matchings of G. We denote by $\epsilon_1 = \epsilon_1(G, w)$ the optimal value of P and for any ϵ we let $P(\epsilon)$ be the set of vectors x such that (x, ϵ) is feasible to P. Then the least core of the matching game induced by (G, w) is defined as

$$\textit{leastcore}(G, w) = P(\epsilon_1).$$

For node-weighted graphs Paulusma [68] obtained an equivalent characterization of the least core that uses only a polynomial number of constraints. This characterization is based on the odd components of the Edmonds-Gallai decomposition of the graph. The following lemma is shown in [68].

Lemma 5.0.8. *Let (G, w) be a weighted graph and $\mathcal{D}(G)$ denote the set of odd components in the Edmonds-Gallai decomposition of G. If $\nu(G, w) < \nu_f(G, w)$ then there exits $D \in \mathcal{D}(G)$ with $|D| > 1$.*

Hence whenever the core of the matching game corresponding to (G, w) is empty, there must exist at least one non-singleton odd component. The following theorem is the main result in [68].

Theorem 5.0.9. *Let (G, y) be a node-weighted graph with respect to the non-negative node weights $y \in \mathbb{R}_+^{|V(G)|}$. Suppose that $\nu(G, y) < \nu_f(G, y)$. Then leastcore(G, y) is the set of* *vectors x that are optimal solutions to the linear program*

$$\begin{aligned}
max \quad & \epsilon \\
s.t. \quad & x(e) \geq y(e) && \text{for all } e \in E(G) \backslash \left(\bigcup_{D \in \mathcal{D}(G)} E(D) \right) \\
& x_i \leq y_i && \text{for all } i \in D, D \in \mathcal{D}(G) \\
& x_i - y_i = x_j - y_j && \text{for all } i, j \in D,\ D \in \mathcal{D}(G) \\
& x(M_1) \geq y(M_1) + \epsilon \\
& x(V(G)) = \nu(G, y) \\
& x \geq 0
\end{aligned}$$

where $\mathcal{D}(G)$ denotes the set of odd components in the Edmonds-Gallai decomposition of G and M_1 is an arbitrary maximum cardinality matching contained in the union of these odd components.

Theorem 5.0.9 essentially says that if G is node-weighted then the following are true

(i) In any least core allocation the edges belonging to the odd components of the Edmonds-Gallai decomposition have non-positive excess while all other edges of the graph have non-negative excess.

(ii) In any least core allocation the edges belonging to the same odd component of the Edmonds-Gallai decomposition must have equal excess.

This implies that any maximum cardinality matching on the union of the odd components has excess ϵ_1. Hence it suffices to include only one constraint of the type $x(M) \geq w(M) + \epsilon$ for the fixed matching M_1. Then any other matching of G will have excess no worse then M_1.

Since this characterization no longer holds true for graphs that are not node-weighted, a different approach is needed for general weights. Our main idea is to find a new set of components that satisfy the two properties of Theorem 5.0.9. That is, the edges inside

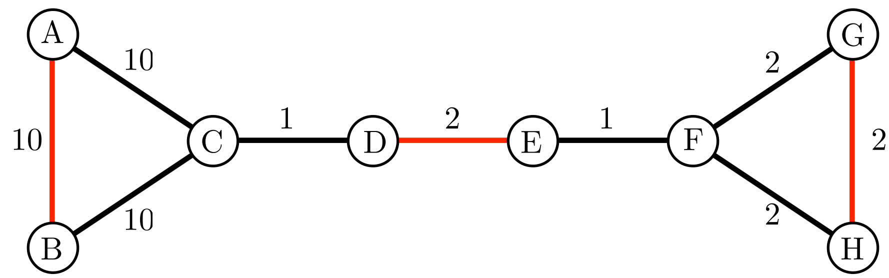

Figure 5.1: A matching game that is not node-weighted.

these components receive non-positive excess in any least core allocation, the edges outside these components receive non-negative excess, and the excess is equal along each of these components.

As an example consider the matching game shown in Figure 5.1. The red edges denote a maximum weight matching of weight 14. This core is empty since the fractional matching whose support consists of the two triangles and the middle edge has weight $20 > 14$. Also, this graph is not node-weighted, since there is no possible assignment of non-negative values to each vertex such that the weight of each edge is equal to the sum of the values of its endpoints.

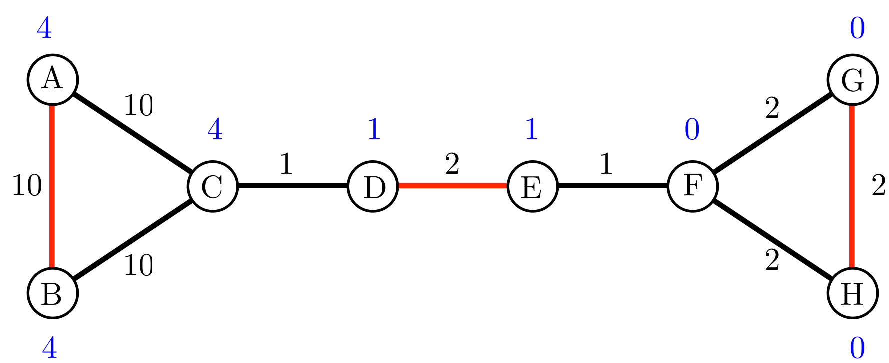

Figure 5.2: A least core allocation with negative excess on both triangles.

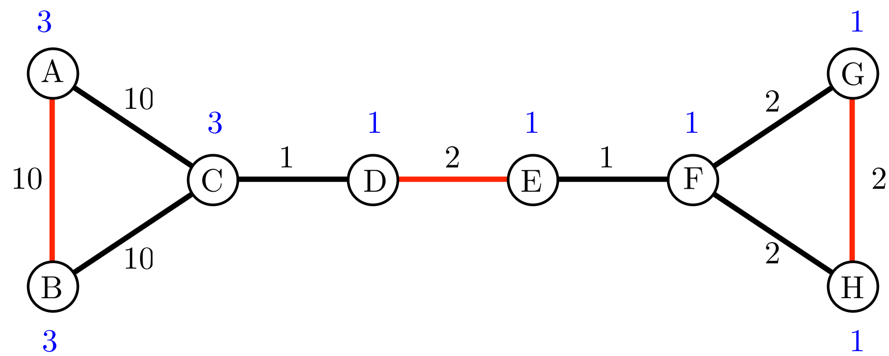

Figure 5.3: A least core allocation with negative excess only on the left triangle.

Note however that the graph in this example does have a perfect matching, meaning that there are no odd components in the Edmonds-Gallai decomposition of the graph. Hence Theorem 5.0.9 does not hold in this case. In fact, the least core for this example consists of all allocations of the form

$$
\begin{aligned}
&x_i = 5 - \alpha_1 \quad i \in \{A, B, C\} \\
&x_i = 1 - \alpha_2 \quad i \in \{F, G, H\} \\
&x_C + x_D \geq 1 \\
&x_D + x_E \geq 2 \\
&x_E + x_F \geq 1 \\
&\alpha_1 + \alpha_2 = 2 \\
&0 \leq \alpha_1 \leq 5 \\
&0 \leq \alpha_2 \leq 1
\end{aligned}
$$

The minimum excess achieved by any matching in this example is -4 and in any least core allocation, all matchings that consists of one edge from each of the two triangles have minimum excess. Figures 5.2 and 5.3 show two different allocations in the least core. In particular, note that the edges of the path receive non-negative excess in any least core allocation, while the edges of the triangles receive non-positive excess. Furthermore, the edges of each triangle always receive equal excess. Hence in this example, the components that satisfy the properties of Theorem 5.0.9 are not the odd components of the Edmonds-Gallai decomposition but the two triangles A, B, C and D, E, F.

The key observation about this example is that the least core of G is a strict subset of the least core of the subgraph of G consisting of only the two triangles and the middle edge of the path. Moreover, this subgraph of G is actually node-weighted, hence its own least core is characterized by Theorem 5.0.9.

In the next section we identify some conditions under which this will always be the case. That is, given a graph that is not node-weighted, we can identify a node-weighted subgraph and express the least core of the original graph as a subset of the least core of this subgraph.

5.1 The least core: a special case

In this section we assume that (G, w) is not a node-weighted graph. We show that if G satisfies certain conditions, then one can identify a node-weighted subgraph of G, that we call $G_=$, such that the least core of G can be characterized in terms of the least core $G_=$.

We start by fixing an arbitrary optimal solution y to the linear program D_{FM}. We will later show that our results are independent of this original choice of dual solution. We define $G_=$ be the subgraph of G induced by the edges $e \in E(G)$ that are tight with respect to y, meaning that $y(e) = w(e)$. Then $(G_=, w)$ is node-weighted with respect to the vector y. When referring to $G_=$ we will use y to denote the edges weights with the understanding that $y(e) = y_i + y_j = w(e)$ for all $e = (i, j) \in E(G_=)$.

Note that $\nu(G_=, y) \leq \nu(G, w)$ since any matching M of $G_=$ with weight $y(M)$ is also a matching of G with weight $w(M) = y(M)$. However, it is possible that $\nu(G_=, y) < \nu(G, w)$ if there were some edges of G that were part of every maximum weight matching but were not tight with respect to y. Since y is an optimal solution to D_{FM} it follows from complementary slackness that $\nu_f(G_=, y) = \nu_f(G, w)$. Hence from our assumption that $\nu(G, w) < \nu_f(G, w)$ we also obtain $\nu(G_=, y) < \nu_f(G_=, y)$. Thus we can apply Theorem 5.0.9 to $(G_=, y)$ to obtain the following corollary.

Corollary 5.1.1. *Let (G, w) be a weighted graph with $\nu(G, w) < \nu_f(G, w)$. Let y be an optimal solution to D_{FM} for (G, w) and let $G_=$ be the subgraph of G induced by the set of edges $\{e \in E(G) : y(e) = w(e)\}$. Then leastcore$(G_=, y)$ is equal to the set of vectors x that are optimal solutions to the linear program:*

$$\begin{aligned} max \quad & \epsilon & & (5.3) \\ s.t. \quad & x(e) \geq y(e) & & \text{for all } e \in E(G_=) \setminus \left(\bigcup_{D \in \mathcal{D}(G_=)} E(D) \right) \\ & x_i \leq y_i & & \text{for all } i \in D,\ D \in \mathcal{D}(G_=) \\ & x_i - y_i = x_j - y_j & & \text{for all } i, j \in D,\ D \in \mathcal{D}(G_=) \\ & x(M_1) \geq y(M_1) + \epsilon & & \\ & x(V(G_=)) = \nu(G_=, y) & & \\ & x \geq 0 & & \end{aligned}$$

where $\mathcal{D}(G_=)$ denotes the set of odd components in the Edmonds-Gallai decomposition of the subgraph $G_=$ and M_1 is an arbitrary maximum cardinality matching on the union of these odd components.

For any $D \in \mathcal{D}(G_=)$ with $|D| > 1$ we let $\gamma(D)$ denote the set of edges of G with both endpoints in D. Note that there might be edges in $\gamma(D)$ that are not in $E(D)$ since they were not tight with respect to the dual solution y. We give these type of edges a special name.

Definition 5.1.2. *An edge e is a* chord *if $e \in \gamma(D) \setminus E(D)$ for some $D \in \mathcal{D}(G_=)$.*

We also define the concept of an extendable allocation that will be crucial for our main theorem. Recall that allocations in leastcore$(G_=, y)$ are defined over the set of vertices $V(G_=)$. Essentially an allocation in leastcore$(G_=, y)$ is extendable if we can lift it to an allocation over $V(G)$ by just assigning value zero to vertices in $V(G) \setminus V(G_=)$ without inducing negative excess on any edges in $E(G) \setminus E(G_=)$ that were not chords. We will show later that this implies that the minimum excess of a matching does not decrease when performing this lift.

Definition 5.1.3. *An allocation $x \in leastcore(G_=, y)$ is* extendable *if for every edge $e \in E(G) \backslash E(G_=)$ that is not a chord we have*

$$\sum_{\substack{i \in V(G_=) \\ \delta(i) \ni e}} x_i \geq w(e)$$

We can now present the main result of this section.

Theorem 5.1.4. *Let (G, w) be a weighted graph with $\nu(G, w) < \nu_f(G, w)$. Let y be an optimal solution to D_{FM} for (G, w) and let $G_=$ be the subgraph of G induced by the set of edges $\{e \in E(G) : y(e) = w(e)\}$. Suppose the following conditions hold*

1. $\nu(G_=, y) = \nu(G, w)$
2. *There exists an extendable allocation in $leastcore(G_=, y)$.*

Then

$$leastcore(G, w) = \tilde{P}(\tilde{\epsilon}_1)$$

where $\tilde{P}$ is the linear program

$$\begin{aligned}
(\tilde{P}) \quad & max \quad \epsilon \\
& s.t. \quad x(e) \geq w(e) && \text{for all } e \in E(G) \backslash \left(\bigcup_{D \in \mathcal{D}(G_=)} \gamma(D) \right) \\
& \qquad x_i \leq y_i && \text{for all } i \in D,\ D \in \mathcal{D}(G_=) \\
& \qquad x_i - y_i = x_j - y_j && \text{for all } i, j \in D,\ D \in \mathcal{D}(G_=) \\
& \qquad x(M_1) \geq y(M_1) + \epsilon \\
& \qquad x(V(G)) = \nu(G, w) \\
& \qquad x \geq 0
\end{aligned}$$

$\tilde{e}_1$ is its optimum value and M_1 is a maximum cardinality matching on $\mathcal{D}(G_=)$.

We note that $\tilde{P}$ has a polynomial number of constraints. We can also see that our definition of $\tilde{P}$ is very similar to linear program describing the least core of $(G_=, y)$ in corollary 5.1.1. However there are two main differences. The first difference is that allocations in the least core of $(G_=, y)$ are vectors x over the set $V(G_=)$ that satisfy $x(V(G_=)) = \nu(G_=, y)$, while solutions to the linear program $\tilde{P}$ are vectors over $V(G)$ that satisfy $x(V(G)) = \nu(G, w)$. The first condition of the theorem ensures that the total value of these allocations is always the same. The second difference is that for the least core of $(G_=, y)$ we only required the constraints $x(e) \geq w(e)$ for edges in $E(G_=)$ that are outside the odd components, while in our definition of $\tilde{P}$ we also impose these constraints for edges in $E(G) \backslash E(G_=)$ that are not chords. This is where the second condition of the theorem will come in since it ensures that there exists at least one allocation in least core of $(G_=, y)$ that satisfies these additional constraints.

We now present the proof of Theorem 5.1.4

Proof. Recall from definition 5.0.7 that $\text{leastcore}(G, w) = P(\epsilon_1)$. It then suffices to show that $\epsilon_1 = \tilde{\epsilon}_1$ and $P(\epsilon_1) = \tilde{P}(\epsilon_1)$. We start by showing that $\tilde{P}(\epsilon) \subseteq P(\epsilon)$ for any ϵ. Hence let $x \in \tilde{P}(\epsilon)$ for some ϵ and take any matching $M \in \mathcal{M}(G)$. It follows from the definition of $\tilde{P}$ that the only edges of G that can have negative excess with respect to x are the ones in $\gamma(D)$ for odd components $D \in \mathcal{D}$. Thus we can lower bound the excess of the matching M by considering only these edges. We obtain

$$x(M) - w(M) \geq \sum_{D \in \mathcal{D}} \sum_{e \in M \cap \gamma(D)} x(e) - w(e)$$

Using the fact that y is a feasible solution to D_{FM} we have

$$\sum_{e \in M \cap \gamma(D)} x(e) - w(e) \geq \sum_{e \in M \cap \gamma(D)} x(e) - y(e) = \sum_{i \in D \cap N(M)} x_i - y_i$$

where $N(M)$ denotes the vertices in D that are covered by M.

