Stabilization and the Core of Matching Games

Lucy Verberk

Stabilization and the Core of Matching Games

A catalogue record is available from the Eindhoven University of Technology Library.

ISBN: 978-90-386-6473-6

Cover design by Lucy Verberk. The cover is based on a spray painting that Lucy made during an activity with the SPOR cluster at Eindhoven University of Technology.

Printed by ADC Nederland

Stabilization and the Core of Matching Games

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de rector magnificus prof.dr. S.K. Lenaerts, voor een commissie aangewezen door het College voor Promoties, in het openbaar te verdedigen op maandag 13 oktober 2025 om 16:00 uur

door

Lucy Petronella Antonia Verberk

geboren te Sint-Oedenrode

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter:	Prof. dr. M.G.J. van den Brand
1e promotor:	Prof. dr. L. Sanità (Bocconi University)
2e promotor:	Prof. dr. F.C.R. Spieksma
leden:	Prof. dr. M.T. de Berg
	Dr. K. Chandrasekaran (University of Illinois)
	Prof. dr. B. Peis (RWTH Aachen University)
	Prof. dr. G. Schäfer (Centrum Wiskunde & Informatica)
adviseur:	Dr. U. Schmidt-Kraepelin

Het onderzoek dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

Contents

Part I

Introduction

Chapter 1

Prologue

In this dissertation we look at two types of *games*: network bargaining games and cooperative matching games. We introduce these using an example.

Adam and Betty both want to apply for a research grant together with their colleague Charlotte. Charlotte, however, only has enough time for one new research project. Both pairs, that is Adam with Charlotte, and Betty with Charlotte, are eligible for an equally funded research grant. So let us say that both pairs can obtain a funding of 1 unit of money.

We can represent this example situation, this instance, by a graph as shown in Figure 1.1. A graph consists of vertices and edges, drawn as circles and lines in Figure 1.1, respectively. In this case, the vertices represent Adam (A), Betty (B) and Charlotte (C), and the edges represent the pairs that can apply for a research grant. Both edges in Figure 1.1 have a number next to them, called the weight of the edge, that represents the amount of funding the pair can receive.

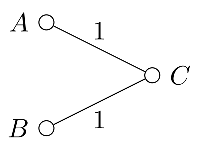


Figure 1.1: An instance.

We first explain network bargaining games using this example. When two persons decide to apply for a grant together they must negotiate how to divide the funding between them, that is, they have to bargain. In our example, Charlotte has a very strong bargaining position, as she has multiple options. Say Adam and Charlotte decide to apply for a research grant together, and they split the funding equally, so both get 0.5. Charlotte can obtain more for herself by suggesting to Betty that they instead apply together and split it such that Charlotte gets 0.6 and Betty gets 0.4. Since Betty currently has nothing, she is happy to accept this offer. But now Adam has nothing, and Charlotte can use this to obtain even more. In particular, Charlotte can suggest to Adam that they apply together and split it such that Charlotte gets 1 and Adam gets 0. Since Adam will not be worse off than he already is, he accepts the offer.

In general, instances can of course be much larger and more complicated. But the idea remains the same: each person wants to form a pair with someone else, and tries to bargain for as much value as possible. So what we are interested in here is the pairs that are formed, and how each pair decides to split the available value. The set of all pairs that are formed is what we call a *matching* in the graph. An outcome to a network bargaining game is described by the pairs that are formed, so a matching, and the value that each person obtains from a pair. We can represent an outcome in a graph, by highlighting the matching, and displaying the value each person obtains from a pair above the corresponding edge. The outcome of the example we discussed before is shown in Figure 1.2. In this figure the bold line indicates that Adam and Charlotte apply for a research grant together, and the values above this edge indicate that Adam gets 0 and Charlotte gets 1.

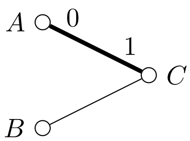


Figure 1.2: An outcome of a network bargaining game.

Now we explain cooperative matching games using the same example, but with a bit more context: Adam, Betty and Charlotte are in a research group, and the leader of their group wants them to cooperate to obtain as much funding as possible for their group. To encourage Adam, Betty and Charlotte to cooperate, the leader of the research group divides the obtained funding over them in such a way that they cannot obtain more funding in smaller groups. In this case, either Adam and Charlotte can apply together and receive a funding of 1, or Betty and Charlotte can apply together and receive a funding of 1, but not both pairs can apply. So as a group they can obtain a funding of at most 1. Next, the leader decides how to divide this funding of 1 over Adam, Betty and Charlotte, by looking at how much funding smaller groups can obtain. Adam and Charlotte can form a smaller group, and obtain a funding of 1. So, if they get less than 1 in the big group, they can obtain more by not cooperating with the big group, and instead forming their own group. To prevent this from happening, Adam and Charlotte together should get at least 1. As there is 1 to divide in total, this means that Betty gets 0. Similarly, Betty and Charlotte can obtain a funding of 1 without Adam, so Adam also gets 0. That leaves Charlotte with the entire funding of 1.

Again, in general, instances can be much larger and more complicated. But the idea remains the same: everyone cooperates to obtain as much funding as possible as a group, and they divide this over the group in such a way that no smaller group can obtain more funding by leaving the big group. What we are interested in here is how much value each person obtains. Even though we are not interested in the specific pairs that are formed here, the total value that the persons can divide over the group still comes from an underlying matching. An outcome to a cooperative matching game is described by the value each person obtains. We can represent an outcome in a graph, by displaying the value each person obtains above the corresponding vertex. The outcome of the example we discussed before is shown in Figure 1.3. In this figure the values above the vertices indicate that Adam and Betty get 0, and Charlotte gets 1.

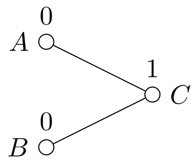


Figure 1.3: An outcome of a cooperative matching game.

At first sight it may seem that the only difference between these games is how we present the outcome. But we will see an example later, where the two games have a different outcome.

Now consider the instance where Adam and Charlotte are eligible for a research grant twice the amount that Betty and Charlotte are eligible for. This new instance is represented in Figure 1.4.

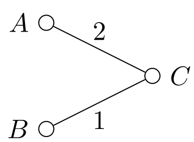


Figure 1.4: An instance.

We first discuss the network bargaining game on this instance. If Adam and Charlotte now decide to apply for a research grant together and split the funding equally, they both get 1. Charlotte still has Betty as an alternative option, but this time that does not make her bargaining position stronger, as she cannot obtain more for herself by switching to Betty. This outcome is represented in Figure 1.5.

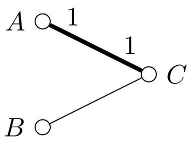


Figure 1.5: An outcome of a network bargaining game.

In the cooperative matching game on this instance, as a group they can obtain a funding of at most 2, by letting Adam and Charlotte apply together. Clearly, the smaller group consisting of Adam and Charlotte can also obtain a funding of 2, which means that Betty gets 0. The smaller group consisting of Betty and Charlotte can obtain a funding of 1. Then Adam can get at most 1: there is 2 to divide in total, minus the 1 that the others can get without him. So a possible outcome, that ensures that all smaller groups stay with the big group, is: Betty gets 0, and Adam and Charlotte both get 1. This outcome is represented in Figure 1.6.

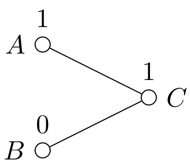


Figure 1.6: An outcome of a cooperative matching game.

All the outcomes we have seen so far are stable: In the network bargaining game, no two persons can both obtain more by deviating from the outcome and instead applying together. In the cooperative matching game, no smaller group can obtain more by leaving the big group. Next, we discuss an instance where, for both games, there are no stable outcomes. We say that such an instance is unstable.

We go back to the instance in Figure 1.1. There, both Adam and Betty got nothing in the final outcome of either game, see Figures 1.2 and 1.3. Hence, they decide that they can also apply for a research grant together instead. They are eligible for a research grant of the same amount as the other pairs. Figure 1.7 represents this new instance.

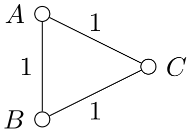


Figure 1.7: An instance.

So, in the network bargaining game, Adam and Betty decide to apply together and split the funding equally, so they both get 0.5. But then the same happens as before: both Adam and Betty have multiple options. Suppose Betty suggest to Charlotte to apply together such that Betty gets 0.75 and Charlotte gets 0.25. Charlotte is happy to accept Betty's offer, as currently she gets 0. However, now that Charlotte gets 0.25 she sees an opportunity to get even more. Charlotte can instead apply with Adam such that she gets 0.6 and Adam gets 0.4. Adam can then instead apply with Betty such that he and Betty both get 0.5. Now we are back where we started. This process is displayed in Figure 1.8. This will continue forever: whatever the current outcome is, there are always two persons that can both obtain more by deviating from the current outcome and instead applying for a research grant together.

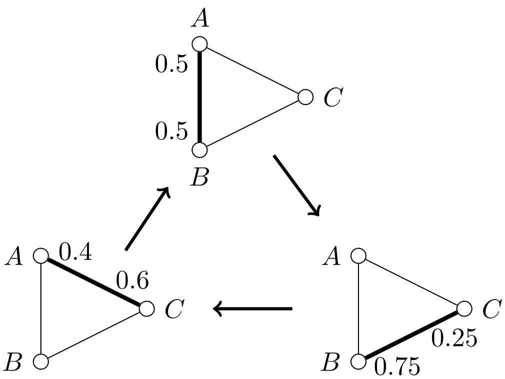


Figure 1.8: Outcomes of a network bargaining game.

For the cooperative matching game on this instance, only one of the three possible pairs can apply, which means that as a group they can obtain a funding of at most 1. The smaller group consisting of Adam and Betty can obtain a funding of 1, which means that Charlotte gets 0. Likewise, the smaller group consisting of Adam and Charlotte can obtain a funding of 1, so Betty also gets 0. But then the smaller group consisting of Betty and Charlotte gets 0, while they can obtain a funding of 1 by forming their own group. So, they leave the big group. Whatever we do here, there is always a smaller group that wants to leave the big group.

A new colleague, David, joins the team. He would also like to apply for a research grant with his colleagues, and, moreover, he has time to start two new research projects. To broaden his knowledge, he does not want to apply with Charlotte, as she works on subjects he is already familiar with, and he does not want to apply with the same person twice. Adam and Betty also have time to start two new research projects, and they too do not want to apply with the same person more than once. This instance is represented in Figure 1.9. In this figure there is a number above or below each vertex, called the capacity of the vertex, that represents the amount of research grants the person can apply for. All the edges still have weight 1, but we left this out of the figure to avoid clutter.

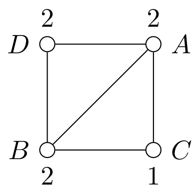


Figure 1.9: An instance.

This is a peculiar instance, as it is unstable in the network bargaining games setting, but it is stable in the cooperative matching games setting. It is interesting to note that this can only happen when not all vertex capacities are one. Let us consider why it is the case for this example.

First we take a look at the network bargaining game on this instance. Say the following pairs apply together for a research grant: Adam with Betty, Adam with David, and Betty with David. They all split the funding equally. This outcome is presented at the top of Figure 1.10. Looking only at Adam, Betty and Charlotte in this outcome, the outcome looks exactly like before in Figure 1.8. And indeed, the three of them form an unstable triangle again: Betty can apply with Charlotte instead of Adam, then Charlotte can apply with Adam instead of Betty, and finally, Adam can apply with Betty instead of Charlotte, to end up where we started. This process is displayed in Figure 1.10. Although none of the outcomes in Figure 1.10 are stable, this does not prove that no stable outcome exists, as other outcomes are possible. Even so, in this instance, there is no stable outcome.

Now for the cooperative matching game on this instance, at most three of the possible pairs can apply at the same time, which means that as a group they can obtain a funding of at most 3. Now let us take a look at what the smaller

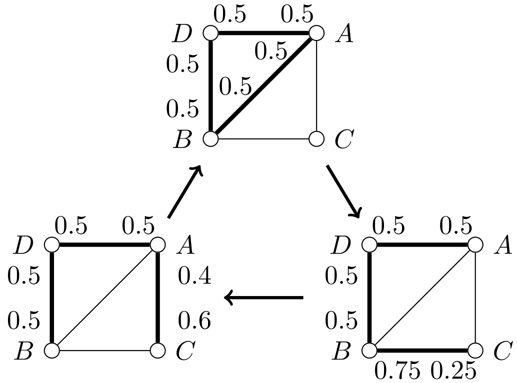


Figure 1.10: Outcomes of a network bargaining game.

groups can obtain. The smaller group consisting of Adam, Betty and David can apply for a total of 3 as well. So Charlotte gets 0, otherwise the three of them would leave the big group. The smaller group consisting of Adam, Betty and Charlotte can apply for a total of 2, which means that David can get at most 1: 3 to divide in total, minus the 2 that the others can obtain without him. The same holds for Adam and Betty, they can also get at most 1. Since we have 3 to divide, we can give all three of them exactly 1. So the outcome here is Adam, Betty and David all get 1 and Charlotte gets 0. This outcome is represented in Figure 1.11, where the numbers above and below each vertex now represent the outcome, instead of the capacities. Any smaller group consisting of only two persons can apply for a funding of at most 1, and in this outcome, all those groups get 1 or 2. So, no smaller group can obtain more by leaving the big group, which means this outcome is stable.

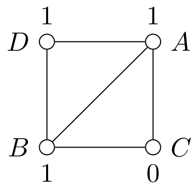


Figure 1.11: An outcome of a cooperative matching game.

In the remainder of Part I we first introduce notation and general definitions. Then we formally state the problem and give an overview of our results and previously known results.

In Part II we look at *stabilizing* network bargaining games: modify an unstable instance to obtain a stable instance. For example, we saw that the instance in Figure 1.7 is unstable, and the instance in Figure 1.1 is stable. So we can stabilize the instance in Figure 1.7 by removing the edge between A and B, thereby not allowing Adam and Betty to apply for a research grant together. Besides removing edges, other modifications that we consider are removing vertices and reducing the capacity of vertices. In our examples, these modifications correspond to not allowing a person to apply for any research grants, and lowering the amount of research grants a person is allowed to apply for.

In Part III we look at the set of stable outcomes for cooperative matching games, which is called the *core*. Given an outcome to the game, we are interested in determining if this outcome is stable, that is, if it is in the core. We also consider a dynamic setting where, after a stable outcome is found for the game, the instance might change. For example, some persons might decide they do not want to participate after all. To anticipate changes, we want to find a stable outcome that minimizes the amount of value we expect participants to lose when the instance changes.

Chapter 2

Notation and General Definitions

In Section 2.1 we introduce graph theory notation and definitions. In Section 2.2 we discuss the circuits of the fractional c-matching polytope and basic fractional c-matchings.

2.1 Graph Theory

A graph $G = (V, E)$ consists of a set V of vertices and a set E of edges connecting the vertices, that is, if $e \in E$ then $e = \{u, v\}$ for some $u, v \in V$. We use the shorthand notation $e = uv$. We say that u and v are adjacent, that u and v are neighbors, and that u and v are incident with e. We denote by $N(v) = \{u \in V : uv \in E\}$ the neighborhood of v, by $N^+(v) = N(v) \cup \{v\}$, and by $\delta(v) = \{e \in E : v \in e\}$ all edges incident with v. We denote by $d_v = |\delta(v)|$ the degree of v. For any $F \subseteq E$, we define $\delta^F(v) = \{e \in F : v \in e\}$ as the set of edges of F incident with v, and $d_v^F = |\delta^F(v)|$ as the degree of v with respect to the edges in F. We let $n = |V|$ and $m = |E|$.

A graph $G = (V, E)$ is bipartite if there are two subsets $V_1, V_2 \subseteq V$ that partition the vertices V, such that for all edges $uv \in E$ either $u \in V_1$, $v \in V_2$ or $v \in V_1$, $u \in V_2$.

We consider graphs with edge weights $w \in \mathbb{R}^E_{\geq 0}$ and vertex capacities $c \in \mathbb{Z}^V_{\geq 0}$. We refer to a graph G with edge weights w and vertex capacities c as (G, w, c), we say that G is a weighted, capacitated graph. If all edge weights are one, we denote this by $(G, 1, c)$, and we say that G is a unit-weight, capacitated graph. If all vertex capacities are one, we denote this by $(G, w, 1)$, and we say that G is a weighted, unit-capacity graph. If G is a unit-weight, unit-capacity graph we denote this by $(G, 1, 1)$.

For any set X, subset $Y \subset X$ and vector $x \in \mathbb{R}^X$, we denote by $x(Y) = \sum_{e \in Y} x_e$. For example, for $F \subseteq E$ we denote by $w(F)$ the total weight of the edges in F. For any two vectors $x, y \in \mathbb{R}^X$, we denote by $x^\top y = \sum_{e \in X} x_e y_e$.

We denote a (uv-)walk W by listing its edges and endpoints sequentially, that is, by $W = (u; e_1, \ldots, e_k; v)$. We define its inverse as $W^{-1} = (v; e_k, \ldots, e_1; u)$. We say a walk is closed if $u = v$. A trail is a walk in which edges do not repeat. A path is a trail in which internal vertices do not repeat. A cycle is a path which starts and ends at the same vertex. If we refer to the edge set of a walk W, we just write W. Note that this can be a multiset.

Let $S \subseteq V$, then $G[S]$ is the graph induced by the vertices of S, and $G \setminus S = G[V \setminus S]$ is the graph with the vertices in S removed. Let $F \subseteq E$, then $G \setminus F = (V, E \setminus F)$ is the graph with the edges in F removed. Let S be a multiset of vertices V. We denote by $G[c_S - 1]$ the graph G with the capacity of all vertices in S reduced by one. Note that, if a vertex appears, for example, twice in S, its capacity is reduced by two. We use c^{S-1} to refer to the capacities in $G[c_S - 1]$. For a vertex $s \in S$, with $S \setminus s$ we mean removing s just once from S.

2.1.1 Matching

Given a graph $(G, w, 1)$, a matching is a subset $M \subseteq E$ such that $d_v^M \leq 1$ for all $v \in V$. We denote the weight of a maximum-weight matching by $\nu(G)$, formally defined as

$$\nu(G) = \max\{w(M) : d_v^M \leq 1 \ \forall v \in V, M \subseteq E\}.$$

The linear programming (LP) relaxation of this problem is given by

$$\nu_f(G) = \max\{w^\top x : x(\delta(v)) \leq 1 \ \forall v \in V, x \geq 0\}.$$

We say that an x feasible for this LP is a fractional matching. The LP dual of this is

$$\tau_f(G) = \min\{1^\top y : y_u + y_v \geq w_{uv} \ \forall uv \in E, y \geq 0\}.$$

We say that a y feasible for this LP is a fractional vertex cover. By standard LP theory and duality theory we have $\nu(G) \leq \nu_f(G) = \tau_f(G)$, for all graphs $(G, w, 1)$. The complementary slackness conditions of $\nu_f(G)$ and $\tau_f(G)$ are

$$x_{uv} = 0 \quad \text{or} \quad y_u + y_v = w_{uv} \qquad \text{for all } uv \in E, \tag{2.1a}$$

$$y_v = 0 \quad \text{or} \quad x(\delta(v)) = 1 \qquad \text{for all } v \in V. \tag{2.1b}$$

In the next section we define c-matchings and structures involving c-matchings M, like M-alternating walks. These structures also apply to matchings.

2.1.2 Capacity-Matching

Given a graph (G, w, c), a c-matching (capacity-matching) is a subset $M \subseteq E$ such that $d_v^M \leq c_v$ for all $v \in V$. Note that we can assume without loss of generality that $c_v \leq d_v$ for all $v \in V$. We denote the weight of a maximum-weight c-matching by $\nu^c(G)$, formally defined as

$$\nu^c(G) = \max\{w(M) : d_v^M \leq c_v \ \forall v \in V, M \subseteq E\}.$$

The LP relaxation of this problem is given by

$$\nu_f^c(G) = \max\{w^\top x : x(\delta(v)) \leq c_v \ \forall v \in V, 0 \leq x \leq 1\}.$$

We say that an x feasible for this LP is a fractional c-matching. The LP dual of this is

$$\tau_f^c(G) = \min\{c^\top y + 1^\top z : y_u + y_v + z_{uv} \geq w_{uv} \ \forall uv \in E, y \geq 0, z \geq 0\}.$$

We say that a (y, z) feasible for this LP is a fractional vertex cover. It will be clear from context if we mean a fractional vertex cover y or a fractional vertex cover (y, z). By standard LP theory and duality theory we have $\nu^c(G) \leq \nu_f^c(G) = \tau_f^c(G)$, for all graphs (G, w, c). The complementary slackness conditions of $\nu_f^c(G)$ and $\tau_f^c(G)$ are

$$x_{uv} = 0 \quad \text{or} \quad y_u + y_v + z_{uv} = w_{uv} \qquad \text{for all } uv \in E, \tag{2.2a}$$
$$y_v = 0 \quad \text{or} \quad x(\delta(v)) = c_v \qquad \text{for all } v \in V, \tag{2.2b}$$
$$z_{uv} = 0 \quad \text{or} \quad x_{uv} = 1 \qquad \text{for all } uv \in E. \tag{2.2c}$$

Given a c-matching M, we say that $v \in V$ is *exposed* if $d_v^M = 0$, *covered* if $d_v^M > 0$, *unsaturated* if $d_v^M < c_v$ and *saturated* if $d_v^M = c_v$. We also use these terms for fractional c-matchings x, for example, v is saturated if $x(\delta(v)) = c_v$.

For a walk W (possibly a multiset) and a c-matching M (not a multiset), we define $W \setminus M = \{e \in W : e \notin M\}$ and $W \cap M = \{e \in W : e \in M\}$. For example, let $W = (u; e_1, e_2, e_3, e_1, e_2; v)$ and $M = \{e_2\}$, then $W \setminus M = \{e_1, e_3, e_1\}$ and $W \cap M = \{e_2, e_2\}$. We let $\triangle$ denote the symmetric difference operator: $W \triangle M = (W \setminus M) \cup (M \setminus W)$ for two sets W and M.

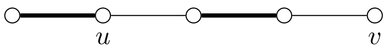
Definition 2.1. We say that a walk W is M-alternating (w.r.t. a c-matching M) if its edges are alternating between M and $E \setminus M$. We say W is M-augmenting if it is M-alternating and $w(W \setminus M) > w(W \cap M)$. An M-alternating uv-walk W is *proper* if $W \triangle M$ is a c-matching.

Definition 2.2. Given an M-alternating walk $W = (u; e_1, \ldots, e_k; v)$ and an $\varepsilon > 0$, the *ε-augmentation* of W is the vector $x^{M/W}(\varepsilon) \in \mathbb{R}^E$ given by

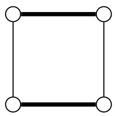
$$x_e^{M/W}(\varepsilon) = \begin{cases} 1 - \kappa(e)\varepsilon & \text{if } e \in M, \\ \kappa(e)\varepsilon & \text{if } e \notin M, \end{cases}$$

where $\kappa(e) = |\{i \in [k] : e_i = e\}|$. We say that W is *feasible* if there exists an $\varepsilon > 0$ such that the corresponding ε-augmentation of W is a fractional c-matching.

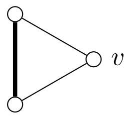
To get a better understanding of proper and feasible, we characterize proper and feasible for different kinds of walks. *(i) Nonclosed walks:* An M-alternating walk $W = (u; e_1, \ldots, e_k; v)$, where $u \neq v$, is proper and feasible if and only if the following hold: (a) either $e_1 \in M$ or $d_u^M \leq c_u - 1$, and (b) either $e_k \in M$ or $d_v^M \leq c_v - 1$. *(ii) Even-length closed walks:* An M-alternating walk $W = (v; e_1, \ldots, e_k; v)$, such that k is even, is always proper and feasible. *(iii) Odd-length closed walks:* An M-alternating walk $W = (v; e_1, \ldots, e_k; v)$, such that k is odd, is proper if and only if either $e_1, e_k \in M$ or $d_v^M \leq c_v - 2$, and feasible if and only if either $e_1, e_k \in M$ or $d_v^M \leq c_v - 1$. See Figure 2.1 for examples of these different kinds of walks.



(a) This nonclosed M-alternating walk is proper and feasible. The nonclosed M-alternating walk between u and v is neither proper nor feasible, because $d_u^M = 1 = c_u$.



(b) This even-length closed M-alternating walk is proper and feasible.



(c) This odd-length closed M-alternating walk is feasible, but not proper, because $d_v^M = 0 = c_v - 1$. If $c_v = 2$, then it would be proper and feasible.

Figure 2.1: Examples of different types of M-alternating walks. All capacities are 1, unless specified otherwise.

Proposition 2.1. *A feasible M-alternating walk with distinct endpoints is proper.*

Proof. Let $W = (u; e_1, \ldots, e_k; v)$ be a feasible M-alternating walk with $u \neq v$. If $e_1 \in M$, then $d_u^{W \triangle M} = d_u^M - 1 \leq c_u - 1$. If $e_1 \notin M$, then $d_u^{W \triangle M} = d_u^M + 1$.

Furthermore, since W is feasible, we have $x^{M/W}(\varepsilon)(\delta(u)) \leq c_u$. We also have $x^{M/W}(\varepsilon)(\delta(u)) = d_u^M + \varepsilon > d_u^M$. Since c_u and d_u^M are both integral, we have $d_u^M + 1 \leq c_u$, which implies $d_u^{W \triangle M} \leq c_u$. The exact same argument can be repeated for v, by replacing u by v, and e_1 by e_k. It follows that W is proper. $\square$

We next define M-blossom, M-flower and M-bi-cycle. See Figure 2.2 for some examples of these structures.

Definition 2.3. An odd cycle $C = (v; e_1, \ldots, e_k; v)$ is called an *M-blossom* if it is M-alternating such that either e_1 and e_k are both in M, or are both not in M. The vertex v is called the *base* of the blossom.

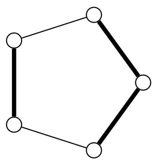
Definition 2.4. An *M-flower* $C \cup P$ consists of an M-blossom C with base v and an M-alternating path $P = (u; e_1, \ldots, e_k; v)$ such that (P, C, P^{-1}) is M-alternating and feasible. The vertex v is called the *root* of the flower. The flower is *M-augmenting* if

$$w(C \setminus M) + 2w(P \setminus M) > w(C \cap M) + 2w(P \cap M).$$

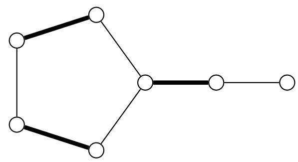
Definition 2.5. An *M-bi-cycle* $C \cup P \cup D$ consists of two M-blossoms C and D with bases u and v, respectively, and an M-alternating path $P = (u; e_1, \ldots, e_k; v)$ such that (P, D, P^{-1}, C) is M-alternating. The bi-cycle is *M-augmenting* if

$$w(C \setminus M) + 2w(P \setminus M) + w(D \setminus M) > w(C \cap M) + 2w(P \cap M) + w(D \cap M).$$

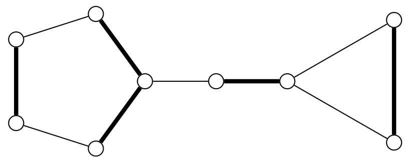
Note that, in the last two definitions, it may happen that P has no edges.