Now for a given $D \in \mathcal{D}$ it follows again from the definition of $\tilde{P}$ that $x_i - y_i$ is the same for all $i \in D$. Moreover, any matching contained in $\gamma(D)$ can cover at most $|D| - 1$ vertices of D, since $|D|$ is odd. Hence we can lower bound the excess of M by the excess of M_1 and obtain

$$x(M) - w(M) \geq x(M_1) - y(M_1) \geq \epsilon.$$

This shows that $x \in P(\epsilon)$, and therefore we have shown that

$$\tilde{P}(\epsilon) \subseteq P(\epsilon) \quad \forall \epsilon \tag{5.4}$$

as well as

$$\tilde{\epsilon}_1 \leq \epsilon_1. \tag{5.5}$$

For the reverse direction, we first define the following relaxation of P

$$\begin{aligned} (\hat{P}) \quad & \max \epsilon \\ \text{s.t. } & x(M) \geq y(M) + \epsilon \qquad \text{for all } M \in \mathcal{M}(G_=) \\ & x(V(G)) = \nu(G, w) \\ & x \geq 0, \end{aligned}$$

with optimum value $\hat{\epsilon}_1$ and where $\mathcal{M}(G_=)$ denotes the set of matchings of $G_=$. Instead of having a constraint for every matching of G as in the linear program P, in $\hat{P}$ we only have constraints for matchings of $G_=$. Since for any matching $M \in \mathcal{M}(G_=)$ we have $w(M) = y(M)$ the constraints of $\hat{P}$ are a subset of the constraints of P. In particular this implies that

$$\epsilon_1 \leq \hat{\epsilon}_1. \tag{5.6}$$

Note that for any vertex $i \in V(G) \setminus V(G_=)$, there is no matching appearing in the constraints of $\hat{P}$ that covers i. It is then easy to see that for any optimal solution $x \in \hat{P}(\hat{\epsilon}_1)$ we must have $x_i = 0$ for all $i \in V(G) \setminus V(G_=)$. Otherwise we could slightly modify x by decreasing on i and increasing uniformly across all other vertices in $V(G) \setminus \{i\}$ so that $x(V(G))$ stays the same. The resulting allocation will still be in $\hat{P}(\hat{\epsilon}_1)$ but now $x(M)$ strictly increased for each matching M that appears in the constraints of $\hat{P}$. Hence there are no x-tight matchings for the new allocation contradicting the optimality of $\hat{\epsilon}_1$.

Using the first condition of Theorem 5.1.4 we therefore obtain

$$x(V(G_=)) = \nu(G_=, y) \qquad \text{for all } x \in \hat{P}(\hat{\epsilon}_1)$$

Hence $\hat{P}(\hat{\epsilon}_1)$ is the set of vectors $x \in \mathbb{R}^{|V(G)|}$ that are optimal solutions to the linear program

$$\begin{aligned} \max\ & \epsilon \\ \text{s.t.}\ & x(M) \geq y(M) + \epsilon \qquad \text{for all } M \in \mathcal{M}(G_=) \\ & x(V(G_=)) = \nu(G_=, y) \\ & x \geq 0 \end{aligned}$$

Now recall from definition 5.0.7 that the least core of $(G_=, y)$ is the set of vectors $x \in \mathbb{R}^{|V(G_=)|}$ that are optimal solutions to the linear program

$$\begin{aligned} \max\ & \epsilon \\ \text{s.t.}\ & x(M) \geq y(M) + \epsilon \qquad \text{for all } M \in \mathcal{M}(G_=) \\ & x(V(G_=)) = \nu(G_=, y) \\ & x \geq 0 \end{aligned}$$

Hence it follows that

$$\hat{P}(\hat{\epsilon}_1) = \left\{x \in \mathbb{R}^{|V(G)|} : x_{V(G_=)} \in \text{leastcore}(G_=, y) \text{ and } x_i = 0 \text{ for all } i \in V(G) \backslash V(G_=)\right\}$$

where $x_{V(G_=)}$ is the projection of x onto the set $V(G_=)$.

Now let x' be an extendable allocation in leastcore$(G_=, y)$ (guaranteed to exist by the second condition) and define the vector x as follows

$$x_i = \begin{cases} x'_i & i \in V(G_=) \\ 0 & i \in V(G) \backslash V(G_=) \end{cases}$$

Then using corollary 5.1.1 and the fact that x' is extendable we obtain that x must satisfy all of the constraints of $\tilde{P}$ and therefore $x \in \tilde{P}(\hat{\epsilon}_1)$. This shows that

$$\hat{\epsilon}_1 \leq \tilde{\epsilon}_1. \tag{5.7}$$

From equations (5.5), (5.6), and (5.7) we obtain $\epsilon_1 = \hat{\epsilon}_1 = \tilde{\epsilon}_1$. This together with (5.4) implies that $\tilde{P}(\tilde{\epsilon}_1) \subseteq P(\epsilon_1)$.

Hence the proof is complete if we show that $P(\epsilon_1) \subseteq \tilde{P}(\epsilon_1)$. Let $x \in P(\epsilon_1)$. Then $x \in \hat{P}(\epsilon_1)$ and letting x' denote the restriction of x to the vertices in $V(G_=)$ we have $x' \in \text{leastcore}(G_=, y)$. Hence using corollary 5.1.1 again we obtain that x' (and therefore x) satisfies all of the constraints of $\tilde{P}$ except possibly the constraints $x(e) \geq w(e)$ for edges $e \in E(G) \backslash E(G_=)$ that are not chords. Let e be such an edge. Since e is not a chord it does not have both endpoints in the same odd component in $\mathcal{D}$. Since each component in $\mathcal{D}$ is factor critical there must exist a matching M that is a maximum cardinality matching on the union of the odd components $\mathcal{D}$ and such that $M \cup \{e\}$ is a valid matching of G. From corollary 5.1.1 we know that M must have minimum excess with respect to x' and therefore also with respect to x. Hence

$$x(M \cup e) - w(M \cup e) = (x(M) - y(M)) + (x(e) - w(e)) = \epsilon_1 + (x(e) - w(e))$$

and since $x \in P(\epsilon_1)$ this implies that $x(e) \geq w(e)$. Thus $x \in \tilde{P}(\epsilon_1)$ and the proof is complete. □

As an example of a graph whose least core is captured by Theorem 5.1.4 recall our graph from Figure 5.1. This instance has a unique optimal solution y to D_{FM} shown in Figure 5.4. The red edges denote all the tight edges with respect to this solution. Hence

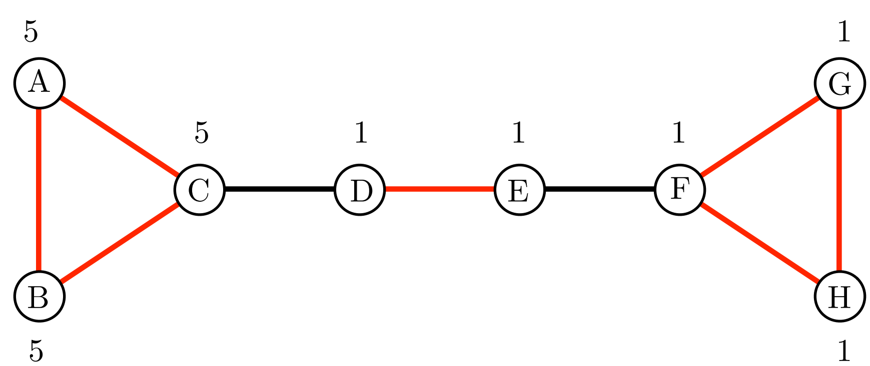

Figure 5.4: Optimal dual solution and equality subgraph.

the graph $G_=$ consists of the two triangles plus the middle edge of the path. Since $G_=$

still contains a matching of weight 14 we have $\nu(G, w) = \nu(G_=, y)$, so the first condition of Theorem 5.1.4 is satisfied. The least core of $(G_=, y)$ is given by all allocations of the form

$$
\begin{aligned}
&x_i = 5 - \alpha_1 \quad i \in \{A, B, C\} \\
&x_i = 1 - \alpha_2 \quad i \in \{F, G, H\} \\
&x_D + x_E \geq 2 \\
&\alpha_1 + \alpha_2 = 2 \\
&0 \leq \alpha_1 \leq 5 \\
&0 \leq \alpha_2 \leq 1
\end{aligned}
$$

To satisfy the second condition of Theorem 5.1.4 we would need an allocation in the least core of $(G_=, y)$ that additionally satisfies $x_C + x_D \geq 1$ and $x_E + x_F \geq 1$. It easy to see that such allocations exists, and both allocations shown in figures 5.2 and 5.3 fit this criteria.

5.1.1 When the first condition of Theorem 5.1.4 fails

While Theorem 5.1.4 describes the least core of certain graphs that are not node-weighted, it still imposes fairly strong conditions. The natural question to ask is what happens with graphs that do not satisfy the conditions of Theorem 5.1.4? Can their least core still be captured using a node-weighted subgraph? We will first look at what happens when the first condition of Theorem 5.1.4 fails. That is, $\nu(G_=, y) < \nu(G, w)$.

An example of this can be seen in figure 5.5 below. The red edges denote the unique maximum weight matching of weight 12. The core is empty since the fractional matching whose support consists of the two triangles has weight $33/2 > 12$. The unique optimal dual solution y to D_{FM} is shown in figure 5.6. The tight edges are the ones belonging to the triangles, hence $G_=$ in this case consists of two disjoint triangles. Note that

$$11 = \nu(G_=, y) < \nu(G, w) = 12.$$

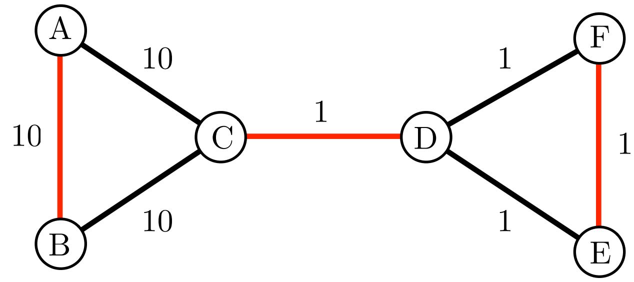

Figure 5.5: An matching game where $\nu(G_=, y) < \nu(G, w)$

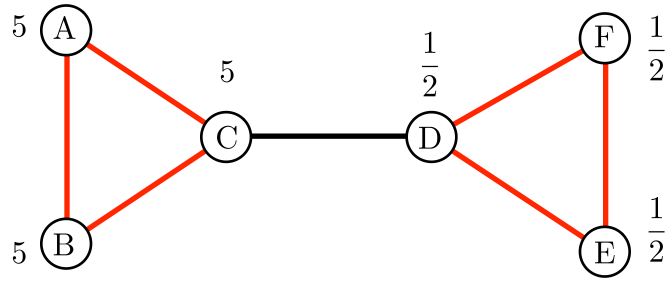

Figure 5.6: Optimal dual solution and equality subgraph.

The least core in this example consists of all allocations x satisfying

$$
\begin{aligned}
&x_i = 5 - \alpha_1 \quad i \in \{A, B, C\} \\
&x_i = 1 - \alpha_2 \quad i \in \{D, E, F\} \\
&\alpha_1 + \alpha_2 = \frac{3}{2} \\
&0 \le \alpha_1 \le 5 \\
&0 \le \alpha_2 \le \frac{1}{2}
\end{aligned}
$$

Figure 5.7 and 5.8 show two different least core allocations. In particular, we can see that the edge (C, D) receives non-negative excess in any least core allocation while the triangle edges receive non-positive excess, and the excess is equal along the edges of each triangle. In some sense we see the same behaviour as before: the least core of (G, w) corresponds to the allocations in the least core of $(G_=, y)$ that satisfy $x_C + x_D \geq 1$. However, the correspondence in this case no longer means that least core of (G, w) is a subset of the least core of $(G_=, y)$ since allocations in the least core of $(G_=, y)$ have total value $\nu(G, y)$ while allocations in the least core of (G, w) have total value $\nu(G, w)$.

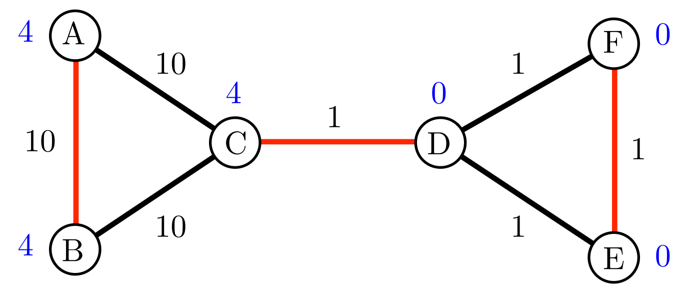

Figure 5.7: A least core allocation with negative excess on both triangles.

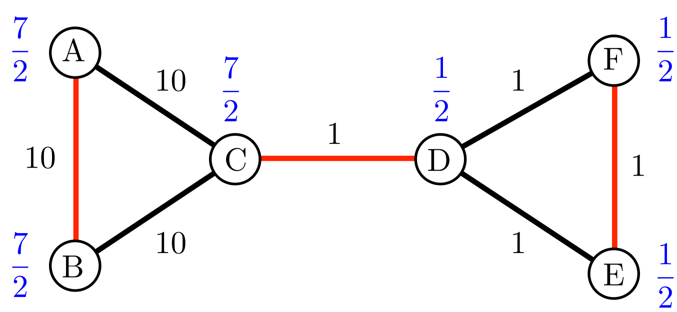

Figure 5.8: A least core allocation with negative excess only on the left triangle.

One way of characterizing the least core of graphs such as the one in figure 5.5 is to prove a more general version Theorem 5.0.9. Instead of the least core, we consider the set of allocations that maximize the minimum excess but assign a total value that can be larger than the maximum weight of matching, but is still strictly less then the maximum weight of a fractional matching. We present this generalised result in the next section.

5.2 A generalisation of the node-weighted result

Recall the linear program $P(G, w)$ that defines the least core of (G, w). For a given $\Delta \geq 0$ suppose we change the program slightly by replacing the constraint $x(V(G)) = \nu(G, w)$ by $x(V(G)) = \Delta$. We then obtain the following definition of the generalised least core.

Definition 5.2.1. *Given a weighted graph (G, w) define the linear program $P(G, w, \Delta)$:*

$$
\begin{aligned}
(P(G, w, \Delta)) \quad & \textit{max } \epsilon \\
& \textit{s.t. } x(M) \geq w(M) + \epsilon \quad \textit{for all } M \in \mathcal{M}(G) \\
& \quad\quad x(V(G)) = \Delta \\
& \quad\quad x \geq 0.
\end{aligned}
$$

Let $\epsilon_1 = \epsilon_1(G, w, \Delta)$ denote the optimal value of $P(G, w, \Delta)$ and $P(G, w, \Delta)(\epsilon)$ the set of vectors x such that (x, ϵ) is feasible for $P(G, w, \Delta)$. Then we define

$$\textit{leastcore}(G, w, \Delta) := P(G, w, \Delta)(\epsilon_1).$$

Note that for $\Delta = \nu(G, w)$ the set $P(G, w, \Delta)(\epsilon_1)$ is just the least core of the matching game (G, w). For a general Δ, we can think $P(G, w, \Delta)(\epsilon_1)$ as just a generalisation of the least core, where instead of having a total value of $\nu(G, w)$ to distribute among the vertices, we now have a total value of Δ. Our objective is the same: maximize the minimum excess $x(M) - w(M)$ of a matching. Note that as long as $\Delta < \nu_f(G, w)$ we will still have $\epsilon_1 < 0$, since $\nu_f(G, w)$ represents the minimum total value required in order to be able to satisfy every edge.

In this section we assume that (G, w) is a node-weighted graph with respect to the node weights $y \in \mathbb{R}_+^{|V(G)|}$. Hence we can replace w by y for the remainder of the section. We review some notation used throughout this section:

- $\mathcal{M}$ is set of matchings of G

- $\mathcal{M}^*$ is the set of maximum cardinality matchings of G

- $(\mathcal{D}, A, \mathcal{C})$ is the Edmonds-Gallai decomposition of G

- $\mathcal{M}_{\mathcal{D}}$ is the set of matchings of G contained in $\bigcup_{D\in\mathcal{D}} E(D)$

- $\mathcal{M}^*_{\mathcal{D}}$ is set of maximum cardinality matchings in $\mathcal{M}_{\mathcal{D}}$

- M_1 is a fixed arbitrary matching in $\mathcal{M}^*_{\mathcal{D}}$.