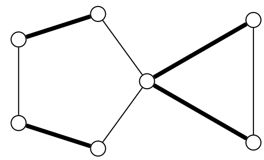
(a) An example of an M-blossom, or an M-flower with empty P.



(b) An example of an M-flower with nonempty P.



(c) An example of an M-bi-cycle with nonempty P.



(d) An example of an M-bi-cycle with empty P.

Figure 2.2: Examples of an M-blossom, M-flower and M-bi-cycle.

In the unit-capacity case it is well-known that a matching M has maximum weight if and only if there do not exist any proper M-augmenting paths or cycles. This generalizes to the capacitated case. We report a proof for completeness.

Theorem 2.1. *A c-matching M in (G, w, c) has maximum weight if and only if G does not contain a proper M-augmenting trail.*

Proof. ($\Rightarrow$) If G contains a proper M-augmenting trail T, then $M \triangle T$ is a c-matching and $w(M \triangle T) > w(M)$, which means M does not have maximum weight.

($\Leftarrow$) Let M be a c-matching in G such that M does not have maximum weight. We will show that G contains a proper M-augmenting trail. Let N be a maximum-weight c-matching, and consider the graph induced by $M \triangle N$. We construct a unit-capacity graph $\hat{G}$:

1. For each $v \in V$, define $b_v = \max\{d_v^{M \setminus N}, d_v^{N \setminus M}\}$, create copies $v_1, \ldots, v_{b_v}$ and add them to $V(\hat{G})$. Initialize $J_M(v) = J_N(v) = \{1, \ldots, b_v\}$.
2. For each $uv \in M \setminus N$, add a single edge $u_i v_j$ to both $E(\hat{G})$ and $\hat{M}$ with edge-weight w_{uv}, where $i \in J_M(u)$ and $j \in J_M(v)$ are chosen arbitrarily. Remove i and j from $J_M(u)$ and $J_M(v)$, respectively.
3. Likewise for each $uv \in N \setminus M$.

Observe that this construction establishes a natural weight-preserving bijection between E and $E(\hat{G})$. Furthermore, the sets $\hat{M}$ and $\hat{N}$ are matchings in $\hat{G}$, and $w(N) > w(M)$ implies $w(\hat{N}) > w(\hat{M})$. In particular, $\hat{M}$ is not maximum-weight in $\hat{G}$. Since $\hat{G}$ has unit-capacities, it contains at least one proper $\hat{M}$-augmenting path or cycle $\hat{T}$. Let $T = (u; e_1, \ldots, e_k; v)$ be the corresponding M-alternating walk in G. Since $\hat{T}$ does not repeat edges and is actually alternating between $\hat{M}$ and $\hat{N}$, T is alternating between $M \setminus N$ and $N \setminus M$, and also does not repeat edges, that is, T is a trail. Since $w(\hat{T} \setminus \hat{M}) > w(\hat{T} \cap \hat{M})$, we also have $w(T \setminus M) > w(T \cap M)$, that is, T is an M-augmenting trail. Thus, we only need to show that T is proper, that is, that $d_u^{T \triangle M} \leq c_u$ and $d_v^{T \triangle M} \leq c_v$.

Case 1: $\hat{T}$ is a proper $\hat{M}$-augmenting path. If $u = v$, then provided that at least one of e_1 and e_k is in M, then $d_u^{T \triangle M} \leq d_u^M \leq c_u$. If on the other hand neither of e_1 and e_k is in M, then the corresponding edges $\hat{e}_1$ and $\hat{e}_k$ in $\hat{T}$ are not in $\hat{M}$, and must therefore be in $\hat{N}$. Let u_i and u_j be the first and last vertices of $\hat{T}$, incident with $\hat{e}_1$ and $\hat{e}_k$, respectively. Note that u_i and u_j are distinct, since $\hat{T}$ is a path. Furthermore, since $\hat{T}$ is proper, u_i and u_j are not incident with edges from $\hat{M}$. Observe that by construction of $\hat{G}$, either all vertices in $\{u_1, \ldots, u_{b_u}\}$ are $\hat{M}$-covered or $\hat{N}$-covered. Since u_i and u_j are not $\hat{M}$-covered, all copies of u must be $\hat{N}$-covered. Hence, $d_u^{M \setminus N} \leq d_u^{N \setminus M} - 2$, which means $d_u^M \leq d_u^N - 2$. Finally, $d_u^{T \triangle M} = d_u^M + 2 \leq d_u^N \leq c_u$.

If $u \neq v$, we again consider whether or not e_1 and e_k are in M. Since u and v are distinct, these cases for e_1 and e_k are independent. If $e_1 \in M$, then $d_u^{T \triangle M} = d_u^M - 1 \leq c_u$. If $e_1 \notin M$, then $d_u^{T \triangle M} = d_u^M + 1$, and $\hat{e}_1$ is in $\hat{N}$, not in $\hat{M}$. Let u_i be the copy of u that is incident with $\hat{e}_1$. Since u_i is the first vertex of $\hat{T}$, and $\hat{T}$ is proper, u_i is not incident with any edge from $\hat{M}$. By construction of $\hat{G}$, every copy of u must be $\hat{N}$-covered. Hence, $d_u^{M \setminus N} \leq d_u^{N \setminus M} - 1$, which means $d_u^M \leq d_u^N - 1$. Therefore, $d_u^{T \triangle M} = d_u^M + 1 \leq d_u^N \leq c_u$. By symmetry of u and v, we also have $d_v^{T \triangle M} \leq c_v$ both if $e_k \in M$ and $e_k \notin M$.

Case 2: $\hat{T}$ is a $\hat{M}$-augmenting cycle. In this case $u = v$ and exactly one of e_1 and e_k is in M and one is not, which means $d_u^{T \triangle M} = d_u^M \leq c_u$. $\square$

2.2 Fractional c-Matching Polytope

The polytope of fractional c-matchings in G is $\mathcal{P}_{\text{FCM}}(G, c)$, formally defined as

$$\mathcal{P}_{\text{FCM}}(G, c) = \left\{ x \in \mathbb{R}^E : x(\delta(v)) \leq c_v \ \forall v \in V, 0 \leq x \leq 1 \right\}.$$

We write $\mathcal{P}_{\text{FCM}}$ if G and c are clear from the context or irrelevant.

We first explain some general polyhedral terminology. A convex combination of points $x_1, \ldots, x_k$ is $\alpha_1 x_1 + \cdots + \alpha_k x_k$, where $\alpha_1, \ldots, \alpha_k \geq 0$ and $\alpha_1 + \cdots + \alpha_k = 1$. An extreme point, or vertex, of a polyhedron is a point in the polyhedron that cannot be written as a convex combination of other points in the polyhedron. Equivalently, an extreme point of a polyhedron ($\subseteq \mathbb{R}^n$) is a point for which n linearly independent constraints are tight, that is, satisfied with equality. An edge of a polyhedron is a line of points such that for all those points the same $n - 1$ linearly independent constraints are tight. The end points of an edge are vertices of the polyhedron, and the edge itself is the line of all convex combinations of those two vertices.

2.2.1 Circuits

Let e be an edge of a polyhedron $\mathcal{P} = \{x \in \mathbb{R}^n : Ax = b, Bx \leq d\}$, where A and B are integral matrices, and b and d are rational vectors. The *edge direction* of e is the (single) vector $v - w$ for any two distinct points v and w on e. Taking two different points results in a scalar multiple of the same edge direction. The *circuits* of a polyhedron (described by A and B) are all potential edge directions that can appear for any choice of rational b and d (see Theorem 1.8 in Finhold [22]). Let $\mathcal{C}(\mathcal{P})$ denote the set of circuits of $\mathcal{P}$ with co-prime integer components. Note that $\mathcal{C}(\mathcal{P})$ contains two edge directions, which are each other's negative, for every potential edge. We use the notion of circuits of a polyhedron, instead of circuits of A and B, as any minimal description of $\mathcal{P}$ yields the same set of circuits (see Lemma 3 in Kafer [31]).

For a characterization of the circuits of the fractional c-matching polytope $\mathcal{P}_{\text{FCM}}(G,c)$ we rely on De Loera et al. [16], who defined five classes of graphs $(\mathcal{E}_1,\mathcal{E}_2,\mathcal{E}_3,\mathcal{E}_4,\mathcal{E}_5)$, listed below. See Figure 2.3 for examples of these subgraphs.

(i) Let $\mathcal{E}_1$ denote the set of all subgraphs $F \subseteq G$ such that F is an even cycle.

(ii) Let $\mathcal{E}_2$ denote the set of all subgraphs $F \subseteq G$ such that F is an odd cycle.

(iii) Let $\mathcal{E}_3$ denote the set of all subgraphs $F \subseteq G$ such that F is a path.

(iv) Let $\mathcal{E}_4$ denote the set of all subgraphs $F \subseteq G$ such that $F = C \cup P$, where C is an odd cycle, and P is a nonempty path that intersects C only at one endpoint.

(v) Let $\mathcal{E}_5$ denote the set of all subgraphs $F \subseteq G$ such that $F = C_1 \cup P \cup C_2$, where C_1 and C_2 are odd cycles, and P is a path satisfying the following: if P is nonempty, then C_1 and C_2 are vertex-disjoint and P intersects each C_i exactly at its endpoints; if P is empty then C_1 and C_2 intersect at only one vertex.

A set of circuits can be associated to the subgraphs in these classes by defining (again, see Figure 2.3 for examples):

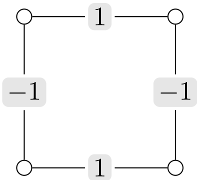
$$\mathcal{C}_1 = \bigcup_{F\in\mathcal{E}_1} \left\{ g \in \{-1,0,1\}^E : \begin{array}{ll} g(e) \neq 0 & \text{iff } e \in E(F) \\ g(\delta(v)) = 0 & \forall v \in V(F) \end{array} \right\},$$

$$\mathcal{C}_2 = \bigcup_{F\in\mathcal{E}_2} \left\{ g \in \{-1,0,1\}^E : \begin{array}{ll} g(e) \neq 0 & \text{iff } e \in E(F) \\ g(\delta(w)) \neq 0 & \text{for one } w \in V(F) \\ g(\delta(v)) = 0 & \forall v \in V(F)\setminus\{w\} \end{array} \right\},$$

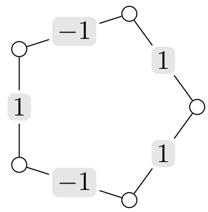
$$\mathcal{C}_3 = \bigcup_{F\in\mathcal{E}_3} \left\{ g \in \{-1,0,1\}^E : \begin{array}{ll} g(e) \neq 0 & \text{iff } e \in E(F) \\ g(\delta(v)) = 0 & \forall v : |\delta(v)\cap E(F)| = 2 \end{array} \right\},$$

$$\mathcal{C}_4 = \bigcup_{F=(C\cup P)\in\mathcal{E}_4} \left\{ g \in \mathbb{Z}^E : \begin{array}{ll} g(e) \neq 0 & \text{iff } e \in E(F) \\ g(\delta(v)) = 0 & \forall v : |\delta(v)\cap E(F)| \geq 2 \\ g(e) \in \{-1,1\} & \forall e \in E(C) \\ g(e) \in \{-2,2\} & \forall e \in E(P) \end{array} \right\},$$

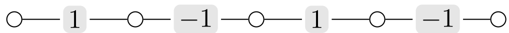
$$\mathcal{C}_5 = \bigcup_{F=(C_1\cup P\cup C_2)\in\mathcal{E}_5} \left\{ g \in \mathbb{Z}^E : \begin{array}{ll} g(e) \neq 0 & \text{iff } e \in E(F) \\ g(\delta(v)) = 0 & \forall v \in V(F) \\ g(e) \in \{-1,1\} & \forall e \in E(C_1\cup C_2) \\ g(e) \in \{-2,2\} & \forall e \in E(P) \end{array} \right\}.$$



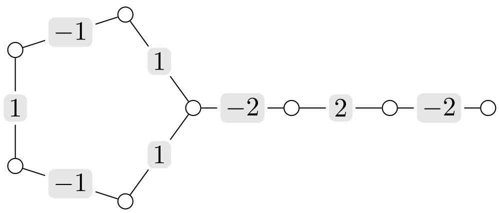
(a) An example of a subgraph in $\mathcal{E}_1$ and a circuit $g \in \mathcal{C}_1$ given on its edges.



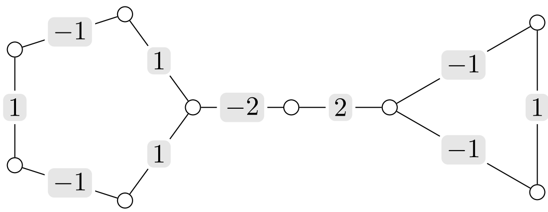
(b) An example of a subgraph in $\mathcal{E}_2$ and a circuit $g \in \mathcal{C}_2$ given on its edges.



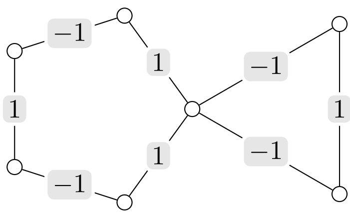
(c) An example of a subgraph in $\mathcal{E}_3$ and a circuit $g \in \mathcal{C}_3$ given on its edges.



(d) An example of a subgraph in $\mathcal{E}_4$ and a circuit $g \in \mathcal{C}_4$ given on its edges.



(e) An example of a subgraph in $\mathcal{E}_5$ with nonempty P and a circuit $g \in \mathcal{C}_5$ given on its edges.



(f) An example of a subgraph in $\mathcal{E}_5$ with empty P and a circuit $g \in \mathcal{C}_5$ given on its edges.

Figure 2.3: Examples of the subgraphs in $(\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, \mathcal{E}_4, \mathcal{E}_5)$ and the circuits defined on them.

De Loera et al. [16] show that $\mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3 \cup \mathcal{C}_4 \cup \mathcal{C}_5$ is the set of circuits of the *fractional matching polytope*, that is, $\mathcal{P}_{\mathrm{FCM}}$ with $c = 1$ and without the (redundant) constraints $x \leq 1$. Since the set of circuits stays the same if the right hand side vector changes (note that the constraints $x \leq 1$ are parallel to $x \geq 0$), the same set of circuits apply to $\mathcal{C}(\mathcal{P}_{\mathrm{FCM}})$. Hence, we have the following proposition.

Proposition 2.2. $\mathcal{C}(\mathcal{P}_{FCM}) = \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3 \cup \mathcal{C}_4 \cup \mathcal{C}_5$.

2.2.2 Basic Fractional c-Matchings

We refer to the vertices of $\mathcal{P}_{\mathrm{FCM}}$ as *basic* fractional c-matchings. The next result is well known, see for example Theorem 20 in Appa and Kotnyek [3] for half-integrality of general polytopes, but we provide a proof for completeness.

Theorem 2.2. *A fractional c-matching x is basic if and only if its components are equal to 0, $\frac{1}{2}$ or 1, and the edges with $x_e = \frac{1}{2}$ induce vertex-disjoint odd cycles with saturated vertices.*

Proof. ($\Rightarrow$) Let x be a basic fractional c-matching, and let H be a connected component of the graph induced by the edges with fractional value in x. First, note that H contains no even cycle, and no inclusion-wise maximal path with distinct endpoints: Otherwise, let D be an even cycle or an inclusion-wise maximal path with distinct endpoints in H. Let $g \in \mathcal{C}_1 \cup \mathcal{C}_3$ be the circuit associated to it. Then, $x + \varepsilon g$ and $x - \varepsilon g$ are both fractional c-matchings, for a small value of ε. However, as x is a convex combination of $x+\varepsilon g$ and $x-\varepsilon g$, this contradicts that x is an extreme point.

Let T be any spanning tree of H. First, assume there exist two distinct edges $f_1, f_2 \in E(H) \setminus E(T)$. Then, adding f_1 (respectively, f_2) to T creates an odd cycle D_1 (respectively, D_2). These cycles cannot intersect in an edge, otherwise their support would contain an even cycle. Hence, they must be edge disjoint. The cycles can also not intersect in more than one vertex, otherwise their support again contains an even cycle. So, they either intersect at one vertex, or are connected via a path in T. In either case, one can associate to these edges a circuit $g \in \mathcal{C}_5$. Then, $x+\varepsilon g$ and $x-\varepsilon g$ are both fractional c-matchings, for a small value of ε, reaching a contradiction again. These arguments show that there is a unique edge $f \in E(H) \setminus E(T)$.

Since H cannot contain inclusion-wise maximal paths, it contains at most one vertex of degree 1. If H contains exactly one such vertex u, then u and the odd cycle (created by adding f to T) are connected via a path. One can associate a circuit $g \in \mathcal{C}_4$ to the edges in this cycle and path. Again, considering $x + \varepsilon g$ and $x - \varepsilon g$ results in a contradiction. So H does not have any vertex of degree

1, which means that the endpoints of f must be the leaves of T. Hence, T is a path and H is a cycle. Necessarily, H must be odd.

Finally, no vertex in H can be unsaturated, as otherwise, we can associate a circuit $g \in \mathcal{C}_2$ to H, where the unsaturated vertex u is the only vertex with $g(\delta(u)) \neq 0$. Once again, we reach a contradiction by considering $x + \varepsilon g$ and $x - \varepsilon g$. Since H is an odd cycle with saturated vertices, the edges in H must have value $\frac{1}{2}$ in x.

In conclusion, if $x_e \notin \{0, 1\}$, it must equal $\frac{1}{2}$ and e must be part of an odd cycle. Furthermore, these odd cycles are vertex-disjoint, and all vertices part of an odd cycle are saturated.

($\Leftarrow$) Consider a vector w with $w_e = 1$ for all edges e in the support of x, and $w_e = -1$ for all other edges. Then, x is the unique optimal solution when maximizing the function w over $\mathcal{P}_{\text{FCM}}$. Hence x is an extreme point. $\square$

We partition the support of a basic fractional c-matching x into the odd cycles induced by $x_e = \frac{1}{2}$-edges: $\mathscr{C}_x = \{C_1, \ldots, C_q\}$ (later referred to as fractional odd cycles), and matched edges: $\mathscr{M}_x = \{e \in E : x_e = 1\}$. We next define two operations that change the value of a basic fractional c-matching on the support of a fractional odd cycle.

Definition 2.6. *Alternate rounding* $C = (v; e_1, \ldots, e_{2k+1}; v) \in \mathscr{C}_x$ *exposing* v means replacing x_e by $\hat{x}_e = 0$ for all $e \in \{e_1, e_3, \ldots, e_{2k+1}\}$ and by $\hat{x}_e = 1$ for all $e \in \{e_2, e_4, \ldots, e_{2k}\}$. Similarly, *alternate rounding* $C \in \mathscr{C}_x$ *covering* v means replacing x_e by $\hat{x}_e = 1$ for all $e \in \{e_1, e_3, \ldots, e_{2k+1}\}$ and by $\hat{x}_e = 0$ for all $e \in \{e_2, e_4, \ldots, e_{2k}\}$.

Let $\mathcal{X}$ be the set of basic maximum-weight fractional c-matchings in G. Define $\gamma(G) = \min_{x \in \mathcal{X}} |\mathscr{C}_x|$, as the minimum number of fractional odd cycles in the support of any basic maximum-weight fractional c-matching in G. Koh and Sanità [36] propose an algorithm to obtain a basic maximum-weight fractional matching with minimum number of fractional odd cycles ($|\mathscr{C}_x| = \gamma(G)$). We generalize their result to c-matchings.

Theorem 2.3. *A basic maximum-weight fractional c-matching x with $|\mathscr{C}_x| = \gamma(G)$ can be computed in polynomial time.*

Proof. Given a graph $(G = (V, E), w, c)$, we reduce it to a unit-capacity graph $(\hat{G} = (\hat{V}, \hat{E}), \hat{w}, 1)$ using the following reduction. We replace each edge $e = uv$ by two new vertices: e_u, e_v, and three new edges: ue_v, $e_v e_u$, $e_u v$ all of weight w_e. We replace each (original) vertex v by c_v copies: $v_1, \ldots, v_{c_v}$ (recall that we can assume without loss of generality that $c_v \leq d_v$), and each edge ve_u incident with v, by an edge between each of the c_v copies of v and e_u, all of the same weight as ve_u (so w_e).

Next, we compute a basic maximum-weight fractional matching $\hat{x}$ in $\hat{G}$ with minimum number of odd cycles ($|\mathscr{C}_{\hat{x}}| = \gamma(\hat{G})$), using the algorithm from Koh and Sanità [36]. Consider the subgraph of $\hat{G}$ corresponding with an edge $e \in E$. Since $\hat{x}$ is basic and has maximum weight, there are only a few possibilities for this subgraph, see Figures 2.4a to 2.4e.

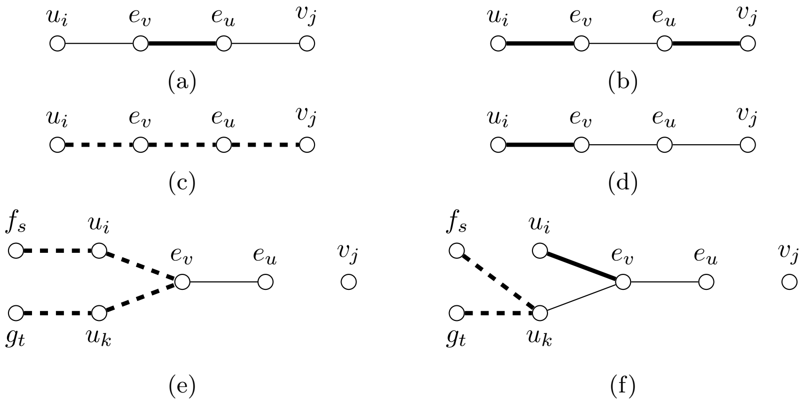


Figure 2.4: Figures 2.4a to 2.4e indicate the possible scenarios for the subgraph of $\hat{G}$ corresponding with an edge $e \in E$. Figure 2.4f shows how the matching in Figure 2.4e can be changed without affecting the weight of the matching. Normal, dashed and bold edges indicate an $\hat{x}$ value of 0, $\frac{1}{2}$ and 1, respectively. No edge between e_u and v_j indicates the value of the edges between e_u and the copies of v are irrelevant. For clarity, we have only drawn the relevant vertices and edges ($e = uv$, $f = us$, $g = ut$).

In fact, we can assume without loss of generality we only have the scenarios shown in Figures 2.4a to 2.4c. Indeed, Figures 2.4d and 2.4e can be transformed into Figures 2.4a and 2.4f, respectively, without affecting $\hat{w}^\top \hat{x}$ and $|\mathscr{C}_{\hat{x}}|$. The scenario in Figure 2.4f, depending on what is happening between e_u and the copies of v, corresponds to the scenario in Figure 2.4b, 2.4d, or 2.4e. If Figure 2.4f corresponds with Figure 2.4d, it can again be transformed into Figure 2.4a. If Figure 2.4f corresponds with Figure 2.4e, then note that now there is a matched edge between e_u and one of the copies of v. So, it can be transformed into Figure 2.4f another time, such that this time Figure 2.4f corresponds to Figure 2.4b.

Then $\hat{x}$ can be translated to a fractional c-matching x in G, as follows: set $x_e = 0$ in case of Figure 2.4a, $x_e = 1$ in case of Figure 2.4b, and $x_e = \frac{1}{2}$ in case of Figure 2.4c. We have $w^\top x = \hat{w}^\top \hat{x} - w(E)$: in Figure 2.4a the weight of x is 0, while the weight of $\hat{x}$ is w_e, in Figure 2.4b the weight of x is w_e, while the weight of $\hat{x}$ is $2w_e$, and finally, in Figure 2.4c the weight of x is $\frac{1}{2}w_e$, while the weight of $\hat{x}$ is $\frac{3}{2}w_e$. A fractional odd cycle $\hat{C}$ in $\hat{x}$ maps to a fractional cycle C in x with length $|\hat{C}|/3$ (three edges in $\hat{C}$ map to one edge in C), which is odd.

Suppose there is a fractional odd cycle C in x such that there is an unsaturated vertex u on C. Then there is a copy u_i of u in $\hat{G}$ that is exposed. Let $e = uv$ be an edge on C. We can then change $\hat{x}$ as follows: set $\hat{x}_{u_i e_v} = 1$, and alternate round $\hat{C}$ exposing e_v. One can check using complementary slackness that this gives a new maximum-weight matching in $\hat{G}$, that is basic and has less fractional odd cycles than $\hat{x}$, a contradiction. So all vertices on fractional odd cycles in x are saturated.

Now suppose that there is some vertex u that is on at least two fractional odd cycles in x. Then there are at least two copies u_i and u_j of u that are both on a fractional odd cycle in $\hat{x}$. Let a_k and b_l be such that $\hat{x}_{u_i a_k} = \hat{x}_{u_j b_l} = \frac{1}{2}$. Now set these to zero, and instead set $\hat{x}_{u_i b_l} = \hat{x}_{u_j a_k} = \frac{1}{2}$. This merges the two fractional odd cycles into one fractional even cycle, and does not change the weight of $\hat{x}$ ($\hat{w}_{u_i a_k} = \hat{w}_{u_j a_k}$ and $\hat{w}_{u_j b_l} = \hat{w}_{u_i b_l}$). To make $\hat{x}$ basic again, we remove this fractional even cycle by alternatingly decreasing/increasing $\hat{x}$ on the edges of the cycle to zero/one. One can check using complementary slackness that this gives a new maximum-weight matching in $\hat{G}$, that is basic and has less fractional odd cycles than $\hat{x}$, a contradiction. So all fractional odd cycles in x are vertex-disjoint.

Hence we find that x is basic. It follows that $|\mathscr{C}_x| = |\mathscr{C}_{\hat{x}}|$.

Note that given an x, we can similarly translate it into $\hat{x}$, such that $\hat{w}^\top \hat{x} = w^\top x + w(E)$ and $|\mathscr{C}_{\hat{x}}| = |\mathscr{C}_x|$. Hence, $\nu_f^c(G) = \nu_f(\hat{G}) - w(E)$ and $\gamma(G) = \gamma(\hat{G})$. So, x is our basic maximum-weight fractional c-matching in G with $|\mathscr{C}_x| = \gamma(G)$. $\square$

Chapter 3

Problem Definition and Results

In this dissertation we look at two types of matching games: *Network Bargaining Games (NBG)* and *Cooperative Matching Games (CMG)*. In Section 3.1 we formally introduce network bargaining games, and we introduce the stabilization problem that we study in Part II. In Section 3.2 we formally introduce cooperative matching games, and we introduce the two problems that we study in Part III. Finally, in Section 3.3 we discuss the connection between network bargaining games and cooperative matching games.