We also assume as throughout this chapter that $\nu(G, y) < \nu_f(G, y)$, meaning that the core of the corresponding matching game is empty. Recall from lemma 5.0.8 that this implies that there exists at least one $D \in \mathcal{D}$ with $|D| > 1$. We now define a generalisation of the linear program of Theorem 5.0.9 that captures the least core of node-weighted graphs by similarly replacing the constraint $x(V(G)) = \nu(G, y)$ by $x(V(G)) = \Delta$.

Definition 5.2.2. *Given a weighted graph (G, w) that is node-weighted with respect to the vector $y \in \mathbb{R}_+^{|V(G)|}$ define the linear program $\tilde{P}(G, y, \Delta)$ as*

$$
\begin{array}{llll}
(\tilde{P}(G,y,\Delta)) & max & \epsilon & \\
 & s.t. & x(e) \geq w(e) & \text{for all } e \in E(G)\setminus \left(\bigcup_{D\in\mathcal{D}(G)} E(D)\right) \\
 & & x_i \leq y_i & \text{for all } i \in D,\ D \in \mathcal{D}(G) \\
 & & x_i - y_i = x_j - y_j & \text{for all } i, j \in D,\ D \in \mathcal{D}(G) \\
 & & x(M_1) \geq y(M_1) + \epsilon & \\
 & & x(V(G)) = \Delta & \\
 & & x \geq 0 &
\end{array}
$$

and let $\tilde{\epsilon} = \tilde{\epsilon}_1(G, y, \Delta)$ denotes its optimum value. As before, $\mathcal{D}(G)$ denotes the set of odd components in the Edmonds-Gallai decomposition of G and M_1 is a fixed maximum cardinality matching on the union of these odd components.

Our main result is the following.

Theorem 5.2.3. *Let (G, y) be a node-weighted graph with respect to the node weights $y \in \mathbb{R}_+^{|V(G)|}$ and such that $\nu(G, y) < \nu_f(G, y)$. Suppose that $\nu(G, y) \leq \Delta < \nu_f(G, y)$. Then*

(i) $\epsilon_1(G, y, \Delta) = \tilde{\epsilon}_1(G, y, \Delta)$ *and*

(ii) $P(G, y, \Delta)(\epsilon_1(G, y, \Delta))) = \tilde{P}(G, y, \Delta)(\tilde{\epsilon}_1(G, y, \Delta)))$.

Note that for $\Delta = \nu(G, y)$ our statement is equivalent to Theorem 5.0.9, hence it suffices to prove it for $\nu(G, y) < \Delta < \nu_f(G, y)$. For the remainder of this section we drop the notations (G, y, Δ) when referring to the linear programs P and $\tilde{P}$.

To prove Theorem 5.2.3 we follow the proof of Paulusma [68] step by step, pointing out where differences arise due to the replacement of $\nu(G, y)$ by Δ. The proof of Paulusma proceeds by a series of lemmas. Our proof will use the exact same lemmas, whose statements remain unchanged. Most of the proofs will also be identical to the ones in [68], and in most instances where there is a change the proof will actually become easier since $\Delta > \nu(G, y)$. The only lemma where this is not the case is lemma 5.2.9, where an entirely different proof is needed to account for our use of Δ.

5.2.1 Proof outline

We first give a short outline of the main structure of the proof.

1. In the first and easier part we will show that $\tilde{P}(\epsilon) \subseteq P(\epsilon)$ for any ϵ. This will follow mostly from the definition of $\tilde{P}$ and it will also imply that

$$\tilde{\epsilon}_1 \leq \epsilon_1.$$

2. In the second and more difficult part we will show that $\epsilon_1 \leq \tilde{\epsilon}_1$. We will accomplish this by introducing a relaxation $\hat{P}$ of P whose optimum value we will denote by $\hat{\epsilon}_1$. Since $\hat{P}$ is a relaxation we will automatically have

$$\epsilon_1 \leq \hat{\epsilon}_1.$$

Moreover, $\hat{P}$ will be easier to analyze then P and will allow us to prove that optimum solutions to $\hat{P}$ have similar structural properties to optimum solutions of $\tilde{P}$. More specifically we will show that there exists at least one special type of allocation $x \in \hat{P}(\hat{\epsilon}_1)$, called a flexible allocation, that can be modified into an allocation $\tilde{x}$ such that $\tilde{x} \in \tilde{P}(\hat{\epsilon}_1)$. This will imply that

$$\hat{\epsilon}_1 \leq \tilde{\epsilon}_1.$$

And thus given our previous inequalities we will have established that

$$\epsilon_1 = \hat{\epsilon}_1 = \tilde{\epsilon}_1.$$

3. In the third and final part we show that $P(\epsilon_1) \subseteq \tilde{P}(\epsilon_1)$. Here we will use our relaxation $\hat{P}$ again since any allocation in $P(\epsilon_1)$ will also be in $\hat{P}(\epsilon_1) = \hat{P}(\hat{\epsilon}_1)$. Then using the structural properties of optimum solutions to $\hat{P}$ we will show that the given allocation also satisfies the constraints of $\tilde{P}$.

5.2.2 Proof of the first part of Theorem 5.2.3

We start by showing the first part of Theorem 5.2.3.

Lemma 5.2.4. *$\tilde{P}(\epsilon) \subseteq P(\epsilon)$ for any ϵ and consequently $\tilde{\epsilon}_1 \leq \epsilon_1$.*

Proof. Let $x \in \tilde{P}_1(\epsilon)$ and let M be any matching of G. Then we can decompose the excess of M as follows:

$$x(M) - y(M) = x(M \cap E(\mathcal{D})) - y(M \cap E(\mathcal{D})) + x(M \backslash E(\mathcal{D})) - y(M \backslash E(\mathcal{D})),$$

and since $x(e) \geq y(e)$ for all $e \in E(G) \backslash E(\mathcal{D})$ we have

$$x(M) - y(M) \geq x(M \cap E(\mathcal{D})) - y(M \cap E(\mathcal{D})).$$

Now since $x_i \leq y_i$ for all $i \in D$ and for any $D \in \mathcal{D}$ we can lower bound the excess of M by the excess of the fixed maximum cardinality matching M_1. Hence

$$x(M) - y(M) \geq x(M_1) - y(M_1) \geq \epsilon.$$

Thus, $x \in P_1(\epsilon)$ as required. □

Hence, to prove Theorem 5.2.3 it remains to show that $\epsilon_1 \leq \tilde{\epsilon}_1$ and $P_1(\epsilon_1) \subseteq \tilde{P}(\epsilon_1)$. We introduce a similar relaxation of $P(\epsilon)$ as in the proof of Theorem 5.1.4.

$$
\begin{aligned}
(\hat{P}) \quad & \max \epsilon \\
& \text{s.t. } x(M) \geq y(M) + \epsilon \quad \text{for all } M \in \mathcal{M}_{\mathcal{D}} \cup \mathcal{M}^* \\
& \qquad x(V(G)) = \Delta \\
& \qquad x \geq 0,
\end{aligned}
$$

with optimum value $\hat{\epsilon}_1$. As before, the set $\hat{P}(\epsilon)$ contains the set of vectors x such that (x, ϵ) is feasible for $\hat{P}$. Since $\hat{P}$ is a relaxation of P we automatically have $P(\epsilon) \subseteq \hat{P}(\epsilon)$ for any ϵ and therefore $\epsilon_1 \leq \hat{\epsilon}_1$.

Instead of having a constraint for every matching of G, in $\hat{P}$ we only include the constraints for maximum cardinality matchings and matchings completely contained in the union of the odd components. The goal behind the definition of $\hat{P}$ is two-fold: on one hand, $\hat{P}$ contains only a small subset of the constraints of P and thus it should hopefully be easier to analyze. On the other hand, we want $\hat{P}$ to maintain the same optimal value as P. That is, even though we have fewer matchings that we need to satisfy in $\hat{P}$, we should not be able to increase the minimum excess. This means that we expect the maximum cardinality matchings together with the matchings in the odd components to be the ones that limit the optimum value of P.

Our goal is to show that optimal allocations in $\hat{P}(\hat{\epsilon}_1)$ satisfy most of the properties of $\tilde{P}$. Then given such an allocation $x \in \hat{P}(\hat{\epsilon}_1)$ we will obtain from x a new allocation in $\tilde{P}(\hat{\epsilon}_1)$. This will then imply that $\hat{\epsilon}_1 \leq \tilde{\epsilon}_1$ and thus we will obtain that the optimal values of all the three linear programs are the same. Once this is established, the proof will be completed by showing that any optimal allocation in $P(\epsilon_1)$ also belongs to $\tilde{P}(\epsilon_1)$.

5.2.3 Existence of x-tight allocations on the odd components

For $x \in \hat{P}(\epsilon)$ and $M \in \mathcal{M}^* \cup \mathcal{M}_{\mathcal{D}}$ we say that M is *x-tight* if $x(M) = y(M) + \epsilon$. The following lemma is crucial toward eventually arguing that the odd components in $\mathcal{D}$ are the ones accounting for any negative excess.

Lemma 5.2.5. *Let $x \in \hat{P}(\epsilon)$ for $\epsilon < 0$. If there are no x-tight matchings in $\mathcal{M}_{\mathcal{D}}$ then ϵ is not optimal.*

Proof. Let $x \in \hat{P}(\epsilon)$ and suppose that there are no x-tight matchings in $\mathcal{M}_{\mathcal{D}}$. We may assume that some matching from the set $\mathcal{M}^*$ must be x-tight since otherwise ϵ cannot be optimal. Let U be the set of all vertices $i \in \mathcal{D}$ with $x_i > 0$. We claim that each x-tight matching $M \in \mathcal{M}^*$ must expose at least one vertex in U. If this was not the case, then since M covers all vertices outside $\mathcal{D}$ (recall the Edmonds-Gallai decomposition) we would have $x(M) = x(V(G))$. But then

$$x(M) = x(V(G)) = \Delta > \nu(G, y) \geq y(M)$$

Since $\epsilon < 0$, this implies that M cannot be x-tight. Let j be an arbitrary vertex in $A \cup \mathcal{C}$. Note that every matching $M \in \mathcal{M}^*$ must cover j. We can then define a new allocation x' as follows:

$$x'_i = x_i - \delta \qquad \forall\, i \in U$$
$$x'_j = x_j + \delta|U|$$

where δ is chosen small enough such that $x'_i \geq 0$ for all $i \in U$ and no matching $M \in \mathcal{M}_{\mathcal{D}}$ becomes x'-tight. Then since every x-tight matching exposes at least one vertex in U and covers vertex j we have $x'(M) > x(M)$ for all x-tight matchings. Thus $x' \in \hat{P}_1(\epsilon)$ and there are no x'-tight matchings, meaning that ϵ is not optimal. □

5.2.4 Averaging over the odd components

In this subsection we will work toward showing that for any optimal allocation in $x \in \hat{P}(\hat{\epsilon}_1)$ we must have $x_i - y_i = x_j - y_j$ for all $i, j \in D$, $D \in \mathcal{D}$. For this, we will need the following definition.

Definition 5.2.6. *Given a vector $x \in \mathbb{R}_+^{|V(G)|}$ and $D \in \mathcal{D}$ we define the average of x and y on D as*

$$x_D := \frac{x(D)}{|D|} \quad y_D := \frac{y(D)}{|D|}$$

We then let $\overline{x} \in \mathbb{R}_+^{|V(G)|}$ *be the vector obtained by averaging* x *on each odd component* $D \in \mathcal{D}$ *with respect to the node weights* y*:*

$$\overline{x}_i := x_D - y_D + y_i \quad \textit{for all } i \in D, D \in \mathcal{D}$$

and leaving x *unchanged on* $A \cup \bigcup \mathcal{C}$*. Note that* $\overline{x}$ *satisfies* $\overline{x}_i - y_i = \overline{x}_j - y_j$ *for all* $i, j \in D, D \in \mathcal{D}$ *and* $\overline{x}(D) = x(D)$ *for all* $D \in \mathcal{D}$*.*

As an initial step we first have to show that whenever x is an optimal allocation so is $\overline{x}$. This seems intuitive since the average excess will be no worse then the minimum excess of an edge. The next lemma proves that this intuition is correct.

Lemma 5.2.7. *Let* $x \in \hat{P}(\epsilon)$*. Then* $\overline{x}(M) \geq y(M) + \epsilon$ *for all* $M \in \mathcal{M}^* \cup \mathcal{M}_{\mathcal{D}}$*.*

Proof. Let $x \in \hat{P}(\epsilon)$. It suffices to show that the constraints are satisfied after averaging x on a single component $D \in \mathcal{D}$. Thus let $D \in \mathcal{D}$ and let x' be given by $x'_i = x_D - y_D + y_i$ for all $i \in V(D)$ and $x'_i = x_i$ for all $i \in V(G) \backslash V(D)$. We first consider the case of a matching $M \in \mathcal{M}^*$. By properties of the Edmonds-Gallai decomposition we know that M either covers all vertices of D or exposes exactly one vertex of D. If M covers all vertices of D then $x'(M) = x(M)$ and we are done. Otherwise there is some vertex $i \in D$ that is not covered by M. Recall that D is factor critical. We may assume wlog that i maximizes $x_i - y_i$ over D since otherwise we can replace M inside D by a near perfect matching that exposes i and the excess will only decrease. But then

$$x_i - y_i \geq x_D - y_D = x'_i - y_i$$

and therefore $x'(M) - y(M) \geq x(M) - y(M)$ and are done.

It remains to consider the case of a matching $M \in \mathcal{M}_{\mathcal{D}}$. Since we are lower bounding the excess of M with respect to x', we may assume wlog that M is a matching in $\mathcal{M}_{\mathcal{D}}$ with minimum excess. If $x(e) > y(e)$ for all $e \in \mathcal{M}_{\mathcal{D}}$ then all matchings in $\mathcal{M}_{\mathcal{D}}$ have strictly positive excess with respect to both x and x'. Hence we may assume that there exists at least one edge in $\mathcal{M}_{\mathcal{D}}$ such that $x(e) < y(e)$. Then since M is a matching in $\mathcal{M}_{\mathcal{D}}$ with minimum excess it follows that M only contains edges with negative excess. Hence if $x_D - y_D > 0$, then M cannot contain any edge of D and therefore $x(M) = x'(M)$ and we are done. On the other hand if $x_D - y_D \leq 0$ then we may assume wlog that M is near-perfect matching on D. Then the argument is the same as for matchings in $\mathcal{M}^*$. $\square$

We are now able to show the full result.

Lemma 5.2.8. *Let $x \in \hat{P}(\hat{\epsilon}_1)$. Then $\overline{x} \in \hat{P}(\hat{\epsilon}_1)$.*

Proof. It follows from the definition of $\overline{x}$ that $\overline{x}(V(G)) = x(V(G))$. From lemma 5.2.7 we also know that $\overline{x}$ satisfies $\overline{x}(M) - y(M) \geq \hat{\epsilon}_1$ for all $M \in \mathcal{M}^* \cap \mathcal{M}_{\mathcal{D}}$. Hence the only thing that remains to be checked is $\overline{x} \geq 0$. We let $y^D_{\min} = \min\{y_i : i \in D\}$ for all $D \in \mathcal{D}$. Then using the definition of $\overline{x}$ it suffices to prove for all $D \in \mathcal{D}$ that

$$x_D \geq y_D - y^D_{\min}.$$

Suppose by contradiction that $x_{D'} < y_{D'} - y^{D'}_{\min}$ for some $D' \in \mathcal{D}$. Then using the definitions of x_D and y_D this is equivalent to

$$\sum_{i \in D'} x_i < \sum_{i \in D'} \left(y_i - y^{D'}_{\min}\right).$$

This implies that $|D'| > 1$ since otherwise the right hand side would be zero, and we know that $x \geq 0$. We may modify x if necessary on the vertices of D' while keeping $x(D')$ the same to ensure that $x_i \leq y_i - y^{D'}_{\min}$ for all $i \in D'$. It is not too hard to check that x will still belong to $\hat{P}(\hat{\epsilon}_1)$ after this modification. But then there must exist at least one vertex $k \in D'$ such that $x_k < y_k - y^{D'}_{\min}$. It follows that k must be covered by x-tight matching $M \in \mathcal{M}^* \cup \mathcal{M}_{\mathcal{D}}$.