3.1 Network Bargaining Games

Network bargaining games were introduced by Kleinberg and Tardos [34] as a generalization of Nash's 2-player bargaining solution [41]. This work by Kleinberg and Tardos [34] is quite popular, and network bargaining games have been studied from a lot of perspectives since then. Bateni et al. [7] introduced capacitated network bargaining games as a generalization of network bargaining games. Instances of network bargaining games are described by a graph $G = (V, E)$ with edge weights $w \in \mathbb{R}^E_{\geq 0}$ (and vertex capacities $c \in \mathbb{Z}^V_{\geq 0}$), where the vertices and the edges model the players and their potential interactions, respectively. We next formulate network bargaining games formally, on unit-capacity and capacitated graphs.

In a unit-capacity instance of NBG, each player can enter in a deal with at most one of their neighbors in the graph, and together they agree on how to split the value of the deal, which is given by the weight of the corresponding edge. Hence, an outcome is naturally associated with a matching $M \subseteq E$ in G representing the deals, and an allocation vector $y \in \mathbb{R}^V_{\geq 0}$ with $y_u + y_v = w_{uv}$ if $uv \in M$, and $y_v = 0$ if v is exposed by M. The *outside option* of a player is the maximum profit a player can receive by abandoning their current deal (if they are in a deal currently) and forming a new deal with a different neighbor, under the condition that this does not decrease the profit of that neighbor. Formally, the outside option of player u with respect to the outcome (M, y) is defined as

$$\alpha_u(M, y) = \max\left\{0, \max_{v:uv \in E \setminus M} \{w_{uv} - y_v\}\right\}.$$

We omit (M, y) when it is clear from context. An outcome (M, y) is *stable* if it satisfies $y_v \geq \alpha_v$ for all players $v \in V$, which means no player has an incentive to deviate from (M, y).

Observation 3.1. The definition of outcome reflects the complementary slackness conditions in Equation (2.1): $y_u + y_v = w_{uv}$ if $uv \in M$, that is, if $x_{uv} \neq 0$, and $y_v = 0$ if v is exposed by M, that is, if $x(\delta(v)) \neq 1$. Moreover, the definition of stable reflects the vertex cover constraints for edges not in M: $y_v \geq \alpha_v \geq w_{uv} - y_u$ for all u such that $uv \in E \setminus M$. It follows that in a stable outcome (M, y), M is a maximum-weight matching and y is a minimum fractional vertex cover.

In a capacitated instance of NBG, each player v can instead enter in a deal with at most c_v of their neighbors. In this case an outcome is naturally associated with a c-matching M in G, and a vector $a \in \mathbb{R}^{2E}_{\geq 0}$ with $a_{uv} + a_{vu} = w_{uv}$ if $uv \in M$ and $a_{uv} = a_{vu} = 0$ otherwise. The allocation vector $y \in \mathbb{R}^{V}_{\geq 0}$ associated with the outcome (M, a) represents the total value acquired by the vertices, formally defined as $y_v = \sum_{u:uv \in E} a_{vu}$ for all $v \in V$. The outside option is now defined as

$$\alpha_u(M, a) = \max\left\{0, \max_{v:uv \in E \setminus M}\left\{w_{uv} - \mathbb{1}[d_v^M = c_v] \min_{w:vw \in M} a_{vw}\right\}\right\},$$

where $\mathbb{1}[d_v^M = c_v]$ equals 1 if $d_v^M = c_v$, and 0 otherwise. The difference with the unit-capacity outside option is that now we have to check if v has already used all of its capacity. Because in that case, v needs to abandon one of its deals to be able to form a new deal with u. So, to make sure that v's profit does not decrease, v needs to get at least $\min_{w:vw \in M} a_{vw}$. But if v has not yet used all of its capacity, then v is free to form another deal, and so v's profit will not decrease if v gets nothing from the deal with u. An outcome (M, a) is *stable* if it satisfies $a_{uv} \geq \alpha_u$ and $a_{vu} \geq \alpha_v$ for all deals $uv \in M$ and $\alpha_u = 0$ if u is unsaturated.

3.1.1 Stabilization

As we will discuss in more detail in Section 3.3, the existence of a stable outcome for the network bargaining game on a graph G is equivalent to the property $\nu_f(G) = \nu(G)$ in unit-capacity graphs, and the property $\nu_f^c(G) = \nu^c(G)$ in capacitated graphs. We say that a graph $(G, w, 1)$ is *stable* if $\nu_f(G) = \nu(G)$, and a graph (G, w, c) is stable if $\nu_f^c(G) = \nu^c(G)$. Recall that $\gamma(G) = \min_{x \in \mathcal{X}} |\mathscr{C}_x|$ denotes the minimum number of fractional odd cycles. As already noted by Koh and Sanità [36] for the unit-capacity case, we have the following.

Proposition 3.1. *A graph (G, w, c) is stable ($\nu_f^c(G) = \nu^c(G)$) if and only if $\gamma(G) = 0$.*

From this proposition it easily follows that not all graphs are stable, for example, odd cycles. The stabilization problem follows naturally: minimally modifying a graph to turn it into a stable graph, that is, such that the modified graph satisfies $\nu_f^c(G) = \nu^c(G)$. Or equivalently: minimally modifying a graph to ensure the existence of a stable outcome for the network bargaining game on the modified graph. Stabilization problems attracted a lot of attention in the literature in the past years, see for example [1, 8, 9, 11, 12, 13, 30, 36, 37]. The modifications that we consider are removing vertices, decreasing the capacity of vertices, and removing edges. In the context of network bargaining games, these modifications correspond with blocking players completely, blocking part of the capacity of players, and blocking interactions between players, respectively. Previous work studied these modifications on unit-capacity graphs. We instead study them on capacitated graphs.

In Part II of this dissertation we study the following stabilization problems.

The vertex-stabilizer problem: Given a graph $G = (V, E)$ with edge weights $w \in \mathbb{R}^E_{\geq 0}$ and vertex capacities $c \in \mathbb{Z}^V_{\geq 0}$, find a minimum-cardinality subset $S \subseteq V$ of vertices such that $\nu_f^c(G \setminus S) = \nu^c(G \setminus S)$.

The capacity-stabilizer problem: Given a graph $G = (V, E)$ with edge weights $w \in \mathbb{R}^E_{\geq 0}$ and vertex capacities $c \in \mathbb{Z}^V_{\geq 0}$, find a minimum-cardinality multiset S of vertices V such that $\nu_f^c(G[c_S - 1]) = \nu^c(G[c_S - 1])$.

The edge-stabilizer problem: Given a graph $G = (V, E)$ with edge weights $w \in \mathbb{R}^E_{\geq 0}$ and vertex capacities $c \in \mathbb{Z}^V_{\geq 0}$, find a minimum-cardinality subset $F \subseteq E$ of edges such that $\nu_f^c(G \setminus F) = \nu^c(G \setminus F)$.

The M-vertex-stabilizer problem: Given a graph $G = (V, E)$ with edge weights $w \in \mathbb{R}^E_{\geq 0}$ and vertex capacities $c \in \mathbb{Z}^V_{\geq 0}$, and a c-matching M in G, find a minimum-cardinality subset $S \subseteq V$ of vertices such that $\nu_f^c(G \setminus S) = \nu^c(G \setminus S)$ and M is a maximum-weight c-matching in $G \setminus S$.

The M-edge-stabilizer problem: Given a graph $G = (V, E)$ with edge weights $w \in \mathbb{R}^E_{\geq 0}$ and vertex capacities $c \in \mathbb{Z}^V_{\geq 0}$, and a c-matching M in G, find a minimum-cardinality subset $F \subseteq E$ of edges such that $\nu_f^c(G \setminus F) = \nu^c(G \setminus F)$ and M is a maximum-weight c-matching in $G \setminus F$.

We refer to the subsets we are looking for in the above problems as ...-stabilizers: for example, a subset $S \subseteq V$ such that $\nu_f^c(G \setminus S) = \nu^c(G \setminus S)$ and M is a maximum-weight c-matching in $G \setminus S$, is called an M-vertex-stabilizer.

In Chapter 4 we discuss the vertex-, capacity- and edge-stabilizer problem, and in Chapter 5 we discuss the M-vertex- and M-edge-stabilizer problem.

Below we discuss the known results, and our results, for these problems. We also discuss related work and open problems.

Known Results: Vertex-Stabilizer. Ahmadian et al. [1] and Ito et al. [30] discuss the vertex-stabilizer problem on unit-weight, unit-capacity graphs. Both prove that the vertex-stabilizer problem in this setting is polynomial-time solvable. In addition, Ahmadian et al. [1] state that for any minimal-cardinality vertex-stabilizer $S \subseteq V$, we have $\nu(G \setminus S) = \nu(G)$, that is, S does not decrease the cardinality of a maximum matching. Koh and Sanità [36] generalize these results to weighted graphs. In particular, they state that the problem is polynomial-time solvable, and that the set $S \subseteq V$ that they compute satisfies $\nu(G \setminus S) \geq \frac{2}{3}\nu(G)$.

Our Results: Vertex- and Capacity-Stabilizer. We prove that on unit-weight, capacitated graphs the vertex-stabilizer problem is NP-hard, and even APX-hard already when $c \leq 3$. We give stronger inapproximability results for arbitrary capacity values. These results are presented in Section 4.1.

Another way to generalize the vertex-stabilizer problem to capacitated graphs, is by reducing the capacity of vertices: When a vertex has capacity one, reducing its capacity by one is the same as removing the vertex. We extend the algorithm of Koh and Sanità [36] to capacitated graphs by reducing the capacity of vertices, and show that $\nu^c(G[c_S - 1]) \geq \frac{2}{3}\nu^c(G)$ is still satisfied for the computed set $S \subseteq V$. Our solution reduces the capacity of each vertex by at most one, which is fair in terms of the network bargaining game, as no player will have its capacity dramatically reduced compared to others. Restricted to unit-weight graphs (still capacitated), we generalize the result of Ahmadian et al. [1], that any minimal-cardinality capacity-stabilizer S satisfies $\nu^c(G[c_S - 1]) = \nu^c(G)$, that is, S does not decrease the cardinality of a maximum matching. This, in particular, also holds for the set S computed by our algorithm. These results are presented in Section 4.3.

Besides extending the previous known results to the capacitated setting, what we find interesting are the new arguments we rely on in our proofs. Previous results mainly used combinatorial techniques. We here instead rely on (new) polyhedral arguments and, in particular, on the notion of circuits of a polytope, which are a key concept in optimization. The main algorithmic idea behind the algorithm of Koh and Sanità [36] is the fact that the minimum number of fractional odd cycles in the support of a basic maximum-weight fractional matching provides a lower bound on the size of a stabilizer. Interestingly, our polyhedral view point allows us not only to deal more broadly with capacitated instances, but also to simplify some of the cardinal arguments previously used in the literature: in particular, our lower bound proof is more general and much simpler than the corresponding one in Koh and Sanità [36] for the unit-capacity setting. The (new) polyhedral tools that we use are presented in Section 4.2. Even though we apply these polyhedral tools only to the fractional c-matching polytope, our main result applies to polytopes in general.

Known Results: Edge-Stabilizer. Bock et al. [11] discuss the edge-stabilizer problem on unit-weight, unit-capacity graphs. They state that this problem is NP-hard, and no efficient $(2-\varepsilon)$-approximation algorithm exists for any $\varepsilon > 0$ assuming the Unique Games Conjecture (Khot [33]). In addition, they state that for any minimum-cardinality edge-stabilizer $F \subseteq E$, we have $\nu(G \setminus F) = \nu(G)$, that is, F does not decrease the cardinality of a maximum matching. Koh and Sanità [36] discuss the edge-stabilizer problem on weighted, unit-capacity graphs. They state that there is no constant factor approximation algorithm possible, unless $P = NP$. In addition, they give an $O(\Delta)$-approximation algorithm, where Δ is the maximum degree in the graph.

Our Results: Edge-Stabilizer. We extend the $O(\Delta)$-approximation algorithm of Koh and Sanità [36] to capacitated graphs. Restricted to unit-weight graphs (still capacitated), we generalize the result of Bock et al. [11], that any minimum-cardinality edge-stabilizer $F \subseteq E$ satisfies $\nu^c(G \setminus F) = \nu^c(G)$, that is, F does not decrease the cardinality of a maximum matching. We note that in weighted, capacitated graphs, there always exists an edge-stabilizer $F \subseteq E$ that satisfies $\nu^c(G \setminus F) = \nu^c(G)$, but that the size of such an edge-stabilizer could be much larger than the size of a minimum edge-stabilizer. These results are presented in Section 4.4. Here we also rely on the polyhedral tools presented in Section 4.2.

Known Results: M-Vertex-Stabilizer. Ahmadian et al. [1] discuss the M-vertex-stabilizer problem on unit-weight, unit-capacity graphs, and they assume the given matching is maximum. They state that the M-vertex-stabilizer problem in this setting is polynomial-time solvable. Koh and Sanità [36] state that the M-vertex-stabilizer problem on unit-weight, unit-capacity graphs is NP-hard, and no efficient $(2-\varepsilon)$-approximation algorithm exists for any $\varepsilon > 0$ assuming the Unique Games Conjecture (Khot [33]). Furthermore, they give an efficient 2-approximation algorithm on weighted, unit-capacity graphs, which is even an exact algorithm if the given matching has maximum weight.