We now show that we can modify x to obtain a new allocation x' such that $(x', \hat{\epsilon}_1)$ is still feasible for $\hat{P}$ but there are no x'-tight matchings. This would then contradict the optimality of $\hat{\epsilon}_1$. We consider two cases. First, if there exists a vertex $i \in V(G)$ that is outside the odd components and $x_i > 0$, then we can decrease x on i and increase x on k by a sufficiently small amount so that the new allocation x' remains feasible. But then there are no x'-tight matchings in $\mathcal{M}_{\mathcal{D}}$, hence by lemma 5.2.5 $\hat{\epsilon}_1$ cannot be optimal. For the second case suppose that $x_i = 0$ for all vertices $i \in V(G)$ that are outside the odd components. Then

$$\bigcup_{D \in \mathcal{D}} x(D) = x(V(G)) = \Delta > \nu(G, y) \geq \sum_{D \in \mathcal{D}} y(D) - y^D_{\min}$$

Since $x_{D'} < y_{D'} - y^{D'}_{\min}$ there must exist a component $\tilde{D} \in \mathcal{D}$ such that $x_{\tilde{D}} > y_{\tilde{D}} - y^{\tilde{D}}_{\min}$. By lemma 5.2.7 we may assume that x is averaged on $\tilde{D}$ and therefore $x_i > 0$ for all $i \in \tilde{D}$. We can then decrease x uniformly on the vertices of $\tilde{D}$ and increase x by the same amount on x_k, choosing a sufficiently small amount so that the new allocation x' remains feasible. Since every x-tight matching in $\mathcal{M}_{\mathcal{D}}$ covers k but must expose at least one vertex of $\tilde{D}$ (odd component) the excess of all such matchings will increase and there will be no x'-tight matchings in $\mathcal{M}_{\mathcal{D}}$. We then arrive at the same contradiction as in the previous case. □

Recall that one of the objectives behind the definition of $\hat{P}$ was to maintain the same optimal value as P. The next lemma establishes a preliminary result towards showing that the optimal values of P and $\hat{P}$ are the same.

Lemma 5.2.9. $\hat{\epsilon}_1 < 0$.

Proof. Let $x \in \hat{P}(\hat{\epsilon}_1)$. By lemma 5.2.8 we may assume that $x = \overline{x}$. If there exists an edge in the odd components with negative excess then clearly $\hat{\epsilon}_1 < 0$. Hence we may assume that $\overline{x}_D \geq y_D$ for all $D \in \mathcal{D}$.

Now take any matching $M \in \mathcal{M}^*$ and recall that for every odd component $D \in \mathcal{D}$ either M covers all the vertices of D or M exposes exactly one vertex of D. Let $\mathcal{D}' \subseteq \mathcal{D}$ denote the set of odd components that have a vertex exposed by M. Since M covers all vertices outside the odd components we have

$$\begin{aligned}\overline{x}(M) - y(M) &= \overline{x}(V(G)) - y(V(G)) - \sum_{D \in \mathcal{D}'} (\overline{x}_D - y_D) \\ &\leq \overline{x}(V(G)) - y(V(G)) \\ &= \Delta - y(V(G)) \\ &< 0,\end{aligned}$$

since $\nu(G, y) < \Delta < y(V(G))$. Hence M has negative excess and therefore $\hat{\epsilon}_1 < 0$. □

For a given $\overline{x} \in \hat{P}(\hat{\epsilon}_1)$ we let $\mathcal{D}_{\max} \subseteq \mathcal{D}$ be the set of odd components on which $\overline{x}_D - y_D$ is maximum among all $D \in \mathcal{D}$.

Lemma 5.2.10. *Let $\overline{x} \in \hat{P}(\hat{\epsilon}_1)$. If there are $\overline{x}$-tight matchings in $\mathcal{M}^*$, then for each $D \in \mathcal{D}_{\max}$ there is a $\overline{x}$-tight $M \in \mathcal{M}^*$ that exposes a vertex of D. Furthermore, any $\overline{x}$-tight matching must expose a vertex of at least one $D \in \mathcal{D}_{\max}$.*

Proof. Suppose $M \in \mathcal{M}^*$ is $\overline{x}$-tight and covers all vertices of D for some $D \in \mathcal{D}_{\max}$. Recall from the definition of the Edmonds-Gallai decomposition that for every odd component there exists a maximum cardinality matching that exposes one of its vertices. Hence let M' be any matching in $\mathcal{M}^*$ that exposes a vertex of D. Let $P \subseteq M \cup M'$ be the unique maximal alternating path starting in D at a vertex i exposed by M' and ending in some other odd component D' at a vertex j exposed by M. Reversing M along P results in another maximum cardinality matching $\bar{M}$, that covers the vertices $(N(M)\backslash i) \cup j$. Since $D \in \mathcal{D}_{\max}$ it follows that $\overline{x}_i - y_i \geq \overline{x}_j - y_j$. Therefore the new matching $\bar{M}$ is also $\overline{x}$-tight. To prove the second claim note that if M would cover all the vertices of every $D \in \mathcal{D}_{\max}$ then $D' \notin \mathcal{D}_{\max}$ meaning that $\bar{M}$ has strictly lower excess then M. But this contradicts the fact that M is $\overline{x}$-tight. □

5.2.5 Structural properties of optimal allocations

We are now able to prove some important properties about optimal allocations in $\hat{P}(\hat{\epsilon}_1)$. Recall that our overall goal is to show that optimal allocations in $\hat{P}(\hat{\epsilon}_1)$ satisfy the constraints of $\tilde{P}$. In particular the constraints of $\tilde{P}$ imply that the excess must be equal and non-positive on each of the odd components. The next lemma shows that this is also the case for $\hat{P}(\hat{\epsilon}_1)$.

Lemma 5.2.11. *Let $x \in \hat{P}(\hat{\epsilon}_1)$. Then*

1. $x = \overline{x}$
2. $x_i \leq y_i$ *for all* $i \in D$, $D \in \mathcal{D}$
3. *Each* $M \in \mathcal{M}^*_D$ *is* x*-tight*

Proof. Let $x \in \hat{P}(\hat{\epsilon}_1)$. We first show (2) and (3) for $\overline{x}$ and then prove that $x = \overline{x}$.

(2) Suppose by contradiction that $\overline{x}_D > y_D$ for some $D \in \mathcal{D}_{\max}$. Then $\overline{x}_i > y_i$ for all $i \in D, D \in \mathcal{D}_{\max}$. This implies that $\mathcal{D}_{\max}$ must be a strict subset of $\mathcal{D}$ since otherwise there would be no $\overline{x}$-tight matchings in $\mathcal{M}_\mathcal{D}$ contradicting lemma 5.2.5. Furthermore, any x-tight matching in $\mathcal{M}_\mathcal{D}$ does not contain any edges from components in $\mathcal{D}_{\max}$. Now if $A \cup \mathcal{C} = \emptyset$ then we can define a new allocation $\overline{x}'$ by decreasing $\overline{x}$ uniformly on $\mathcal{D}_{\max}$ by a total value δ, where δ is chosen sufficiently small so that $\overline{x}'_i \geq y_i$ for all $i \in D, D \in \mathcal{D}_{\max}$; and then increasing $\overline{x}$ uniformly on $\mathcal{D} \backslash \mathcal{D}_{\max}$ by a total value of δ. Then $\overline{x}'$ will be a feasible allocation that has no $\overline{x}'$-tight matchings thus contradicting the optimality of $\hat{\epsilon}_1$. Hence we may assume that $A \cup \mathcal{C} \neq \emptyset$. Then let $j \in A \cup \mathcal{C}$ and define a new allocation $\overline{x}'$ by decreasing $\overline{x}$ uniformly on $\mathcal{D}_{\max}$ by a total value δ, where δ is chosen sufficiently small so that $\overline{x}'_i \geq y_i$ for all $i \in D, D \in \mathcal{D}_{\max}$; and then increasing $\overline{x}$ uniformly on $\{j\} \cup \mathcal{D} \backslash \mathcal{D}_{\max}$ by a total value of δ. Then it is easy to see that $\overline{x}'$ will again be a feasible allocation that has no $\overline{x}'$-tight matchings.

(3) From lemma 5.2.5 we know that there exists a $\overline{x}$-tight matching $M \in \mathcal{M}_\mathcal{D}$. Then from (2) we may assume that $M \in \mathcal{M}^*_\mathcal{D}$. Finally note that the excess of any matching $M \in \mathcal{M}^*_\mathcal{D}$ is equal since

$$\overline{x}(M) - y(M) = \sum_{D \in \mathcal{D}} (|D| - 1)(\overline{x}_D - y_D).$$

(1) Let M be a matching in $\mathcal{M}^*_\mathcal{D}$ with minimum excess with respect to x. Then $x(M) - y(M) \leq \overline{x}(M) - y(M)$ with equality if and only if $x = \overline{x}$. But from (3) we have

$$\overline{x}(M) - y(M) = \hat{\epsilon}_1$$

and since $x \in \hat{P}(\hat{\epsilon}_1)$ we must have $x(M) \geq y(M) + \hat{\epsilon}_1$. Thus $x = \overline{x}$. □

We point out here that while this result is crucial to establish the desired behaviour of solutions in $\hat{P}(\hat{\epsilon}_1)$ over the odd components $\mathcal{D}$ this lemma is not sufficient to show our end goal that we can get an allocation in $\hat{P}(\hat{\epsilon}_1)$ that satisfies all the constraints of $\tilde{P}$. In particular over the next sections we will prove the crucial property that edges outside the odd components receive non-negative excess.

We have established so far that maximum cardinality matchings on the odd components are always x-tight. Our next step is to show that there are maximum cardinality matchings of G that are also x-tight. The next lemma considers a special case where vertices outside the odd components receive zero value.

Lemma 5.2.12. *Let $x \in \hat{P}(\hat{\epsilon}_1)$. If $x(A \cup \mathcal{C}) = 0$ then every $M \in \mathcal{M}^*$ is x-tight.*

Proof. If $A \cup \mathcal{C} = \emptyset$ then $\mathcal{M}^* = \mathcal{M}^*_{\mathcal{D}}$ so the lemma is true by lemma 5.2.11. Hence suppose that $A \cup \mathcal{C} \neq \emptyset$ and $x(A \cup \mathcal{C}) = 0$. Recall that every maximum cardinality matching $M \in \mathcal{M}^*$ can be decomposed as

$$M = M_{\mathcal{C}} \cup M_{A,\mathcal{D}} \cup M_{\mathcal{D}}$$

where $M_{\mathcal{C}}$ is a perfect matching in the union of all even components $\mathcal{C}$; $M_{\mathcal{D}}$ induces a near-perfect matching in all odd components $D \in \mathcal{D}$; and $M_{A,\mathcal{D}}$ is a matching that matches A completely into $\cup\mathcal{D}$ the union of odd components. By lemma 5.2.11 we know that $M_{\mathcal{D}}$ is x-tight, therefore

$$x\left(M_{\mathcal{C}} \cup M_{A,\mathcal{D}}\right) \geq y\left(M_{\mathcal{C}} \cup M_{A,\mathcal{D}}\right).$$

Now let B be the vertices in $\mathcal{D}$ that are connected to vertices in A by $M_{A,\mathcal{D}}$. Then

$$y\left(M_{\mathcal{C}} \cup M_{A,\mathcal{D}}\right) = y(\mathcal{C} \cup A) + y(B).$$

By lemma 5.2.11 we have $x_i \leq y_i$ for all $i \in B$ and since we assume that $x(A \cup \mathcal{C}) = 0$ we obtain

$$y(\mathcal{C} \cup A) + y(B) \leq x(B) \leq y(B)$$

meaning that $y_i = 0$ for all $i \in \mathcal{C} \cup A$ and $x_i = y_i$ for all $i \in B$. This means that

$$\begin{aligned} x(M) &= x(M_{\mathcal{C}}) + x(A) + x(B) + x(M_{\mathcal{D}}) \\ &= y(B) + y(M_{\mathcal{D}}) + \hat{\epsilon}_1 \\ &= y(M) + \hat{\epsilon}_1 \end{aligned}$$

and therefore M is x-tight as required. □

We finish this subsection with the following useful technical lemma.

Lemma 5.2.13. *Let $x \in \hat{P}(\hat{\epsilon}_1)$. Then there exists some x-tight $M \in \mathcal{M}^*$. Moreover, if $D \in \mathcal{D}_{\max}$ or $|D| > 1$ then there is some x-tight $M \in \mathcal{M}^*$ exposing a vertex of D.*

Proof. If $A \cup \mathcal{C} = \emptyset$ then $\mathcal{M}^* = \mathcal{M}^*_{\mathcal{D}}$ and all of these matchings are x-tight by lemma 5.2.11. On the other hand if $x(A \cup \mathcal{C}) = 0$ then again every matching in $\mathcal{M}^*$ is x-tight from lemma 5.2.12. Hence the proof follows since for every D, there exists an x-tight matching in $\mathcal{M}^*$ exposing a vertex of D.

Now assume that assume that $A \cup \mathcal{C} \neq \emptyset$ and $x(A \cup \mathcal{C}) > 0$. Suppose by contradiction that there are no x-tight matchings in $\mathcal{M}^*$. Then we can decrease x uniformly on $A \cup \mathcal{C}$ and increase x uniformly by the same amount on $\mathcal{D}$ where this amount is chose sufficiently small so that the new allocation is feasible but has no x-tight matchings in $\mathcal{M}^*_{\mathcal{D}}$ thus contradicting lemma 5.2.11.

Hence there must be some x-tight matching in $\mathcal{M}^*$. Then by lemma 5.2.10 we know that for every $D \in \mathcal{D}_{\max}$ there is an x-tight matching in $\mathcal{M}^*$ exposing a vertex of D. We are left to prove the same result for $D \in \mathcal{D} \backslash \mathcal{D}_{\max}$ with $|D| > 1$. Let D be such a component. Then $x_i < y_i$ for all $i \in D$ by lemma 5.2.11. If D is covered by every x-tight $M \in \mathcal{M}^*$ then since $x(A \cup \mathcal{C}) > 0$ we can decrease x on some $j \in A \cup \mathcal{C}$ with $x_j > 0$ and increase x by the same amount uniformly on the vertices of D. Then again we obtain a new feasible allocation with no x-tight matching in $\mathcal{M}^*_{\mathcal{D}}$, contradicting lemma 5.2.11. □

5.2.6 Flexible allocations

Our general strategy is to show that there exists a certain allocation $x \in \hat{P}(\hat{\epsilon}_1)$ from which we can obtain a new allocation $\tilde{x}$ that satisfies all of the constraints of $\tilde{P}$ with the same $\hat{\epsilon}_1$. Our choice of allocation in $\hat{P}(\hat{\epsilon}_1)$ will therefore have to satisfy some useful properties. The following definition makes these properties exact.

Recall that given any odd component $D \in \mathcal{D}$ and a maximum cardinality matching $M \in \mathcal{M}^*$ then M either covers all vertices of D or exposes exactly one vertex of D. In the latter case we say that M exposes D.

Definition 5.2.14. *An allocation $x \in \hat{P}(\hat{\epsilon}_1)$ is* flexible *if for each $D \in \mathcal{D}$ there is some x-tight $M \in \mathcal{M}^*$ exposing D.*

Lemma 5.2.15. *Flexible allocations exists.*

Proof. Let $x \in \hat{P}(\hat{\epsilon}_1)$ and suppose that x is not flexible. Then by lemma 5.2.13 there exists a singleton component $D \in \mathcal{D}$ with $D = \{i\}$ such that every x-tight matching in $\mathcal{M}^*$ covers i. This implies that $A \neq \emptyset$ and by lemma 5.2.12 we must also have $x(A \cup \mathcal{C}) > 0$, since otherwise every maximum cardinality matching is x-tight contradicting our assumption that x is not flexible. We now decrease x uniformly on vertices $j \in A \cup \mathcal{C}$ with $x_j > 0$ and increase x_i by the same amount until some matching $M \in \mathcal{M}^*$ that exposes i becomes tight. Note that this modification preserves all the x-tight matchings in $\mathcal{M}^*$. Hence the claim follows by induction. □

Given a flexible allocation $\hat{x} \in \hat{P}(\hat{\epsilon}_1)$ we let $\alpha_0 < \cdots < \alpha_p$ denote the different values that $\hat{x}_D - y_D$ takes on each $D \in \mathcal{D}$, and let

$$\mathcal{D} = \mathcal{D}_0 \cup \cdots \cup \mathcal{D}_p$$

be the corresponding partition of $\mathcal{D}$. Then $\hat{x}_i - y_i = \alpha_i$ for all $i \in D$ where $D \in \mathcal{D}_i$, and $\mathcal{D}_p = \mathcal{D}_{\max}$.

Lemma 5.2.16. *There exists a partition $A = A_0 \cup \cdots \cup A_p$, with some of the A_i possibly empty, such that $M \in \mathcal{M}^*$ is $\hat{x}$-tight if and only if M matches each non-empty A_i completely into $\mathcal{D}_i$.*

Proof. If $A = \emptyset$ the lemma is trivially true. Hence assume that $A \neq \emptyset$. Since $\hat{x}$ equalizes the excess on each odd component (by lemma 5.2.11) we can see that the excess of a matching $M \in \mathcal{M}^*$ depends only on how many nodes of A are matched into each $\mathcal{D}_i$. The lemma then follows from the following claim.

Claim 1. Let H be a bipartite graph with node classes A and B. Suppose that $B = B_0 \cup \cdots \cup B_p$ is a partition of B and edges incident with vertices of B_i have weight α_i, where $\alpha_0 < \cdots < \alpha_p$. Let $\mathcal{M}^*$ be the set of matchings that completely matches A into B, and assume that $\mathcal{M}^* \neq \emptyset$. Let $\mathcal{M}^*_{\min}$ be the set of matchings in $\mathcal{M}^*$ of minimum weights. Finally suppose that each $b \in B$ is left uncovered by some $M \in \mathcal{M}^*_{\min}$. There exists a partition $A = A_0 \cup \cdots \cup A_p$, with some of the A_i possibly empty, such that $M \in \mathcal{M}^*_{\min}$ if and only if M matches each non-empty A_i completely into B_i.

We omit the proof of this claim since it identical to lemma 4.15 in [68] and the result is independent of our use of Δ. $\square$

5.2.7 Completing the proof of Theorem 5.2.3

We are now ready to complete the proof of Theorem 5.2.3. This is accomplished by the following two final lemmas.

Lemma 5.2.17. $\epsilon_1 = \hat{\epsilon}_1 = \tilde{\epsilon}_1$

Proof. In lemma 5.2.4 we already showed that $\tilde{\epsilon}_1 \leq \epsilon_1$. Moreover we also have observed that $\epsilon_1 \leq \hat{\epsilon}_1$ since $\hat{P}$ was defined as a relaxation of P. Hence to prove the lemma it suffices to show that $\hat{\epsilon}_1 \leq \tilde{\epsilon}_1$. We do this by constructing an allocation $x \in \tilde{P}(\hat{\epsilon}_1)$.

Let $\hat{x} \in \hat{P}(\hat{\epsilon}_1)$ be a flexible allocation with corresponding partition $\mathcal{D} = \mathcal{D}_0 \cup \cdots \cup \mathcal{D}_p$ and $A = A_0 \cup \cdots \cup A_p$. We define a new allocation $\tilde{x}$ as follows

$$\tilde{x}_i = \begin{cases} y_i & i \in \mathcal{C} \\ \hat{x}_i & i \in \mathcal{D} \\ y_i - \alpha_j & i \in A_j,\, 0 \leq j \leq p \end{cases}$$

We now show that $\tilde{x} \in \tilde{P}(\hat{\epsilon}_1)$. Since $\alpha_j \leq 0$ for all $0 \leq j \leq p$ (see lemma 5.2.11) we clearly have $\tilde{x}_i \geq 0$ for all $i \in V(G)$. Then using lemma 5.2.11 the only constraints that remain to be checked are

$$\begin{aligned} &(1) \quad \tilde{x}(V(G)) = \Delta \\ &(2) \quad x(e) \geq y(e) \qquad \text{for all } e \in E(G) \backslash \bigcup_{D \in \mathcal{D}} E(D) \end{aligned}$$

To prove the first item let $M \in \mathcal{M}^*$ be an $\hat{x}$-tight matching and decompose it as

$$M = M_{\mathcal{C}} \cup M_{A,\mathcal{D}} \cup M_{\mathcal{D}}$$

Since $M_{\mathcal{D}}$ is also $\hat{x}$-tight by lemma 5.2.11 we conclude that

$$\hat{x}\left(M_{\mathcal{C}} \cup M_{A,\mathcal{D}}\right) = y\left(M_{\mathcal{C}} \cup M_{A,\mathcal{D}}\right) = \tilde{x}\left(M_{\mathcal{C}} \cup M_{A,\mathcal{D}}\right)$$

where the last equation follows by definition of $\tilde{x}$. But then we also $\tilde{x}_i = \hat{x}_i$ for all $i \in \mathcal{D}$ thus we must have

$$\tilde{x}(V(G)) = \hat{x}(V(G)) = \Delta$$

Finally to prove the second item let $e \in E(G) \setminus \bigcup_{D \in \mathcal{D}} E(D)$. If e has both endpoints in $A \cup \mathcal{C}$ then the claim follows from the fact that $\tilde{x}_i \geq y_i$ for all $i \in A \cup \mathcal{C}$. Hence assume that e is an edge between $\mathcal{D}$ and A. If e joins $\mathcal{D}_i$ to A_i then by definition $\tilde{x}(e) = y(e)$. Hence assume that e joins $\mathcal{D}_i$ to A_j for some $j \neq i$. Suppose that $\tilde{x}(e) < y(e)$. Then it must be the case that $\alpha_i < \alpha_j$. Now since $\hat{x}$ is flexible there exists an $\hat{x}$-tight matching $M \in \mathcal{M}^*$ exposing a vertex of D. Since D is factor critical and $\hat{x} - y$ is constant on D we may assume wlog that M exposes the endpoint of e. Since M is $\hat{x}$-tight lemma 5.2.16 implies that the endpoint of e in A_j is matched into $\mathcal{D}_j$ by some edge $f \in M$. But then our assumption that $\alpha_i < \alpha_j$ means that the matching $M \setminus f \cup e$ has excess strictly less then M, a contradiction. Hence $\tilde{x} \in \tilde{P}(\hat{\epsilon}_1)$ and the proof is complete. □

Lemma 5.2.18. $P(\epsilon_1) \subseteq \tilde{P}(\epsilon_1)$

Proof. Let $x \in P(\epsilon_1)$. Then $x \in \hat{P}(\epsilon_1)$ and by lemma 5.2.17 this means that $x \in \hat{P}(\hat{\epsilon}_1)$. Then by lemma 5.2.11 x satisfies all of the constraints of $\tilde{P}(\epsilon_1)$ except possible $x(e) \geq y(e)$ for $e \in E(G) \setminus \cup_{D \in \mathcal{D}} E(D)$. Thus let $e \in E(G) \setminus \cup_{D \in \mathcal{D}} E(D)$. Then there exists a matching $M \in \mathcal{M}^*_{\mathcal{D}}$ such that $M \cup e$ is a valid matching of G. By lemma 5.2.11 M is x-tight. Since $x \in P(\epsilon_1)$ we must have $x(M \cup e) \geq y(M \cup e) + \epsilon_1$. Thus $x(e) \geq y(e)$. □

5.3 The least core: a stronger result

In this section we use Theorem 5.2.3 to prove a stronger version of Theorem 5.1.4. Hence we assume that (G, w) is not node-weighted and $\nu(G, w) < \nu_f(G, w)$. As before, we fix an arbitrary optimal solution y to the linear program D_{FM} and let $G_=$ be the subgraph of G induced by the edges that are tight with respect to y. For a given Δ, recall the definition of leastcore$(G_=, y, \Delta)$ in 5.2.1 as the set $P(G_=, y, \Delta)(\epsilon_1(G_=, y, \Delta))$. We also recall the definition of the linear program $\tilde{P}(G_=, y, \Delta)$ in 5.2.2 together with its optimum value $\tilde{\epsilon}_1(G_=, y, \Delta)$.

Since y is an optimal solution, once again we have $\nu_f(G_=, y) = \nu_f(G, w) = y(V(G_=))$ and therefore our assumption that $\nu(G, w) < \nu_f(G, w)$ also implies that $\nu(G_=, y) < \nu_f(G_=, y)$. Hence we can apply Theorem 5.2.3 to $(G_=, y)$ to obtain the following corollary.

Corollary 5.3.1. *Let (G, w) be a weighted graph with $\nu(G, w) < \nu_f(G, w)$. Let y be an optimal solution to D_{FM} for (G, w) and let $G_=$ be the subgraph of G induced by the set of edges $\{e \in E(G) : y(e) = w(e)\}$. Suppose that $\nu(G_=, y) \leq \Delta < \nu_f(G_=, y)$. Then*

(i) $\epsilon_1(G_=, y, \Delta) = \tilde{\epsilon}_1(G_=, y, \Delta)$ *and*

(ii) $P(G_=, y, \Delta)(\epsilon_1(G_=, y, \Delta))) = \tilde{P}(G_=, y, \Delta)(\tilde{\epsilon}_1(G_=, y, \Delta)))$.

We let $\mathcal{D}(G_=)$ denote the set of odd components in the Edmonds-Gallai decomposition of $G_=$ and as before M_1 is an arbitrary maximum cardinality matching on the union of the odd components. Recall that a chord is an edge $e \in E(G) \backslash E(G_=)$ such that $e \in \gamma(D)$ for some odd component $D \in \mathcal{D}(G_=)$. We adapt the definition of an extendable allocation to the generalised least core.

Definition 5.3.2. *An allocation $x \in$ leastcore$(G_=, y, \Delta)$ is* extendable *if for every edge $e \in E(G) \backslash E(G_=)$ that is not a chord we have*

$$\sum_{i \in V(G_=): e \in \delta(i)} x_i \geq w(e)$$

We can now present the main result of this section.

Theorem 5.3.3. *Let (G, w) be a weighted graph with $\nu(G, w) < \nu_f(G, w)$. Let y be an optimal solution to D_{FM} for (G, w) and let $G_=$ be the subgraph of G induced by the set of edges $\{e \in E(G) : y(e) = w(e)\}$. Suppose that leastcore$(G_=, y, \Delta)$ contains an extendable allocation for some $\nu(G_=, y) \leq \Delta < y(V(G_=))$. Then*

$$\textit{leastcore}(G, w, \Delta) = \tilde{P}(\tilde{\epsilon}_1)$$

where $\tilde{P}$ *is the linear program*

$$
\begin{array}{llll}
(\tilde{P}) & max & \epsilon & \\
& s.t. & x(e) \geq w(e) & \text{for all } e \in E(G) \setminus \left(\bigcup_{D \in \mathcal{D}(G_=)} \gamma(D) \right) \\
& & x_i \leq y_i & \text{for all } i \in D,\ D \in \mathcal{D}(G_=) \\
& & x_i - y_i = x_j - y_j & \text{for all } i, j \in D,\ D \in \mathcal{D}(G_=) \\
& & x(M_1) \geq y(M_1) + \epsilon & \\
& & x(V(G)) = \Delta & \\
& & x \geq 0 &
\end{array}
$$

and $\tilde{\epsilon}_1$ *is its optimum value.*

The proof of this theorem is essentially identical to the one of Theorem 5.1.4 except that we replace $\nu(G, w)$ by Δ and instead of using corollary 5.1.1 of Theorem 5.0.9 we now use the more general corollary 5.3.1 of Theorem 5.2.3 that allows us to replace $\nu(G_=, y)$ by Δ as well.

Proof. Recall from definition 5.2.1 that leastcore$(G, w, \Delta) = P(G, w, \Delta)(\epsilon_1(G, w, \Delta))$. For ease of notation we denote the linear program $P(G, w, \Delta)$ by P and its optimum value $\epsilon_1(G, w, \Delta)$ by ϵ_1. We then have to show that $\epsilon_1 = \tilde{\epsilon}_1$ and $P(\epsilon_1) = \tilde{P}(\epsilon_1)$.

We start by showing that $\tilde{P}(\epsilon) \subseteq P(\epsilon)$ for any ϵ. Hence let $x \in \tilde{P}(\epsilon)$ for some ϵ. Then the constraints $x(V(G)) = \Delta$ and $x \geq 0$ are clearly satisfied since they are present in the description of $\tilde{P}$, so remains to check that $x(M) \geq w(M) + \epsilon$ for any fixed matching $M \in \mathcal{M}(G)$. It follows from the definition of $\tilde{P}$ that the only edges of G that can have negative excess with respect to x are the ones in $\gamma(D)$ for odd components $D \in \mathcal{D}$. Thus we can lower bound the excess of the matching M by considering only these edges. We obtain

$$x(M) - w(M) \geq \sum_{D \in \mathcal{D}} \sum_{e \in M \cap \gamma(D)} x(e) - w(e)$$

Using the fact that y is a feasible solution to D_{FM} we have

$$\sum_{e \in M \cap \gamma(D)} x(e) - w(e) \geq \sum_{e \in M \cap \gamma(D)} x(e) - y(e) = \sum_{i \in D \cap N(M)} x_i - y_i$$

where $N(M)$ denotes the vertices in D that are covered by M.

Now for a given $D \in \mathcal{D}$ it follows again from the definition of $\tilde{P}$ that $x_i - y_i$ is the same for all $i \in D$. Moreover, any matching contained in $\gamma(D)$ can cover at most $|D| - 1$ vertices of D, since $|D|$ is odd. Hence we can lower bound the excess of M by the excess of M_1 and obtain

$$x(M) - w(M) \geq x(M_1) - y(M_1) \geq \epsilon.$$

This shows that $x \in P(\epsilon)$, and therefore we have shown that

$$\tilde{P}(\epsilon) \subseteq P(\epsilon), \tag{5.8}$$

as well as

$$\tilde{\epsilon}_1 \leq \epsilon_1. \tag{5.9}$$

We will now show the reverse direction, that $\epsilon_1 \leq \tilde{\epsilon}_1$. We first define the following relaxation of P

$$\begin{aligned} (\hat{P}) \quad & \max \epsilon \\ & \text{s.t. } x(M) \geq y(M) + \epsilon \quad \text{for all } M \in \mathcal{M}(G_=) \\ & \qquad x(V(G)) = \Delta \\ & \qquad x \geq 0, \end{aligned}$$

with optimum value $\hat{\epsilon}_1$. Since the constraints of $\hat{P}$ are a subset of the constraints of P we immediately have

$$\epsilon_1 \leq \hat{\epsilon}_1. \tag{5.10}$$

Note that for any vertex $i \in V(G) \backslash V(G_=)$, there is no matching appearing in the constraints of $\hat{P}$ that covers i. It is then easy to see that for any optimal solution $x \in \hat{P}(\hat{\epsilon}_1)$ we must have $x_i = 0$ for all $i \in V(G) \backslash V(G_=)$ and therefore

$$x(V(G)) = x(V(G_=)) = \Delta \quad \text{for all } x \in \hat{P}(\hat{\epsilon}_1).$$

Hence $\hat{P}(\hat{\epsilon}_1)$ is the set of vectors $x \in \mathbb{R}^{|V(G)|}$ that are optimal solutions to the linear program

$$\begin{aligned} &\max\ \epsilon \\ &\text{s.t. } x(M) \geq y(M) + \epsilon \quad \text{for all } M \in \mathcal{M}(G_=) \\ &\qquad x(V(G_=)) = \Delta \\ &\qquad x \geq 0 \end{aligned}$$

Hence it follows that

$$\hat{P}(\hat{\epsilon}_1) = \left\{x \in \mathbb{R}^{|V(G)|} : x_{V(G_=)} \in \text{leastcore}(G_=, y, \Delta) \text{ and } x_i = 0 \text{ for all } i \in V(G)\backslash V(G_=)\right\}$$

where $x_{V(G_=)}$ is the projection of x onto the set $V(G_=)$. Now let x' be an extendable allocation in leastcore$(G_=, y, \Delta)$ (guaranteed to exist by the hypothesis of the theorem) and define the vector x as follows

$$x_i = \begin{cases} x'_i & i \in V(G_=) \\ 0 & i \in V(G)\backslash V(G_=) \end{cases}$$

Then using corollary 5.3.1 and the fact that x' is extendable we obtain that x must satisfy all of the constraints of $\tilde{P}$ and therefore $x \in \tilde{P}(\hat{\epsilon}_1)$. This shows that

$$\hat{\epsilon}_1 \leq \tilde{\epsilon}_1. \tag{5.11}$$

From equations (5.9), (5.10), and (5.11) we obtain $\epsilon_1 = \hat{\epsilon}_1 = \tilde{\epsilon}_1$. This together with (5.8) implies that $\tilde{P}(\tilde{\epsilon}_1) \subseteq P(\epsilon_1)$.