Our Results: M-Vertex-Stabilizer. We extend the 2-approximation/ exact algorithm of Koh and Sanità [36] to capacitated graphs, by building upon an auxiliary construction of Farczadi et al. [21]. These results are presented in Section 5.1.

Known Results: M-Edge-Stabilizer. Bock et al. [11] discuss the M-edge-stabilizer problem on unit-weight, unit-capacity graphs, and they assume the given matching is maximum. They state that the M-edge-stabilizer problem in this setting is NP-hard, and no efficient $(2-\varepsilon)$-approximation algorithm exists for any $\varepsilon > 0$ assuming the Unique Games Conjecture (Khot [33]). Furthermore, they give an efficient 2-approximation algorithm.

Our Results: M-Edge-Stabilizer. We generalize the 2-approximation algorithm of Bock et al. [11] to capacitated graphs and arbitrary given matchings. On the other hand, we show that a straightforward generalization of their 2-approximation algorithm to weighted graphs does not work. These results are presented in Section 5.2.

Related Work. Stabilizer problems have been studied extensively in the literature, see Chandrasekaran [12] for a survey on the subject. We have already mentioned the stabilizer variants that we study in this dissertation, but there are more. Here we mention some known results of other stabilizer variants.

Ahmadian et al. [1] and Ito et al. [30] also discuss a vertex-weighted variant of the vertex-stabilizer problem, on unit-(edge-)weighted, unit-capacity graphs. Here each vertex has a nonnegative weight, which represents the cost of removing this vertex. Both show that this problem is NP-hard.

In addition, Ito et al. [30] study stabilizing by adding vertices or edges, on unit-weight, unit-capacity graphs. They show both problems can be solved in polynomial time. They also consider an edge-weighted variant of the edge-addition-stabilizer problem, where each possible edge has a nonnegative weight, which represents the cost of adding this edge. They show that this problem is NP-hard.

Chandrasekaran et al. [13] study the minimum fractional additive stabilizer problem. In this problem one is given a unit-weight, unit-capacity graph $G = (V, E)$, and the goal is to find a vector $c \in \mathbb{R}^E_{\geq 0}$ with minimum $\sum_{e \in E} c_e$ such that the graph G with edge weights $1 + c$ is stable. They give hardness results for this problem, and a nearly matching approximation algorithm.

A slightly different, but related problem is the problem of finding the minimum number of blocking pairs. For a unit-capacity network bargaining game and allocation y, a blocking pair is a pair $uv \in E$ of players such that $y_u + y_v < w_{uv}$. Biró et al. [9] show that this problem is NP-complete, even on unit-weight graphs. Könemann et al. [37] give an approximation algorithm for finding a minimum number of blocking pairs in sparse graphs. The blocking value of a blocking pair $uv \in E$ is $w_{uv} - y_u - y_v$. Biró et al. [8] show that, for weighted graphs, finding the minimum number of blocking pairs, and finding the minimum total blocking value, are NP-complete problems. In general, the set of all blocking pairs of an allocation y is not an edge-stabilizer, as for example pointed out by Könemann et al. [37]. Even so, the hardness proof of Biró et al. [8] does work for edge-stabilizers, hence also proving that the edge-stabilizer problem is NP-complete on weighted graphs.

Open Problems. Koh and Sanità [36] show that the vertex-stabilizer problem is polynomial-time solvable if $c = 1$, and we show that it is APX-hard already when $c \leq 3$. It remains to determine the hardness of the vertex-stabilizer problem for $c \leq 2$.

We note that there always exists a weight-preserving edge-stabilizer, but that in general, minimum edge-stabilizers are not weight-preserving. Koh and Sanità [36] show that there is no constant factor approximation algorithm possible for the edge-stabilizer problem, unless $P = NP$. Their proof also applies to the weight-preserving edge-stabilizer problem. This leaves finding a nonconstant approximation algorithm for the weight-preserving edge-stabilizer problem.

We show that, for the M-edge-stabilizer problem, a straightforward generalization of the 2-approximation algorithm of Bock et al. [11] to weighted graphs does not work. It remains to find an approximation algorithm, and perhaps a stronger hardness result, for the M-edge-stabilizer problem on weighted graphs.

3.2 Cooperative Matching Games

Cooperative matching games were introduced in the seminal paper of Shapley and Shubik 50 years ago [50], and have been widely studied since then. Cooperative matching games have also been studied in the capacitated setting, for example by Biró et al. [10]. Instances of cooperative matching games are described by a graph $G = (V, E)$ with edge weights $w \in \mathbb{R}^E_{\geq 0}$ (and vertex capacities $c \in \mathbb{Z}^V_{\geq 0}$), where the vertices model the players. The value of a maximum-weight (c-)matching, $\nu(G)$ ($\nu^c(G)$), is the total value that players can collectively accumulate. We next formulate cooperative matching games formally, on unit-capacity and capacitated graphs.

In a unit-capacity instance of CMG, each player can participate in one coalition with a subset of other players. If a subset of players $S \subseteq V$ forms a coalition, they can distribute the value of a maximum-weight matching in $G[S]$ ($\nu(G[S])$) among themselves. We define an allocation vector $y \in \mathbb{R}^V_{\geq 0}$, where y_v is the value allocated to player $v \in V$. An allocation is *stable* if $\sum_{v \in S} y_v \geq \nu(G[S])$ for all $S \subseteq V$, which means no subset of players has an incentive to deviate from the current set of coalitions, to form a coalition on their own. We are interested in the *core* of CMG, which consists of all stable allocation vectors when the *grand coalition* ($S = V$) is formed. The core is formally defined as

$$\text{core}(G) = \left\{ y \in \mathbb{R}^V_{\geq 0} : \sum_{v \in S} y_v \geq \nu(G[S]) \forall S \subseteq V, \sum_{v \in V} y_v = \nu(G) \right\}.$$

In a capacitated instance of CMG the value of subsets is instead given by maximum-weight c-matchings. The above applies by replacing all ν by ν^c.

We sometimes refer to unit-capacity cooperative matching games as matching games, and to capacitated cooperative matching games as c-matching games.

In Part III of this dissertation we consider two problems that involve cooperative matching games. We introduce these two problems in the next two sections.

3.2.1 Core Separation of 2-Matching Games

In Chapter 6 we consider 2-matching games, which are c-matching games with $c_v \leq 2$ for all players $v \in V$. We are interested in the problem of separating over the core:

Determine if a given allocation $y \in \mathbb{R}^V$ belongs to the core, or find a coalition that violates the corresponding constraint in

$$y(S) \geq \nu^c(G[S]) \text{ for all } S \subset V, \quad y(V) = \nu^c(G). \tag{3.1}$$

Known Results. Separating over the core of matching games is solvable in linear time; given $y \in \mathbb{R}^V_{\geq 0}$ with $y(V) = \nu(G)$, it is equivalent to checking if $y_u + y_v \geq w_{uv}$ for all edges $uv \in E$. In fact, the core admits a compact LP formulation: y is in the core if and only if y is a minimum fractional vertex cover with total value $\nu(G)$. This was first shown for bipartite graphs by Shapley and Shubik [50], and later generalized to arbitrary graphs by for example Deng et al. [17] and Paulusma [42]. Differently, Biró et al. [10] show that separating over the core of c-matching games (which they call multiple partners matching games) is co-NP-complete, even on bipartite graphs with $c = 3$ and $w = 1$ (see Theorem 13 in Biró et al. [10]). On the other hand, 2-matching games seem to still behave nicely: they state that separating over the core of 2-matching games is solvable in polynomial time (see Theorem 12 in Biró et al. [10]). However, their proof contains a flaw.

Our Results. Our first result is to fix the flaw in the proof of Theorem 12 in Biró et al. [10], hence showing that separating over the core of 2-matching games is solvable in polynomial time. We show this in Section 6.1. Having a polynomial-time separation oracle over the (convex) set of core allocations, implies that we can optimize over the corresponding polytope in polynomial time via the ellipsoid method (Grötschel et al. [25], Grötschel et al. [26], Khachiyan [32]). A natural question is then whether there exists a compact extended formulation for it. In fact, there exist polytopes for which a polynomial-time separation oracle is known, but no compact extended formulation exists, such as the perfect matching polytope (Rothvoss [45]). Our second result is a positive answer to this question: there exists a compact extended formulation that describes the core of 2-matching games. We show this in Section 6.2.

Open Problems. The compact LP formulation for the core of matching games implies that the core is nonempty if and only if the graph is stable. (This relation is explored more in Section 3.3.) Consequently, the stabilization results for unit-capacity graphs also apply to matching games, in the sense that they ensure a nonempty core. For c-matching games, having a nonempty core is not equivalent to stability of the graph, which leaves the problem of stabilizing c-matching games. And perhaps our compact extended formulation can be exploited for stabilizing 2-matching games.

3.2.2 Two-Stage Assignment Games

In Chapter 7 we study a two-stage stochastic version of the assignment game. The assignment game is a matching game on a bipartite graph. A two-stage problem is a problem split in two stages, where the instance can change between the stages, and the objective is typically to minimize the difference between the solutions of the two stages. In a two-stage stochastic problem, the second-stage instance is sampled from a probability distribution. Studying combinatorial problems in the two-stage setting is a popular area of research (see for example [4, 14, 27, 39, 44, 51, 52]). Recently, this setting has been studied for prominent game theory problems such as stable matchings (Bampis et al. [5], Faenza et al. [20]). In particular, Faenza et al. [20] study a two-stage stochastic stable matching problem where in the second stage the set of vertices changes. It seems natural to study this setting applied to other prominent games involving the structure of matchings. We here study this two-stage stochastic setting in the context of cooperative matching games. We focus on assignment games, that is, cooperative matching games on bipartite graphs, because they are guaranteed to have a nonempty core.

Given a first-stage assignment game instance, in a second stage we can have some players leaving the game, new players joining the game, and/or some additions and removals in the edge set of the original instance. Formally, we represent this as having a new graph describing the instance in the second stage, that can be any bipartite graph as long as it keeps the same bipartition as the first-stage graph for the vertices that stay in the game. The second-stage instance is sampled from some distribution $\mathcal{D}$. We denote the starting (first-stage) graph by $G_0 = (V_0, E_0)$, and for any second-stage scenario $S \sim \mathcal{D}$, we denote the corresponding graph by $G_S = (V_S, E_S)$. We want to compute a core allocation in both stages. The goal is to minimize the expected total loss of the remaining players (that is, decrease in allocation value). Mathematically, the two-stage stochastic assignment game is

$$\min_{y \in \text{core}(G_0)} \mathbb{E}_{S\sim\mathcal{D}} \left[\min_{y^S \in \text{core}(G_S)} \sum_{v \in V_0 \cap V_S} \lambda_v \left[y_v - y_v^S \right]^+ \right], \tag{2SAG}$$

where $[x]^+ = \max\{0, x\}$, and $\lambda \geq 0$ is the dissatisfaction of players per unit loss of allocation value.

Our results. We first consider the setting where the probability distribution $\mathcal{D}$ is given explicitly in Section 7.1. We observe that the problem can be modeled as an LP, and hence it is solvable in time polynomial in the size of the graph and the number of second-stage scenarios. Interestingly, we prove that the feasible region is an integral polyhedron. For this, we show that the problem can be modeled as a flow problem in a suitable auxiliary graph, and then exploit duality. We leverage this integrality result in two ways.

First, we exploit it when considering a probability distribution given implicitly, as described in Section 7.2. The integrality result allows us to mimic the arguments used in Faenza et al. [20] for two-stage stable matching, hence showing that in this setting the problem is computationally hard to solve, but it can be approximated using the well-known sample average approximation (SAA) method (Kleywegt et al. [35]).

Second, the integrality property reveals a close relationship with the well-known multistage vertex cover problem, which we discuss in Section 7.3. It is known that the multistage vertex cover problem is NP-hard even with only two stages and a bipartite graph at each stage (Fluschnik et al. [23]). However, as a consequence of our findings, we can show that the problem becomes polynomial-time solvable when the bipartition remains consistent across all stages.

3.3 Connection Between the Games

As briefly touched upon in Chapter 1, network bargaining games and cooperative matching games seem quite related. And, indeed, they are. In this section we explore the connection between the existence of stable outcomes for NBG, the existence of stable allocations for CMG (nonempty core), and stability of the graph, that is, the property $\nu_f(G) = \nu(G)$ in unit-capacity graphs, and the property $\nu_f^c(G) = \nu^c(G)$ in capacitated graphs. In particular, we show that these three properties are equivalent in unit-capacity graphs, and that this equivalence does not completely extend to capacitated graphs. We conclude this section with a remark about what consequences this equivalence has for the stabilization problem.

Theorem 3.1. *Given a graph* $(G, w, 1)$*, the following are equivalent:*

(i) *G is stable ($\nu_f(G) = \nu(G)$),*

(ii) *there exists a stable outcome for the network bargaining game on G,*

(iii) *there exists an allocation in the core of the cooperative matching game on G.*

The first part of this equivalence, $(i) \iff (ii)$, follows from Kleinberg and Tardos [34], and the second part, $(i) \iff (iii)$, is given in Theorem 1 in Deng et al. [17]. We also give an alternative proof.

Proof. $((i) \Rightarrow (ii))$ Since G is stable, we have $\nu(G) = \tau_f(G)$. That means there exists a maximum-weight matching M and a minimum fractional vertex cover y that both have total value $\nu(G)$. We show that (M, y) is a stable outcome. Let x be the indicator vector of M, then x is a maximum-weight fractional matching. Complementary slackness tells us that $x_{uv} = 0$ or $y_u + y_v = w_{uv}$ for all edges $uv \in E$, and $y_v = 0$ or $x(\delta(v)) = 1$ for all vertices $v \in V$. It follows that, if $uv \in M$, that is, if $x_{uv} = 1$, then $y_u + y_v = w_{uv}$, and if v is exposed, that is, if $x(\delta(v)) = 0$, then $y_v = 0$. So (M, y) is an outcome. Furthermore, because y is a fractional vertex cover, we have $y_u + y_v \geq w_{uv}$ for all edges $uv \in E$. Hence, (M, y) is a stable outcome.

($(ii) \Rightarrow (i)$) Let (M, y) be a stable outcome. Then y is a fractional vertex cover, as $y_u + y_v \geq w_{uv}$ for all edges $uv \in E$ and $y \geq 0$. Thus, $\tau_f(G) \leq 1^\top y$. Furthermore, we know that (M, y) satisfies $y_u + y_v = w_{uv}$ if $uv \in M$ and $y_v = 0$ is v is exposed. This implies $\sum_{v \in V} y_v = w(M)$. Finally, M is a matching, so $w(M) \leq \nu(G)$. Combining all of this, we find that $\tau_f(G) \leq \nu(G)$, which implies that G is stable.

($(i) \Rightarrow (iii)$) Since G is stable, we have $\nu(G) = \tau_f(G)$. That means there exists a fractional vertex cover y with total value $\nu(G)$. We show that y is in the core. First note that $y \in \mathbb{R}^V_{\geq 0}$ is an allocation, and that the total value of y is $\nu(G)$. Let $S \subseteq V$, and let M be a maximum-weight matching in $G[S]$. Note that $V(M) \subseteq S$. We have

$$\nu(G[S]) = \sum_{uv \in M} w_{uv} \leq \sum_{uv \in M} y_u + y_v = \sum_{v \in V(M)} y_v \leq \sum_{v \in S} y_v,$$

so y is stable, and hence, y is in the core.

($(iii) \Rightarrow (i)$) Let y be an allocation in the core. Then, $y \in \mathbb{R}^V_{\geq 0}$, $\sum_{v \in S} y_v \geq \nu(G[S])$ for all $S \subseteq V$, and $\sum_{v \in V} y_v = \nu(G)$. If we choose $S = \{u, v\}$ for any edge $uv \in E$, the inequality tells us that $y_u + y_v \geq w_{uv}$. Hence, y is a fractional vertex cover of total value of $\nu(G)$. It follows that $\tau_f(G) \leq \nu(G)$, which implies that G is stable. $\square$

This equivalence does not completely extend to capacitated graphs: We still have $(i) \iff (ii)$, proven in Corollary 3.3 in Bateni et al. [7], and $(i) \implies (iii)$, which follows from Lemma 3.4 in Bateni et al. [7][1]. However, $(iii) \not\Longrightarrow (i), (ii)$, which was shown using an example in Theorem 11 in Biró et al. [10][2]. We again also give an alternative proof.

Theorem 3.2. *Given a graph (G, w, c), and the following statements:*

(i) G is stable ($\nu^c_f(G) = \nu^c(G)$),

(ii) there exists a stable outcome for the network bargaining game on G,

(iii) there exists an allocation in the core of the cooperative matching game on G,

we have $(i) \iff (ii)$ and $(i) \implies (iii)$, but $(iii) \not\Longrightarrow (i), (ii)$.

Proof. ($(i) \Rightarrow (ii)$) Since G is stable, we have $\nu^c(G) = \tau^c_f(G)$. Let M be a maximum-weight c-matching with indicator vector x, and let (y, z) be a minimum fractional vertex cover. Because G is stable, they satisfy complementary

[1] Bateni et al. [7] assume that the graph is bipartite, but bipartiteness is not needed in their proof.

[2] Biró et al. [10] actually investigates the relation between CMG and the stable fixtures problem with payments, but the latter turns out to be the same as NBG.

slackness (see Equation (2.2)). We define $a \in \mathbb{R}_+^{2E}$ as $a_{uv} = y_u + \frac{1}{2} z_{uv}$ if $x_{uv} = 1$ and $a_{uv} = 0$ otherwise. We show that (M, a) is a stable outcome. By definition we already have $a_{uv} = a_{vu} = 0$ if $uv \notin M$. Let $uv \in M$, we have

$$a_{uv} + a_{vu} = y_u + \tfrac{1}{2} z_{uv} + y_v + \tfrac{1}{2} z_{vu} = y_u + y_v + z_{uv} = w_{uv},$$

where the last equality holds by complementary slackness. Hence, (M, a) is an outcome. We verify (M, a) is stable. Consider a_{uv} for some edge $uv \in M$. We make a distinction between three cases:

(1) $\alpha_u = 0$,

(2) $\alpha_u = w_{uw}$ for some w such that $uw \notin M$ and w unsaturated,

(3) $\alpha_u = w_{uw} - a_{wx}$ for some w, x such that $uw \notin M$, $wx \in M$ and w saturated.

In case (1) we clearly satisfy $a_{uv} \geq \alpha_u$. In case (2), we have $y_u + y_w + z_{uw} \geq w_{uw}$ by feasibility of (y, z) and $y_w = 0$ and $z_{uw} = 0$ by complementary slackness. Hence, $y_u \geq w_{uw}$. Then,

$$a_{uv} = y_u + \tfrac{1}{2} z_{uv} \geq y_u \geq w_{uw} = \alpha_u.$$

In case (3) we have $y_u + y_w + z_{uw} \geq w_{uw}$ by feasibility of (y, z), and $z_{uw} = 0$ by complementary slackness. Hence, $y_u + y_w \geq w_{uv}$. Then,

$$\begin{aligned} \alpha_u = w_{uw} - a_{wx} \leq y_u + y_w - (y_w + \tfrac{1}{2} z_{wx}) = y_u - \tfrac{1}{2} z_{wx} \\ \leq y_u \leq y_u + \tfrac{1}{2} z_{uv} = a_{uv}. \end{aligned}$$

In all cases we have $a_{uv} \geq \alpha_u$. Now consider α_u for some unsaturated vertex u. Since M is a maximum-weight matching, we have that all vertices v such that $uv \notin M$ must be saturated. Hence,

$$\begin{aligned} \alpha_u &= \max\left(0, \max_{uv \notin M} w_{uv} - \mathbb{1}\left[d_v^M = c_v\right] \min_{vw \in M} a_{vw}\right), \\ &= \max\left(0, \max_{uv \notin M} w_{uv} - \min_{vw \in M} y_v + \tfrac{1}{2} z_{vw}\right), \\ &= \max\left(0, \max_{uv \notin M} w_{uv} - y_v - \tfrac{1}{2} \min_{vw \in M} z_{vw}\right). \end{aligned}$$

We have $y_u + y_v + z_{uv} \geq w_{uv}$ by feasibility, and $y_u = 0$ and $z_{uv} = 0$ by complementary slackness, for any $uv \notin M$. Hence, $y_v \geq w_{uv}$, or equivalently, $w_{uv} - y_v \leq 0$ for any $uv \notin M$. Then,

$$w_{uv} - y_v - \tfrac{1}{2} \min_{vw \in M} z_{vw} \leq -\tfrac{1}{2} \min_{vw \in M} z_{vw} \leq 0,$$

for any $uv \notin M$. Consequently, $\alpha_u = \max\{0, \leq 0\} = 0$, and so (M, a) is a stable outcome.

($(ii) \Rightarrow (i)$) Let (M, a) be a stable outcome. Let x be the indicator vector of M, and define (y, z) as follows:

$$y_u = \begin{cases} \min_{uv \in M} a_{uv} & \text{if } d_u^M = c_u, \\ 0 & \text{otherwise,} \end{cases} \qquad z_{uv} = \begin{cases} w_{uv} - y_u - y_v & \text{if } uv \in M, \\ 0 & \text{otherwise.} \end{cases}$$

Observe that x and (y, z) are defined in such a way that they satisfy complementary slackness (see Equation (2.2)). Since x is the indicator vector of M, it is feasible for $\nu^c(G)$, and hence also for $\nu_f^c(G)$. If in addition (y, z) is feasible for $\tau_f^c(G)$, then by complementary slackness, x and (y, z) form an optimal primal-dual pair for $\nu_f^c(G)$ and $\tau_f^c(G)$. Therefore, $w^\top x = \nu_f^c(G) \geq \nu^c(G) \geq w(M) = w^\top x$, which implies G is stable. It remains to verify that (y, z) is a fractional vertex cover.

By definition we have $y \geq 0$. Let $uv \in M$, then we have

$$y_u + y_v \leq \min_{uw \in M} a_{uw} + \min_{vw \in M} a_{vw} \leq a_{uv} + a_{vu} = w_{uv}.$$

The first inequality holds by definition of y and the last equality by definition of outcome. So, we have $y_u + y_v \leq w_{uv}$ for all $uv \in M$. It follows that $z \geq 0$. To check the constraints $y_u + y_v + z_{uv} \geq w_{uv}$ for all $uv \in E$, we make a distinction between three cases:

(a) $uv \in M$,

(b) $uv \notin M$, $d_u^M = c_u$ and $d_v^M = c_v$,

(c) $uv \notin M$, $d_u^M = c_u$ and $d_v^M < c_v$.

If $uv \notin M$, at least one of u and v is saturated, because M is a maximum-weight c-matching. The case that only v is saturated is similar to case (c), and can be argued in the same way. In case (a) the constraint is satisfied, and even holds with equality, by construction of z. In case (b) we have

$$y_u + y_v + z_{uv} = \min_{uw \in M} a_{uw} + \min_{vx \in M} a_{vx} \geq \alpha_u + \alpha_v \geq w_{uv} - y_v + w_{uv} - y_u,$$

where the first inequality holds because (M, a) is stable. We can rewrite this inequality as $2(y_u + y_v) + z_{uv} \geq 2w_{uv}$. By definition $z_{uv} = 0$, so we can divide both sides by two to find that the constraint is satisfied. In case (c) we have

$$y_u + y_v + z_{uv} = \min_{uw \in M} a_{uw} \geq \alpha_u \geq w_{uv},$$

where the first inequality holds because (M, a) is stable. This shows that the constraint is also satisfied in this case. Therefore, (y, z) is a fractional vertex cover.

($(i) \Rightarrow (iii)$) Let (y, z) be optimal for $\tau_f^c(G)$. Since G is stable, $c^\top y + 1^\top z = \nu^c(G)$. We define $y^*_v = c_v y_v + \sum_{u:uv\in E} \frac{1}{2} z_{uv}$ for all $v \in V$. We show that y^* is an allocation in the core. Note that $y^* \geq 0$, and $\sum_{v\in V} y^*_v = c^\top y + 1^\top z = \nu^c(G)$. It remains to show that y^* is stable. Consider any $S \subseteq V$. Let M be a maximum-weight c-matching in $G[S]$, so $w(M) = \nu^c(G[S])$. We have

$$\nu^c(G[S]) = \sum_{uv\in M} w_{uv} \leq \sum_{uv\in M} y_u + y_v + z_{uv} \leq \sum_{v\in V(M)} c_v y_v + \sum_{uv\in M} z_{uv},$$

where the first inequality holds because (y, z) is a fractional vertex cover, and the second because on the left hand side y_v is counted d_v^M times, and $d_v^M \leq c_v$. Since $V(M) \subseteq S$, $M \subseteq E(S) \subseteq E$, $y \geq 0$ and $z \geq 0$, we have

$$\nu^c(G[S]) \leq \sum_{v\in S} c_v y_v + \sum_{uv\in E(S)} z_{uv} = \sum_{v\in S}\left(c_v y_v + \sum_{u:uv\in E(S)} \tfrac{1}{2} z_{uv}\right),$$
$$\leq \sum_{v\in S}\left(c_v y_v + \sum_{u:uv\in E} \tfrac{1}{2} z_{uv}\right) = \sum_{v\in S} y^*_v.$$

Hence y^* is stable, and so y^* is in the core.

($(iii) \not\Rightarrow (i)$) We prove this by giving a graph for which there is an allocation in the core, but which is not stable. Let G be the graph given in Figure 3.1. It is quite easy to see that $\nu^c(G) = 3$ and $\nu_f^c(G) = 3.5$, thus G is not stable. One can check that $y = (1, 1, 1, 0)$ is in the core.

Figure 3.1: The graph G with unit-weights. On the left: the graph G where the values close to the vertices indicate the capacities. Bold edges indicate a maximum c-matching. On the right: the graph G where the values close to the vertices indicate the allocation y. A maximum fractional c-matching is given by $x_e = \frac{1}{2}$ for dashed edges, $x_e = 1$ otherwise.

($(iii) \not\Rightarrow (ii)$) Follows directly from $(iii) \not\Longrightarrow (i)$ and $(ii) \Longrightarrow (i)$. □

The next corollary follows from the proof of $(ii) \Longrightarrow (i)$.

Corollary 3.1. *Given a graph (G, w, c), if (M, a) is a stable outcome for the network bargaining game on G, M is a maximum-weight c-matching in G.*

As a consequence of Theorems 3.1 and 3.2, the stabilization problem we discussed before in Section 3.1.1 also applies to matching games, but not to c-matching games. In particular, in unit-capacity graphs, stabilizing a graph with the minimum number of modifications also means ensuring a nonempty core for cooperative matching games with the minimum number of modifications. On the other hand, in capacitated graphs, stabilizing a graph with the minimum number of modifications, does ensure a nonempty core for cooperative matching games, but not necessarily with the minimum number of modifications. The previously known stabilization results study stabilization problems on unit-capacity graphs, and hence minimally stabilize both network bargaining games and cooperative matching games. We instead consider stabilization problems on capacitated graphs, and hence minimally stabilize only network bargaining games.