Hence the proof is complete if we show that $P(\epsilon_1) \subseteq \tilde{P}(\epsilon_1)$. Let $x \in P(\epsilon_1)$. Then $x \in \hat{P}(\epsilon_1)$ and letting x' denote the restriction of x to the vertices in $V(G_=)$ we have $x' \in$ leastcore$(G_=, y, \Delta)$. Hence using corollary 5.3.1 again we obtain that x' (and therefore x) satisfies all of the constraints of $\tilde{P}$ except possibly the constraints $x(e) \geq w(e)$ for edges $e \in E(G)\backslash E(G_=)$ that are not chords. Let e be such an edge. Since e is not a chord it does not have both endpoints in the same odd component in $\mathcal{D}$. Since each component in $\mathcal{D}$ is factor critical there must exist a matching M that is a maximum cardinality matching on the union of the odd components $\mathcal{D}$ and such that $M \cup \{e\}$ is a valid matching of G.

From corollary 5.3.1 we know that M must have minimum excess with respect to x' and therefore also with respect to x. Hence

$$x(M \cup e) - w(M \cup e) = (x(M) - y(M)) + (x(e) - w(e)) = \epsilon_1 + (x(e) - w(e))$$

and since $x \in P(\epsilon_1)$ this implies that $x(e) \geq w(e)$. Thus $x \in \tilde{P}(\epsilon_1)$ and the proof is complete. □

Applying Theorem 5.3.3 for $\Delta = \nu(G, w)$ we obtain a direct strengthening of Theorem 5.1.4, where the first condition that $\nu(G_=, y) = \nu(G, w)$ is no longer needed. That is, the least core of (G, w) is captured by the polynomial sized linear program $\tilde{P}$ as long as leastcore$(G_=, y, \nu(G, w))$ contains an extendable allocation. To see that this is a valid choice of Δ, note that $\nu(G_=, y) \leq \nu(G, w)$ and from complementary slackness $\nu_f(G, w) = \nu_f(G_=, y)$. Thus our original assumption that $\nu(G, w) < \nu_f(G, w)$ also implies that $\nu(G, w) < \nu_f(G_=, y)$. The result is listed below.

Corollary 5.3.4. *Let (G, w) be a weighted graph with $\nu(G, w) < \nu_f(G, w)$. Let y be an optimal solution to D_{FM} for (G, w) and let $G_=$ be the subgraph of G induced by the set of edges $\{e \in E(G) : y(e) = w(e)\}$. Suppose that leastcore$(G_=, y, \nu(G, w))$ contain an extendable allocation. Then*

$$\textit{leastcore}(G, w) = \tilde{P}(\tilde{\epsilon}_1)$$

where $\tilde{P}$ is the linear program

$$\begin{aligned}
(\tilde{P}) \quad & \textit{max} \quad \epsilon \\
& \textit{s.t.} \quad x(e) \geq w(e) && \textit{for all } e \in E(G) \setminus \left(\bigcup_{D \in \mathcal{D}(G_=)} \gamma(D) \right) \\
& \qquad x_i \leq y_i && \textit{for all } i \in D,\ D \in \mathcal{D}(G_=) \\
& \qquad x_i - y_i = x_j - y_j && \textit{for all } i, j \in D,\ D \in \mathcal{D}(G_=) \\
& \qquad x(M_1) \geq y(M_1) + \epsilon \\
& \qquad x(V(G)) = \Delta \\
& \qquad x \geq 0
\end{aligned}$$

and $\tilde{e}_1$ is its optimum value.

5.4 Choice of optimal dual solution

The hypothesis of corollary 5.3.4 requires the existence of an extendable allocation in the set leastcore$(G_=, y, \nu(G,w))$. Since this set is clearly dependent on our choice of optimal dual solution y to D_{FM} a natural question that arises is whether one has to check this condition for every single possible choice of y. In this section we show that this is not the case by proving that whether (G,w) satisfies the hypothesis is independent of our choice of optimal dual vector y.

Let (G,w) be a weighted graph with $\nu(G,w) < \nu_f(G,w)$ and y an optimal dual solution to D_{FM}. Let $G_=$ be the subgraph of G induced by the set of edges that are tight with respect to y, and let $\mathcal{D}$ be the set of odd components in the Edmonds-Gallai decomposition of $G_=$. We say that a maximum cardinality matching of $G_=$ exposes an odd component $D \in \mathcal{D}$ if M covers exactly $|D|-1$ vertices of D. We start with the following lemma.

Lemma 5.4.1. *For every $D \in \mathcal{D}$ with $|D| > 1$ there exists a maximum cardinality matching M^* of $G_=$ that exposes D, and in addition satisfies the following property:*

- *for all singleton components $\{j\} \in \mathcal{D}$ that are exposed by M^* we have $y'(j) = 0$ in any optimal dual solution y' to D_{FM}.*

Proof. Let x be a basic fractional matching of G. Then it follows from complementary slackness that the support of x belongs to $E(G_=)$. From Theorem 2.1.4 we have $x_e \in \left\{0, \frac{1}{2}, 1\right\}$ for all $e \in E(G)$ and the edges $\left\{e \in E(G) : x_e = \frac{1}{2}\right\}$ induce vertex disjoint odd cycles. Let $S \subseteq \mathcal{D}$ be the set of singleton odd components that are covered by x. Then by complementary slackness any singleton component $\{j\} \in \mathcal{D} \backslash S$ must have $y'(j) = 0$ in any optimal dual solution y' to D_{FM}. Since x is basic it must induce an integer matching M on $G_=[S \cup A]$ that covers all vertices of S. Let $A_S \subseteq A$ denote the set of vertices in A covered by M.

Now let M^* be any maximum cardinality matching of $G_=$ that exposes D. Suppose that M^* exposes some vertex $i \in S$ and let f be the edge in M covering i. Let P be the unique maximal alternating path in $M \cup M^*$ that starts with edge f. Note that P cannot end with an edge of M since this would imply the existence of a vertex in A_S that is exposed by M^*, but since M^* is a maximum cardinality matching it must cover all vertices of A, and therefore A_S. Hence P must end with an edge of M^*. Then the endpoint of P cannot be in S since all vertices in S are covered by M. Hence P ends in a vertex of an odd component $D' \in \mathcal{D} \backslash S$. Now reversing the edges of M^* along P results in a maximum cardinality matching of $G_=$ that covers i. Since this process did not expose any vertices of S that were already covered by M^* and kept D exposed, we can repeat the argument until M^* covers every vertex of S. □

The following key lemma shows that the vertices belonging to non-singleton odd components must receive the same value in any optimal dual solution.

Lemma 5.4.2. *Let $D \in \mathcal{D}$ with $|D| > 1$. Then for all $i \in D$ we have $y'(i) = y(i)$ for any other optimal solutions y' to D_{FM}.*

Proof. Let $D \in \mathcal{D}$ with $|D| > 1$ and let M^* be a maximum cardinality matching of $G_=$ that exposes D and satisfies the property of lemma 5.4.1. We show that for any vertex $i \in D$ we can use this maximum cardinality matching M^* to obtain a maximum weight fractional matching x whose support contains an odd cycle passing through i. To do this, we first need to recall a result of Cornuéjols and Pulleyblank 2.1.5 that says that if H is a factor critical graph with $|H| > 1$ then for any vertex i of H there exits a perfect fractional matching on H whose support consists of disjoint edges and a single odd cycle containing vertex i.

We obtain a fractional matching x of $G_=$ from M^* by making the following two changes:

- replace M^* inside D by a perfect fractional matching whose support contains an odd cycle passing through vertex i
- for any other odd component $D' \neq D$ with $|D'| > 1$ that is exposed by M^* replace M^* inside D' by any perfect fractional matching

Then it follows by choice of M^* that x covers every vertex $j \in V(G_=)$ with $y_j > 0$. Since all edges of $G_=$ are tight we obtain

$$\sum_{e \in E(G_=)} x_e w_e = y(V(G_=)) = \nu_f(G, w)$$

Hence x is a maximum weight fractional matching of (G, w) meaning that it is an optimal solution to P_{FM}. Then by complementary slackness all edges in the support of x, in particular all edges of the odd cycle containing vertex i, are tight with respect to any optimal dual solution to D_{FM}. This fixes the values of every vertex on the odd cycle including vertex i. Repeating this argument for every vertex of D then completes the proof of the lemma. □

We now look at what happens when we have two distinct optimal solutions to D_{FM}, y_1 and y_2. For $i \in \{1, 2\}$ we define G_i to be the subgraph of G induced by the set of edges $E_i := \{e \in E(G) : y_i(e) = w(e)\}$ and let $\mathcal{D}_i$ denote the odd components in the Edmonds-Gallai decomposition of G_i. Our goal will be to show that leastcore$(G_1, y_1, \nu(G, w))$ contains an extendable allocation if and only if leastcore$(G_2, y_2, \nu(G, w))$ contains an extendable allocation. We start with the following lemma.

Lemma 5.4.3. *$V(G_1) = V(G_2)$.*

Proof. Suppose that $i \in V(G_1)$. To show that $i \in V(G_2)$ it suffices to find a basic solution x to P_{FM} whose support contains an edge incident with i since then by complementary slackness this edge must be tight with respect to any optimal dual solution including y_2. Let x be any basic solution to P_{FM} and assume that x does not cover i, since otherwise there is nothing to show. Let $M := \{e \in E(G) : x(e) = 1\}$. By complementary slackness we have $M \subseteq E_1$. Let f be any edge in $E_1 \setminus M$ that covers i. Now let P be the unique maximal M-alternating path in E_1 that starts with edge f.

We first consider the case where P has odd length. Then P must end in a vertex j exposed by M. Since x is basic, this means that either x does not cover j or the support of x contains an odd cycle that includes vertex j. Suppose first that x does not cover j. Then by complementary slackness we must have $y_1(j) = 0$. Since all edges of P are in E_1, they are all tight with respect to y_1, hence we can flip the edges of M along P to obtain a fractional matching that covers i and has the same weight as x. Suppose now that j belongs to an odd cycle C in the support of x. Let M_j be the maximum cardinality matching on C that exposes j. We can then replace C by M_j and flip the edges of M along P to obtain a fractional matching that covers i and has the same weight as x.

It remains to consider the case where P has even length. Then P ends in a vertex j that has degree one in G_1. If $y_1(j) > 0$ then we modify y_1 slightly by decreasing y_1 on j and increasing y_1 by the same amount on the unique neighbour of j in G_1 until either $y_1(j)$ becomes zero or some new edge becomes tight and joins G_1, in which case we can extend our path P. Note that this maintains the property that y_1 is an optimal dual solution to D_{FM}. Hence we may assume wlog that if P has even length then it ends in a vertex j with $y_1(j) = 0$. Then we can once again flip the edges of M along P to obtain a fractional matching that covers i and has the same weight as x. This completes the proof of the lemma. □

Our next lemma establishes an important connection between the odd components of G_1 and G_2.

Lemma 5.4.4. *Two vertices i and j belong to the same odd component in $\mathcal{D}_1$ if and only if they belong to the same odd component in $\mathcal{D}_2$.*

Proof. Let $D \in \mathcal{D}_1$ with $|D| > 1$ and let $i \in D$. We first show how to construct a maximum cardinality matching of G_2 that exposes i, thus proving that i must also belong to an odd component in $\mathcal{D}_2$. Let M_2 be a maximum cardinality matching of G_2 that exposes some arbitrary non-singleton component of $\mathcal{D}_2$ and satisfies the property of lemma 5.4.1. If M_2 exposes i we are done, hence assume that M_2 covers i. Now let M_1 be a maximum cardinality matching of G_1 that exposes D and satisfies the property of lemma 5.4.1. Since D is factor critical we may assume wlog that i is the vertex of D that is exposed by M_1.

Let $P \subseteq M_1 \cup M_2$ be the unique maximal alternating path starting at i. Then P must end at some vertex j that is exposed by either M_1 or M_2. From lemma 5.4.1 we can conclude that either $y_1(j) = y_2(j) = 0$ or j belongs to a non-singleton component of $\mathcal{D}_1$ or $\mathcal{D}_2$ (depending on if it is exposed by M_1 or M_2). But then we can apply lemma 5.4.2 to obtain that $y_1(j) = y_2(j)$. Note that from the same lemma 5.4.2 we also have $y_1(i) = y_2(i)$ since i belongs to a non-singleton components of $\mathcal{D}_1$.

Hence we found an alternating path P whose endpoints have the same value in y_1 and y_2. In addition for every odd edge e of P we have

$$y_1(e) \geq y_2(e) = w(e)$$

since $e \in M_2 \subseteq E_2$ and y_1 is feasible solution. Similarly for every even edge e of P we have

$$y_2(e) \geq y_1(e) = w(e)$$

since $e \in M_1 \subseteq E_1$ and y_2 is feasible solution. Then a simple inductive argument shows that we must have $y_1(k) = y_2(k)$ for all vertices k of P, meaning that $P \subseteq E_1 \cap E_2$. But then we can reverse the edges of M_2 along P to obtain another maximum cardinality matching of G_2 that exposes i. This shows that all vertices of D belong to odd components in $\mathcal{D}_2$. Now by lemma 5.4.2 we know that $y_1(i) = y_2(i)$ for all $i \in D$. Hence $E(D) \in E_1 \cap E_2$ and thus all vertices of D must belong to the same odd component in $\mathcal{D}_2$. □

We then have the following lemma.

Lemma 5.4.5. *An edge e is a chord of G_1 if and only if e is a chord of G_2.*

Proof. Let $e = (i, j)$ be a chord of G_1. Then since $e \notin E(G_1)$ we must have $y_1(e) > w(e)$. From lemma 5.4.2 we have $y_1(e) = y_2(e)$ and therefore we must have $y_2(e) > w(e)$ and $e \notin E(G_2)$. Moreover, from lemma 5.4.4 we know that i and j must belong to the same odd component in $\mathcal{D}_2$ as well. Hence e is a chord of G_2. □

We can now prove our main result.

Theorem 5.4.6. *Let (G, w) be a weighted graph with $\nu(G, w) < \nu_f(G, w)$. Let y_1 and y_2 be two distinct optimal solutions to D_{FM} for (G, w) and for each $i \in \{1, 2\}$ let G_i be the subgraph of G induced by the edges $\{e \in E(G) : y_i(e) = w(e)\}$. Then leastcore$(G_1, y_1, \nu(G, w))$ contains an extendable allocation if and only if leastcore$(G_2, y_2, \nu(G, w))$ contains an extendable allocation.*

Proof. For $i \in \{1, 2\}$ let ϵ_i be the minimum excess of an allocation in leastcore$(G_i, y_i, \nu(G, w))$. Suppose that x is an extendable allocation in leastcore$(G_1, y_1, \nu(G, w))$. Recall from lemma 5.4.3 that $V(G_1) = V(G_2)$. Now take any matching $M \subseteq E_2$ of G_2. We can decompose the excess of M as follows

$$x(M) - w(M) = x(M \cap E_1) - w(M \cap E_1) + x(M \backslash E_1) - w(M \backslash E_1)$$

Consider an edge $e \in M \backslash E_1$. Since $M \subseteq E_2$ we have $e \in E_2$ and therefore e cannot be a chord of G_2. Then by lemma 5.4.5 e cannot be a chord of G_1 either. Since x is extendable with respect to G_1 and $V(G_1) = V(G_2)$ from lemma 5.4.3 we have

$$\sum_{i \in V(G_2): e \in \delta(i)} x(i) \geq w(e)$$

from which we obtain

$$x(M \backslash E_1) \geq w(M \backslash E_1)$$

and therefore

$$x(M) - w(M) \geq x(M \cap E_1) - w(M \cap E_1) \geq \epsilon_1.$$

Hence (x, ϵ_1) is feasible for the linear program defining $\text{leastcore}(G_2, y_2, \nu(G, w))$. Thus $\epsilon_2 \leq \epsilon_1$, and by a symmetric argument we can show that $\epsilon_1 \leq \epsilon_2$. Hence we must have $\epsilon_1 = \epsilon_2$ and therefore $x \in \text{leastcore}(G_2, y_2, \nu(G, w))$.

It remains to show that x is extendable with respect to G_2. Hence let e be an edge in $E(G) \backslash E_2$ that is not a chord of G_2. Then by lemma 5.4.5 e is not a chord of G_1 either. Hence the fact that x is extendable with respect to G_1 and $V(G_1) = V(G_2)$ implies that

$$\sum_{i \in V(G_2): e \in \delta(i)} x(i) \geq w(e)$$

so x is extendable with respect to G_2. □

5.5 Computing the nucleolus

In this section we show how the polynomial sized characterization of the least core given in Theorem 5.2.3 leads to a polynomial time algorithm for computing the nucleolus. Hence let (G, w) be a weighted graph satisfying the hypothesis of Theorem 5.2.3 so that the least core of (G, w) is given by

$$\text{leastcore}(G, w) = \tilde{P}_1(\tilde{\epsilon}_1)$$

where $\tilde{P}_1$ is the linear program

$$
\begin{aligned}
(\tilde{P}_1) \quad \max \quad & \epsilon \\
\text{s.t.} \quad & x(e) \geq w(e) && \text{for all } e \in E(G) \setminus \left(\bigcup_{D \in \mathcal{D}} \gamma(D) \right) \\
& x_i \leq y_i && \text{for all } i \in D,\ D \in \mathcal{D} \\
& x_i - y_i = x_j - y_j && \text{for all } i, j \in D,\ D \in \mathcal{D} \\
& x(M_1) \geq y(M_1) + \epsilon \\
& x(V(G)) = \nu(G, w) \\
& x \geq 0
\end{aligned}
$$

and $\tilde{e}_1$ is its optimum value.

Recall that y is an optimal solution to D_{FM} and $\mathcal{D}$ is the set of odd components in the Edmonds-Gallai decomposition of $G_=$, the subgraph of G induced by the tight edges with respect to y. Also M_1 is a fixed maximum cardinality matching on the union of the odd components in $\mathcal{D}$. Hence

$$
\begin{aligned}
& y(e) = w(e) \text{ for all } e \in E(D), D \in \mathcal{D} \\
& y(e) \geq w(e) \text{ for all } e \in E(G)
\end{aligned}
$$

For the remainder of this section we use the notation $N = V(G)$ and $\nu = \nu(G, w)$.

The contents of this section are very similar to [68] where it is shown that the polynomial characterization of the least core of node-weighted matching games given in Theorem 5.0.9 can be used to obtain a polynomial time algorithm for computing the nucleolus. The idea is to prove by induction that each subsequent linear program in Maschler's scheme can also be expressed with a polynomial number of constraints.

Recall from section 2.3.3 that the nucleolus can be obtained as the solution to the following sequence of linear programs:

$$
\begin{aligned}
(P_1) \quad \max \quad & \epsilon \\
\text{s.t.} \quad & x(S) \geq \nu(S) + \epsilon \quad \text{for all } S \notin \{\emptyset, N\} \\
& x(N) = \nu(N)
\end{aligned}
$$

with optimum value ϵ_1. If $P_1(\epsilon_1)$ does not consist of a single point we proceed by defining Fix $P_1(\epsilon_1)$ to be the set of coalitions that are fixed by $P_1(\epsilon_1)$

$$\text{Fix } P_1(\epsilon_1) := \{S \subseteq N : x(S) = x'(S) \text{ for all } x, x' \in P_1(\epsilon_1)\}$$

And we solve the second linear program

$$\begin{array}{lll} (P_2) & \max & \epsilon \\ & \text{s.t.} & x(S) \geq \nu(S) + \epsilon \quad \text{for all } S \notin \text{Fix } P_1(\epsilon_1) \\ & & x \in P_1(\epsilon_1) \end{array}$$

defining in a similar way ϵ_2 and

$$\text{Fix } P_2(\epsilon_2) := \{S \subseteq N : x(S) = x'(S) \text{ for all } x, x' \in P_2(\epsilon_2)\}$$

We continue in this way until we reach

$$\begin{array}{lll} (P_r) & \max & \epsilon \\ & \text{s.t.} & x(S) \geq \nu(S) \qquad \text{for all } S \notin \text{Fix } P_{r-1}(\epsilon_{r-1}) \\ & & x \in P_{r-1}(\epsilon_{r-1}) \end{array}$$

such that $P_r(\epsilon_r)$ consists of a single point $\eta(N, v)$, the nucleolus of the game.

The problem with this scheme is that each linear program P_i contains an exponential number of constraints. By Theorem 5.2.3 we can replace P_1 by the equivalent $\tilde{P}_1$ containing only a polynomial number of constraints, since they define the same set of optimal solution. We now show that each P_k for $k \geq 2$ can be replaced by an equivalent formulation that contains only a polynomial number of constraints.

For $k \geq 2$ we define the linear program

$$\begin{array}{llll} (\tilde{P}_k) & \max & \epsilon & \\ & \text{s.t.} & x \in \tilde{P}_{k-1}(\tilde{\epsilon}_{k-1}) & \\ & & x(e) \leq w(e) + \epsilon_1 - \epsilon & \text{for all } e \in \bigcup_{D \in \mathcal{D}} E(D),\ e \notin \text{Fix } \tilde{P}_{k-1}(\tilde{\epsilon}_{k-1}) \\ & & x(e) \geq w(e) - \epsilon_1 + \epsilon & \text{for all } e \in E(G) \backslash \left(\bigcup_{D \in \mathcal{D}} \gamma(D) \right),\ e \notin \text{Fix } \tilde{P}_{k-1}(\tilde{\epsilon}_{k-1}) \\ & & x_i \geq -\epsilon_1 + \epsilon & \text{for all } i \in N, i \notin \text{Fix } \tilde{P}_{k-1}(\tilde{\epsilon}_{k-1}) \end{array}$$

with optimal value $\tilde{\epsilon}_k$ and let

$$\text{Fix } \tilde{P}_k(\tilde{\epsilon}_k) := \left\{ S \subseteq N : x(S) = x'(S) \text{ for all } x, x' \in \tilde{P}_k(\tilde{\epsilon}_k) \right\}.$$

The following theorem shows that we can replace P_k by $\tilde{P}_k$.

Theorem 5.5.1. *We have $\epsilon_k = \tilde{\epsilon}_k$ and $P_k(\epsilon_k) = \tilde{P}_k(\tilde{\epsilon}_k)$ for all $k = 1, \cdots, r$.*

Proof. For $k = 1$ the claim follows from Theorem 5.2.3. We proceed by induction on k. Assume that $\epsilon_{k-1} = \tilde{\epsilon}_{k-1}$ and $P_{k-1}(\epsilon_{k-1}) = \tilde{P}_{k-1}(\tilde{\epsilon}_{k-1})$. We prove the induction step using two lemmas.

Lemma 5.5.2. *$P_k(\epsilon) \subseteq \tilde{P}_k(\epsilon)$ and therefore $\epsilon_k \leq \tilde{\epsilon}_k$.*

Proof. Let $x \in P_k(\epsilon)$. Then $x \in P_1(\epsilon_1) = \tilde{P}(\tilde{\epsilon}_1)$.

Let $f \in E(G) \setminus (\cup_{D \in \mathcal{D}} \gamma(D))$ such that $f \notin \text{Fix } \tilde{P}_{k-1}(\tilde{\epsilon}_{k-1})$. Choose a maximum cardinality matching on the odd components $M \in \mathcal{M}^*_{\mathcal{D}}$ such that $M \cup f$ is also matching. Such an M always exists since each component $D \in \mathcal{D}$ is factor critical and e does not have both endpoints in the same component of $\mathcal{D}$. It follows from the definition of $\tilde{P}_1(\tilde{\epsilon}_1)$ that $N(M)$, the set of vertices covered by M, is fixed since all matchings in $\mathcal{M}^*_{\mathcal{D}}$ have minimum excess. Thus $N(M)$ must also by fixed by $\tilde{P}_{k-1}(\tilde{\epsilon}_{k-1})$. Since we assumed that $f \notin \text{Fix } \tilde{P}_{k-1}(\tilde{\epsilon}_{k-1})$ we must have $N(M \cup f) \notin \text{Fix } \tilde{P}_{k-1}(\tilde{\epsilon}_{k-1})$ as well. Thus, by the constraints in P_k we have

$$x(N(M \cup f)) \geq w(M \cup f) + \epsilon$$

and since M is fixed

$$x(N(M)) = w(M) + \tilde{\epsilon}_1$$

and hence $x(f) - w(f) \geq \epsilon - \tilde{\epsilon}_1$, satisfying the constraint in $\tilde{P}_k$.

Next, let $i \in N$ such that $i \notin \text{Fix } \tilde{P}_{k-1}(\tilde{\epsilon}_{k-1})$. The argument is similar. We choose a maximum cardinality matching on the odd components $M \in \mathcal{M}^*_{\mathcal{D}}$. Then, by the constraints in P_k we have

$$x(N(M)) + x_i \geq w(M) + \epsilon$$

and since M is fixed

$$x(N(M)) = w(M) + \tilde{\epsilon}_1$$

and hence $x_i \geq \epsilon - \tilde{\epsilon}_1$ as required.

Finally, consider any edge $f \in \cup_{D \in \mathcal{D}} E(D)$ such that $f \notin \text{Fix } \tilde{P}_{k-1}(\tilde{\epsilon}_{k-1})$. Choose a maximum cardinality matching on the odd components $M \in \mathcal{M}^*_{\mathcal{D}}$ such that M covers both endpoints of f. Then since f is not fixed, neither is $N(M) \backslash N(f)$. Hence, by the constraints of P_k we have

$$x(N(M)) - x(N(f)) \geq w(M) - w(f) + \epsilon$$

and using the fact that M is fixed we obtain $x(f) - w(f) \leq \tilde{\epsilon}_1 - \epsilon$ as required. □

Lemma 5.5.3. *$\tilde{P}_k(\epsilon) \subseteq P_k(\epsilon)$ and therefore $\tilde{\epsilon}_k \leq \epsilon_k$.*

Proof. Let $x \in \tilde{P}_k(\epsilon)$. Again this implies that $x \in P_1(\epsilon_1) = \tilde{P}(\tilde{\epsilon}_1)$. We want to show that

$$x(S) \geq \nu(S) + \epsilon$$

for all $S \subset N$ not yet fixed by $P_{k-1}(\epsilon_{k-1}) = \tilde{P}_{k-1}(\tilde{\epsilon}_{k-1})$. Hence let S be such a set. Since $x \geq 0$ it is sufficient to show the above inequality for $S = N(M)$ where M is a matching of G. Further, since $x(e) - w(e) \geq 0$ for all $e \notin \cup_{D \in \mathcal{D}} \gamma(D)$ for $D \in \mathcal{D}$, we can assume that $M \in \cup_{D \in \mathcal{D}} \gamma(D)$. Since y is a feasible solution to D_{FM} we have

$$x(M) - w(M) \geq x(M) - y(M)$$

And since $x_i - y_i = x_D - y_D$ for all $i \in D$, $D \in \mathcal{D}$ by the constraints of $\tilde{P}_1$ we obtain

$$x(M) - y(M) \geq \sum_{D \in \mathcal{D}} |M \cap \gamma(D)|(x_D - y_D)$$

Letting $k_D = |M \cap \gamma(D)|$ for all $D \in \mathcal{D}$, it then suffices to show that

$$\sum_{D \in \mathcal{D}} k_D(x_D - y_D) \geq \epsilon$$

Choose a matching $M' \subseteq \cup_{D \in \mathcal{D}} E(D)$ such that $|M' \cap E(D)| = k_D$ for all $D \in \mathcal{D}$ and $M' \subseteq M^*$ for some $M^* \in \mathcal{M}^*_{\mathcal{D}}$. Then

$$x(M') - w(M') = \sum_{D \in \mathcal{D}} k_D(x_D - y_D)$$

Note that we can always choose such an M' such that M' is also not fixed. Now since M^* is fixed, M' is not fixed, and $M' \subseteq M^*$ we must have at least one edge $f \in M^* \backslash M'$ that is not fixed. Hence by the constraints of $\tilde{P}$ we have

$$x(f) - w(f) \leq \epsilon_1 - \epsilon$$

Further, for each edge $e \in M^* \backslash (M' \cup f)$ we have $x(e) \leq w(e)$ by the constraints in $\tilde{P}_1$.

$$\begin{aligned} x(M') - w(M') &= (x(M^*) - w(M^*)) - (x(f) - w(f)) - \sum_{e \in M^* \backslash (M' \cup f)} (x(e) - w(e)) \\ &\geq (x(M^*) - w(M^*)) - (x(f) - w(f)) \\ &= \epsilon_1 - (\epsilon_1 - \epsilon) \\ &= \epsilon \end{aligned}$$

as required. □

This completes the proof of the theorem. □

Theorem 5.5.1 shows that we can replace each linear program in Maschler's scheme by an equivalent one containing only a polynomial number of constraints. We thus obtain the following corollary.

Corollary 5.5.4. *Let (G, w) be a weighted graph and (N, v) the corresponding matching game. If (G, w) satisfies the hypothesis of Theorem 5.2.3 then the nucleolus of (N, v) can be computed in polynomial time.*

Chapter 6

Conclusion

The results presented in this thesis open many avenues for future work, some of which we present below.

- Bridging the gap between network bargaining games with general capacities and their cooperative games counterpart, c-matching games.

  We have shown that the equivalences observed in the case of unit capacities do not extend to general capacities. In particular, our results imply that the set of allocations corresponding to balanced solutions form a strict subset of the intersection of the core and prekernel. A natural question is to investigate further properties of this subset. We believe that this subset characterizes allocations with important structural properties, and the study of these allocations can be of interest to the cooperative games community. Since the intersection of the core and prekernel is not always convex, it would be particularly interesting to investigate whether this subset is better behaved. As a further point, the nucleolus is known to lie in the intersection of the core and prekernel, provided that the core is non-empty. However, it is not known whether the nucleolus always corresponds to a balanced outcome. Since both the notion of the nucleolus and balanced outcomes aim to capture fair solutions, showing that there is an equivalence is an important step that would further validate both concepts.

- Investigating further properties of c-matching games.

  While matching games are fairly well studied, very little is known about the more general c-matching variant. Several fundamental questions about c-matching games are still open. As a first question one could investigate the question of whether the nucleolus can be computed efficiently for c-matching games with non-empty core, for example when the underlying graph is bipartite. This setting poses significantly more challenges then that of matching games, and it is evident that there is a need for the development of some new techniques and structural tools for solving these problems. Another interesting avenue would be the exploration of the connection between c-matching games and many-to-many stable matchings. In the case of many-to-many stable matchings, it is known that if the underlying graph is bipartite then stable matchings can be computed efficiently. This indicates that a similar result should be possible for c-matching games on bipartite graphs.

- The existence of cyclic three dimensional stable matchings.

  The problem of determining whether every instance of a cyclic three dimensional matching possess a stable solution is still open. The results of this thesis seem to suggest that the existence an efficient algorithm for computing such solutions is unlikely. An interesting avenue would be to approach the existence question from a non-algorithmic perspective.

- New extensions of network games.

  One can also study the network games proposed in this thesis in a dynamic and decentralized setting, where there is no central authority for enforcing a matching protocol and the matching platform evolves over time. Such a setting can arise in many electronic applications, for example matching markets on the internet that involve a large number of agents interacting in real time. Scenarios such as these call for the design of dynamic algorithms that allow for several externalities, such as peer effects, incomplete information, or side payments. An important question would be to develop such algorithm that converge towards stable outcomes efficiently.

References

[1] David J. Abraham. *Algorithmics of Two-sided Matching Problems*. PhD thesis, University of Glasgow, 2003. 14

[2] Robert J. Aumann and Michael Maschler. Game theoretic analysis of a bankruptcy problem from the talmud. *Journal of Economic Theory*, 36(2):195–213, 1985. 17, 18

[3] Yossi Azar, Benjamin E. Birnbaum, Elisa Celis, Nikhil R. Devanur, and Yuval Peres. Convergence of local dynamics to balanced outcomes in exchange networks. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*, pages 293–302. IEEE Computer Society, 2009. 9

[4] Haris Aziz, Oded Lachish, Mike Paterson, and Rahul Savani. Wiretapping a hidden network. In *Internet and Network Economics, 5th International Workshop, WINE 2009, Rome, Italy, December 14-18, 2009. Proceedings*, volume 5929 of *Lecture Notes in Computer Science*, pages 438–446. Springer, 2009. 21

[5] Michel L. Balinski. On maximum matching, minimum covering and their connections. In *Proceedings of the Princeton Symposium on Mathematical Programming (Princeton Univ., 1967)*, pages 303–311. Princeton Univ. Press, Princeton, N.J., 1970. 25

[6] MohammadHossein Bateni, MohammadTaghi Hajiaghayi, Nicole Immorlica, and Hamid Mahini. The cooperative game theory foundations of network bargaining games. In *Automata, Languages and Programming, 37th International Colloquium, ICALP 2010, Bordeaux, France, July 6-10, 2010, Proceedings, Part I*, pages 67–78, 2010. 7, 8, 43, 52, 54, 62

[7] Michael H. Birnbaun and Ulrich Schmidt. An experimental investigation of violations of transitivity in choice under uncertainty. *Journal of Risk and Uncertainty*, 37(1):77–91, 2008. 11

[8] Péter Biró and Eric McDermid. Three-sided stable matchings with cyclic preferences. *Algorithmica*, 58(1):5–18, 2010. 15, 49

[9] Rodica Branzei, Elena Inarra, Stef Tijs, and José Manuel Zarzuelo. A simple algorithm for the nucleolus of airport profit games. *International Journal of Game Theory*, 34(2):259–272, 2006. 21

[10] Elisa Celis, Nikhil R. Devanur, and Yuval Peres. Local dynamics in bargaining networks via random-turn games. In *Internet and Network Economics - 6th International Workshop, WINE 2010, Stanford, CA, USA, December 13-17, 2010. Proceedings*, volume 6484 of *Lecture Notes in Computer Science*, pages 133–144. Springer, 2010. 9

[11] Tanmoy Chakraborty, Stephen Judd, Michael Kearns, and Jinsong Tan. A behavioral study of bargaining in social networks. In *Proceedings 11th ACM Conference on Electronic Commerce (EC-2010), Cambridge, Massachusetts, USA, June 7-11, 2010*, pages 243–252. ACM, 2010. 5

[12] Georgios Chalkiadakis, Edith Elkind, and Michael Wooldridge. *Computational Aspects of Cooperative Game Theory.* Synthesis Lectures on Artificial Intelligence and Machine Learning, 16. Morgan & Claypool Publishers, Williston, VT, 2012. 21

[13] Ning Chen, Pinyan Lu, and Hongyang Zhang. Computing the nucleolus of matching, cover and clique games. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada.* AAAI Press, 2012. 20

[14] Karen S. Cook, Richard M. Emerson, Mary R. Gillmore, and Toshio Yamagishi. Distribution of power in exchange networks: Theory and experimental results. *American Journal of Sociology*, pages 275–305, 1983. 4

[15] Karen S. Cook and Toshio Yamagishi. Power in exchange networks: A power-dependence formulation. *Social Networks*, 14(3):245–265, 1992. 2

[16] William J. Cook, William H. Cunningham, William R. Pulleyblank, and Alexander Schrijver. *Combinatorial Optimization*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, Inc., New York, 1998. A Wiley-Interscience Publication. 26

[17] Gérard Cornuéjols and William R. Pulleyblank. Critical graphs, matchings and tours or a hierarchy of relaxations for the travelling salesman problem. *Combinatorica. An International Journal of the János Bolyai Mathematical Society*, 3(1):35–52, 1983. 25

[18] Xiao Deng and Christos H. Papadimitriou. On the complexity of cooperative solution concepts. *Mathematics of Operations Research*, 19(2):257–266, 1994. 20

[19] Xiaotie Deng, Qizhi Fang, and Xiaoxun Sun. Finding nucleolus of flow game. *Journal of Combinatorial Optimization*, 18(1):64–86, 2009. 21

[20] Xiaotie Deng, Toshihide Ibaraki, and Hiroshi Nagamochi. Algorithmic aspects of the core of combinatorial optimization games. *Mathematics of Operations Research*, 24(3):751–766, 1999. 32

[21] Moez Draief and Milan Vojnovic. Bargaining dynamics in exchange networks. *CoRR*, abs/1202.1089, 2012. 9

[22] Jack Edmonds. Maximum matching and a polyhedron with $0,1$-vertices. *Journal of Research of the National Bureau of Standards*, 69B:125–130, 1965. 26

[23] Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics. Journal Canadien de Mathématiques*, 17:449–467, 1965. 27

[24] Jenő Egerváry. On combinatorial properties of matrices. *Matematikai Lapok. Bolyai János Matematikai Társulat*, 38:16–28, 1931. 25

[25] Edith Elkind, Leslie Ann Goldberg, Paul W. Goldberg, and Michael Wooldridge. Computational complexity of weighted threshold games. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada*, pages 718–723. AAAI Press, 2007. 21

[26] Edith Elkind and Dmitrii V. Pasechnik. Computing the nucleolus of weighted voting games. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2009, New York, NY, USA, January 4-6, 2009*, pages 327–335. SIAM, 2009. 21

[27] Kimmo Eriksson, Jonas Sjöstrand, and Pontus Strimling. Three-dimensional stable matching with cyclic preferences. *Mathematical Social Sciences*, 52(1):77–87, 2006. 15

[28] Ulrich Faigle, Walter Kern, Sándor P. Fekete, and Winfried Hochstättler. On the complexity of testing membership in the core of min-cost spanning tree games. *International Journal of Game Theory*, 26(3):361–366, 1997. 21

[29] Ulrich Faigle, Walter Kern, Sándor P. Fekete, and Winfried Hochstättler. The nucleon of cooperative games and an algorithm for matching games. *Mathematical Programming*, 83:195–211, 1998. 20

[30] Ulrich Faigle, Walter Kern, and Jeroen Kuipers. Note computing the nucleolus of min-cost spanning tree games is np-hard - computing the nucleolus of min-cost spanning tree games is np-hard. *International Journal of Game Theory*, 27(3):443–450, 1998. 21

[31] Ulrich Faigle, Walter Kern, and Jeroen Kuipers. On the computation of the nucleolus of a cooperative game. *International Journal of Game Theory*, 30(1):79–98, 2001. 8, 9, 31, 43, 50, 62

[32] Linda Farczadi, Konstantinos Georgiou, and Jochen Könemann. Network bargaining with general capacities. In *Algorithms - ESA 2013 - 21st Annual European Symposium, Sophia Antipolis, France, September 2-4, 2013. Proceedings*, pages 433–444, 2013. iii

[33] Linda Farczadi, Konstantinos Georgiou, and Jochen Könemann. Stable marriage with general preferences - extended abstract. In *Algorithmic Game Theory - 7th International Symposium, SAGT 2014, Haifa, Israel, September 30 - October 2, 2014. Proceedings*, pages 25–36, 2014. iii

[34] Peter Fishburn. Preference structures and their numerical representations. *Theoretical Computer Science*, 217(2):359–383, 1999. ORDAL '96 (Ottawa, ON). 11

[35] Peter C. Fishburn. Nontransitive preferences in decision theory. *Journal of Risk and Uncertainty*, 4(2):113–134, 1991. 11

[36] David Gale and Lloyd S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 120(5):386–391, 2013. 9

[37] David Gale and Marilda Sotomayor. Some remarks on the stable matching problem. *Discrete Applied Mathematics. The Journal of Combinatorial Algorithms, Informatics and Computational Sciences*, 11(3):223–232, 1985. 10, 45

[38] Tibor Gallai. Kritische Graphen. II. *Magyar Tud. Akad. Mat. Kutató Int. Közl.*, 8:373–395 (1964), 1963. 27

[39] Tibor Gallai. Maximale Systeme unabhängiger Kanten. *Magyar Tud. Akad. Mat. Kutató Int. Közl.*, 9:401–413 (1965), 1964. 27

[40] Daniel Granot, Michael Maschler, Guillermo Owen, and Weiping R. Zhu. The kernel/nucleolus of a standard tree game. *International Journal of Game Theory*, 25(2):219–244, 1996. 21

[41] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, second edition, 1993. 35

[42] Dan Gusfield and Robert W. Irving. *The Stable Marriage Problem: Structure and Algorithms*. Foundations of Computing Series. MIT Press, Cambridge, MA, 1989. 9, 10

[43] Magnús M. Halldórsson, Kazuo Iwama, Shuichi Miyazaki, and Hiroki Yanagisawa. Improved approximation results for the stable marriage problem. *ACM Transactions on Algorithms*, 3(3), 2007. 14

[44] Chien-Chung Huang. Circular stable matching and 3-way kidney transplant. *Algorithmica*, 58(1):137–150, 2010. 15, 49

[45] Robert W. Irving. Stable marriage and indifference. *Discrete Applied Mathematics*, 48(3):261–272, 1994. 14

[46] Robert W. Irving, David Manlove, and Sandy Scott. Strong stability in the hospitals/residents problem. In *STACS 2003*, volume 2607 of *Lecture Notes in Comput. Sci.*, pages 439–450. Springer, Berlin, 2003. 15

[47] Kazuo Iwama and Shuichi Miyazaki. A survey of the stable marriage problem and its variants. In *Informatics Education and Research for Knowledge-Circulating Society, 2008. ICKS 2008. International Conference on*, pages 131–136. IEEE, 2008. 47

[48] Kazuo Iwama, Shuichi Miyazaki, and Naoya Yamauchi. A 1.875: approximation algorithm for the stable marriage problem. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7-9, 2007*, pages 288–297. SIAM, 2007. 11

[49] Kazuo Iwama, Shuichi Miyazaki, and Hiroki Yanagisawa. A 25/17-approximation algorithm for the stable marriage problem with one-sided ties. *Algorithmica*, 68(3):758–775, 2014. 11, 14

[50] Steven J.Bram, William V. Gehrlein, and Fred S. Roberts, editors. *The Mathematics of Preference, Choice and Order*. Studies in Choice and Welfare. Springer-Verlag, Berlin, 2009. Essays in honor of Peter C. Fishburn. 11

[51] Yashodhan Kanoria, Mohsen Bayati, Christian Borgs, Jennifer T. Chayes, and Andrea Montanari. A natural dynamics for bargaining on exchange networks. *CoRR*, abs/0911.1767, 2009. 8, 41

[52] Yashodhan Kanoria, Mohsen Bayati, Christian Borgs, Jennifer T. Chayes, and Andrea Montanari. Fast convergence of natural bargaining dynamics in exchange networks. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 1518–1537. SIAM, 2011. 39

[53] Walter Kern and Daniël Paulusma. Matching games: The least core and the nucleolus. *Mathematics of Operations Research*, 28(2):294–308, 2003. 20, 87

[54] Zoltán Király. Linear time local approximation algorithm for maximum stable marriage. *Algorithms*, 6(3):471–484, 2013. 10

[55] Jon M. Kleinberg and Éva Tardos. Balanced outcomes in social exchange networks. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 295–304. ACM, 2008. 8, 42, 43, 69

[56] Donald E. Knuth. *Stable marriage and its relation to other combinatorial problems: An introduction to the mathematical analysis of algorithms*. Amer. Math. Soc., Providence, RI, 1997. 9, 12

[57] Jeroen Kuipers. *A Polynomial Time Algorithm for Computing the Nucleolus of Convex Games*. Maastricht University, 1996. 21

[58] Jean Lemaire. An application of game theory: cost allocation. *Astin Bulletin*, 14(01):61–81, 1984. 16

[59] László Lovász and Michael D. Plummer. *Matching Theory*. AMS Chelsea Publishing, Providence, RI, 2009. Corrected reprint of the 1986 original [MR0859549]. 23

[60] David Manlove. The structure of stable marriage with indifference. *Discrete Applied Mathematics*, 122(1-3):167–181, 2002. 14

[61] David Manlove. *Algorithmics of Matching Under Preferences*, volume 2 of *Series on Theoretical Computer Science*. World Scientific Publishing Co. Pte. Ltd., Singapore, 2013. With a foreword by Kurt Mehlhorn. 9, 11

[62] David Manlove, Robert W. Irving, Kazuo Iwama, Shuichi Miyazaki, and Yasufumi Morita. Hard variants of stable marriage. *Theoretical Computer Science*, 276(1-2):261–279, 2002. 10, 47, 74

[63] Michael Maschler, Bezalel Peleg, and Lloyd S. Shapley. Geometric properties of the kernel, nucleolus, and related solution concepts. *Mathematics of Operations Research*, 4(4):303–338, 1979. 20, 33

[64] Kenneth O. May. Intransitivity, utility, and the aggregation of preference patterns. *Econometrica. Journal of the Econometric Society*, 22:1–13, 1954. 11

[65] Eric McDermid. A 3/2-approximation algorithm for general stable marriage. In *Automata, Languages and Programming, 36th International Colloquium, ICALP 2009, Rhodes, Greece, July 5-12, 2009, Proceedings, Part I*, volume 5555 of *Lecture Notes in Computer Science*, pages 689–700. Springer, 2009. 14

[66] Nimrod Megiddo. Computational complexity of the game theory approach to cost allocation for a tree. *Mathematics of Operations Research*, 3(3):189–196, 1978. 21

[67] John F. Nash. The bargaining problem. *Econometrica. Journal of the Econometric Society*, 18:155–162, 1950. 3

[68] Daniël Paulusma. *Complexity Aspects of Cooperative Games*. ProQuest LLC, Ann Arbor, MI, 2001. Thesis (Dr.)–Universiteit Twente (The Netherlands). 20, 32, 33, 36, 37, 38, 87, 103, 115, 128

[69] Bezalel Pelegand and Peter Sudhölter. *Introduction to the Theory of Cooperative games*, volume 34 of *Theory and Decision Library. Series C: Game Theory, Mathematical Programming and Operations Research*. Springer, Berlin, second edition, 2007. 3

[70] William R. Pulleyblank. Fractional matchings and the Edmonds-Gallai theorem. *Discrete Applied Mathematics. The Journal of Combinatorial Algorithms, Informatics and Computational Sciences*, 16(1):51–58, 1987. 25

[71] Sharon C. Rochford. Symmetrically pairwise-bargained allocations in an assignment market. *Journal of Economic Theory*, 34(2):262–281, 1984. 3, 4

[72] Alvin E. Roth and Marilda Sotomayor. *Two-sided Matching*, volume 18 of *Econometric Society Monographs*. Cambridge University Press, Cambridge, 1990. A study in game-theoretic modeling and analysis, With a foreword by Robert Aumann. 9, 10

[73] David Schmeidler. The nucleolus of a characteristic function game. *SIAM Journal on Applied Mathematics*, 17:1163–1170, 1969. 16, 30

[74] Alexander Schrijver. *Theory of Linear and Integer Programming*. Wiley-Interscience Series in Discrete Mathematics. John Wiley & Sons, Ltd., Chichester, 1986. A Wiley-Interscience Publication. 35

[75] Lloyd S. Shapley and Martin Shubik. The assignment game. I. The core. *International Journal of Game Theory*, 1(2):111–130, 1972. 3, 6

[76] Tamás Solymosi and Tirukkannamangai E. S. Raghavan. An algorithm for finding the nucleolus of assignment games. *International Journal of Game Theory*, 23(2):119–143, 1994. 20

[77] Tamás Solymosi, Tirukkannamangai E. S. Raghavan, and Stef Tijs. Computing the nucleolus of cyclic permutation games. *European Journal of Operational Research*, 162(1):270–280, 2005. 21

[78] Richard E. Stearns. Convergent transfer schemes for N-person games. *Transactions of the American Mathematical Society*, 134:449–459, 1968. 9

[79] John H. Vande Vate. Linear programming brings marital bliss. *Operations Research Letters*, 8(3):147–153, 1989. 46, 47, 77

[80] David Willer. *Network Exchange Theory*. Praeger Publishers, 1999. 5