Part II

Stabilization

Chapter 4

The Stabilizer Problem

In this chapter we discuss the vertex-, capacity- and edge-stabilizer problems, which we defined before as follows.

The vertex-stabilizer problem: Given a graph $G = (V, E)$ with edge weights $w \in \mathbb{R}^E_{\geq 0}$ and vertex capacities $c \in \mathbb{Z}^V_{\geq 0}$, find a minimum-cardinality subset $S \subseteq V$ of vertices such that $\nu^c_f(G \setminus S) = \nu^c(G \setminus S)$.

The capacity-stabilizer problem: Given a graph $G = (V, E)$ with edge weights $w \in \mathbb{R}^E_{\geq 0}$ and vertex capacities $c \in \mathbb{Z}^V_{\geq 0}$, find a minimum-cardinality multiset S of vertices V such that $\nu^c_f(G[c_S - 1]) = \nu^c(G[c_S - 1])$.

The edge-stabilizer problem: Given a graph $G = (V, E)$ with edge weights $w \in \mathbb{R}^E_{\geq 0}$ and vertex capacities $c \in \mathbb{Z}^V_{\geq 0}$, find a minimum-cardinality subset $F \subseteq E$ of edges such that $\nu^c_f(G \setminus F) = \nu^c(G \setminus F)$.

We start by showing the hardness of the vertex-stabilizer problem in Section 4.1. In Section 4.2 we state our polyhedral results, which we use in the following sections. Finally, in Sections 4.3 and 4.4 we give our results for the capacity- and edge-stabilizer problem, respectively.

Section 4.1 is based on (part of) [V1]. Sections 4.2 to 4.4 are based on [V3].

4.1 Vertex-Stabilizer

We start by proving the vertex-stabilizer problem is APX-hard already when $c \leq 3$.

Theorem 4.1. *The vertex-stabilizer problem on capacitated graphs is APX-hard, even if all edges have unit-weight and $c \leq 3$.*

We use a reduction from the well-known vertex cover problem.

Minimum Vertex Cover: *Given a graph $G = (V, E)$, find a* vertex cover *of minimum cardinality, where a vertex cover is a set $C \subseteq V$ such that every edge in E is incident to at least one vertex in C.*

It is known that Minimum Vertex Cover is APX-hard even in subcubic graphs, that is, graphs where each vertex has degree at most 3 (Alimonti and Kann [2]).

Proof of Theorem 4.1. Given a graph G, we construct an instance $(G_\Gamma, 1, c)$ of the vertex-stabilizer problem as follows.

For a pair of adjacent vertices u and v in G, we replace the edge $e = uv$ with a gadget Γ_{uv} consisting of vertices $V(\Gamma_{uv})$ and edges $E(\Gamma_{uv})$:

$$V(\Gamma_{uv}) = \{u, v, e_1, e_2, e_3, e_4\},$$
$$E(\Gamma_{uv}) = \{ue_1, ve_1, e_1e_2, e_2e_3, e_2e_4, e_3e_4\}.$$

See Figure 4.1 for reference. For each gadget Γ_{uv}, we set the capacities of

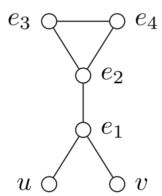


Figure 4.1: Example of the gadget Γ_{uv}.

u and v (in G_Γ) to their respective degrees in G (at most 3 by assumption). Furthermore, we set the capacity of e_1 to 2, and the capacities of e_2, e_3, and e_4 all to 1. The edges are set to have unit-weights. Note that $c \leq 3$. The key point is:

Claim 4.1. *G has a vertex cover of size at most k if and only if $(G_\Gamma, 1, c)$ has a vertex-stabilizer of size at most k.*

Proof. ($\Rightarrow$) Let C be a vertex cover of G such that $|C| \leq k$. Note that C corresponds with a subset of the vertices in G_Γ. We claim that C is a vertex-stabilizer of G_Γ. To see this, we create a c-matching and fractional vertex cover in $G_\Gamma \setminus C$ that satisfy complementary slackness, therefore proving that $G_\Gamma \setminus C$ is stable.

Note that no gadget Γ_{uv} in $G_\Gamma \setminus C$ retains both u and v, because C is a vertex cover of G. Therefore, in each gadget Γ_{uv} we can consider the c-matching

$$M_{\Gamma_{uv}} = \bigcup_{t \in \{u,v,e_2\} \setminus C} \{e_1 t\} \cup \{e_3 e_4\}.$$

Then $M = \bigcup_{uv \in E(G)} M_{\Gamma_{uv}}$ is a c-matching in $G_\Gamma \setminus C$, by the choice of capacity of each vertex from V. We next construct a fractional vertex cover $(\tilde{y}, \tilde{z})$, where:

- $\tilde{y}_t = \frac{1}{2}$ for all $t \in \cup_{e=uv \in E}\{e_2, e_3, e_4\}$, and $\tilde{y}_t = 0$ otherwise (that is, for all $t \in \cup_{e=uv\in E}\{e_1\} \cup V$),
- $\tilde{z}_f = \frac{1}{2}$ for all edges $f \in E(G_\Gamma)$ with $f = e_1 e_2$ for some $e = uv \in E$, $\tilde{z}_f = 1$ for all edges $f \in E(G_\Gamma)$ with $f = e_1 u, e_1 v$, for some $e = uv \in E$, and $\tilde{z}_f = 0$ otherwise.

One can check that M and $(\tilde{y}, \tilde{z})$ satisfy complementary slackness.

($\Leftarrow$) Let S' be a vertex-stabilizer of $(G_\Gamma, 1, c)$ with cardinality $|S'| \leq k$. We will find a vertex cover of G with size at most k using the following claim:

Claim 4.2. *S' contains at least one vertex from every gadget Γ_{uv} of G_Γ.*

Proof. Suppose for the sake of contradiction that there is a gadget Γ_{uv} of G_Γ which does not contain any vertices of S'. Since $G_\Gamma \setminus S'$ is stable, the associated c-matching LP has an integral optimal solution x^*.

Construct a vector $\overline{x}$ as follows:

- $\overline{x}_f = x^*_f$ for all $f \in E(G_\Gamma) \setminus E(\Gamma_{uv})$,
- $\overline{x}_{ue_1} = \overline{x}_{ve_1} = 1$,
- $\overline{x}_{e_1 e_2} = 0$,
- $\overline{x}_{e_2 e_3} = \overline{x}_{e_2 e_4} = \overline{x}_{e_3 e_4} = \frac{1}{2}$.

Note that $\overline{x}$ is a feasible solution of the c-matching LP, and $\sum_{e \in E(\Gamma_{uv})} \overline{x}_e = 3.5$. However, observe that $\sum_{e \in E(\Gamma_{uv})} x^*_e \leq 3$. To see this, note that the graph has unit-weights and each matched edge uses two capacity units (one for each of its endpoints). The total capacity that edges in Γ_{uv} can use in any matching is at most $\sum_{q \in V(\Gamma_{uv})} \min\{c_q, d_q^{E(\Gamma_{uv})}\} = 7$ ($3 \cdot 1$ for e_2, e_3, e_4, plus 2 for e_1, plus $2 \cdot 1$ for u, v). So any matching in this gadget has value at most $7/2 = 3.5$. By integrality of x^* we have $\sum_{e \in E(\Gamma_{uv})} \overline{x}_e = 3.5 > 3 \geq \sum_{e \in E(\Gamma_{uv})} x^*_e$, and hence $\sum_{e \in E(G_\Gamma)} \overline{x}_e > \sum_{e \in E(G_\Gamma)} x^*_e$, which contradicts the optimality of x^*. $\square$

Let $S'_{uv} = S' \cap V(\Gamma_{uv})$ for each gadget Γ_{uv}. Create a new set T'_{uv} by

$$T'_{uv} = \begin{cases} S'_{uv} \cap \{u, v\} & \text{if } S'_{uv} \cap \{u, v\} \neq \emptyset, \\ \{u\} \text{ xor } \{v\} \text{ (chosen arbitrarily)} & \text{otherwise.} \end{cases}$$

Note that $\emptyset \neq T'_{uv} \subseteq \{u, v\}$ by construction of T'_{uv}, and by Claim 4.2. Let $T' = \bigcup_{uv \in E(G)} T'_{uv}$. Since T' contains at least one of u and v for every gadget Γ_{uv}, and does not contain any vertices in $V(\Gamma_{uv}) \setminus \{u, v\}$, then T' is clearly a vertex cover in G. By construction, we have $|T'| \leq |S'|$, and so we have found a vertex cover whose size is at most k. □

The above result shows that any minimum vertex-stabilizer of $(G_\Gamma, 1, c)$ is of the same size as any minimum vertex cover of G. Further, any efficient α-approximation algorithm for the vertex-stabilizer problem would also yield an efficient α-approximation algorithm for minimum vertex cover. Since minimum vertex cover in subcubic graphs is APX-hard, this shows that the vertex-stabilizer problem is APX-hard, even when $c \leq 3$. □

The above hardness result uses bounded capacities for the vertices ($c \leq 3$). We next show a stronger inapproximability result for arbitrarily large capacities.

Theorem 4.2. *The vertex-stabilizer problem on capacitated graphs is NP-complete, even if all edges have unit-weight. Furthermore, no efficient $n^{1/3-\varepsilon}$-approximation algorithm exists for any $\varepsilon > 0$, unless $P = NP$.*

Note that, given an unstable graph (G, w, c), removing all vertices (but two) trivially yields a stable graph. This gives a (trivial) n-approximation algorithm for the vertex-stabilizer problem. The theorem above essentially implies that one cannot hope for a much better approximation. To prove it, we use:

Minimum Independent Dominating Set (MIDS): *Given a graph $G = (V, E)$, compute a minimum-cardinality subset $S \subseteq V$ that is independent (for all $uv \in E$ at most one of u and v is in S) and dominating (for all $v \in V$ at least one $u \in N^+(v)$ is in S).*

There is no efficient $n^{1-\varepsilon}$-approximation for any $\varepsilon > 0$ for the MIDS problem, unless $P = NP$ (Corollary 3 in Halldórsson [28]).

Proof of Theorem 4.2. The decision variant of the problem asks to find a vertex-stabilizer of size at most k. This problem is in NP, since if a vertex set S is given, it can be verified in polynomial time if $|S| \leq k$ and if $\nu^c(G \setminus S) = \nu^c_f(G \setminus S)$. We prove the NP-hardness and approximation factor by giving an approximation-preserving reduction from the MIDS problem.

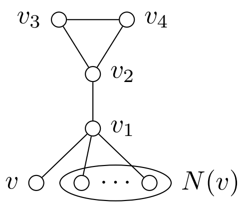
Let $G = (V, E)$ be an instance of the MIDS problem. For $v \in V$, we define the gadget Γ_v by

$$V(\Gamma_v) = N^+(v) \cup \{v_1, v_2, v_3, v_4\}\,,$$
$$E(\Gamma_v) = \left\{uv_1 : u \in N^+(v)\right\} \cup \{v_1v_2, v_2v_3, v_3v_4, v_2v_4\}\,.$$

For $e = uv \in E$ and $i \in \{1, \ldots, n\}$, we define the gadget Γ^i_{uv} by

$$V(\Gamma^i_{uv}) = \left\{u, v, e^i_1, e^i_2, e^i_3, e^i_4, e^i_5\right\},$$
$$E(\Gamma^i_{uv}) = \left\{ue^i_1, ve^i_1, e^i_1e^i_2, e^i_1e^i_3, e^i_3e^i_4, e^i_4e^i_5, e^i_3e^i_5\right\}.$$

See Figure 4.2 for an example of these gadgets. Now let G' be defined as the



(a) Gadget Γ_v. (b) Gadget Γ^i_{uv}.

Figure 4.2: Examples of gadgets.

union of all Γ_v and all Γ^i_{uv}, such that vertices from V overlap. We set the capacity as follows: $c_v = d_v^{E(G')} = (n+1)d^E_v + 1$ for all $v \in V$, $c_{v_1} = d^E_v + 1$ for all $v \in V$, $c_{e^i_1} = c_{e^i_3} = 2$ for $e^i_1, e^i_3 \in V(\Gamma^i_{uv})$ for all $e = uv \in E$ and $i \in \{1, \ldots, n\}$, and $c_v = 1$ for all remaining $v \in V(G')$. All edges are set to have unit-weight. The key point is:

Claim 4.3. *G has an independent dominating set of size at most k if and only if $(G', 1, c)$ has a vertex-stabilizer of size at most k.*

Proof. ($\Rightarrow$) Let S be an independent dominating set of G of size k. The vertices in S naturally correspond with vertices in G'. We show that S is a vertex-stabilizer of $(G', 1, c)$.

We define a c-matching M and fractional vertex cover (y, z) on $G' \setminus S$ as follows. First, set $y_v = 0$ for all $v \in V \setminus S$.

Next, for all $v \in V$, consider Γ_v. Add $\{uv_1 : u \in N^+(v) \setminus S\} \cup \{v_1v_2, v_3v_4\}$ to M. Note that at least one vertex from $N^+(v)$ is in S, since S is dominating. Set $y_{v_1} = 0$, $y_{v_2} = 1$, $y_{v_3} = y_{v_4} = 0.5$, $z_e = 1$ for all $e \in \{uv_1 : u \in N^+(v) \setminus S\}$ and $z_e = 0$ for the remaining edges in the gadget.

Finally, for all $e = uv \in E$ and $i \in \{1, \dots, n\}$, consider Γ^i_{uv}. Since S is independent, at most one of u and v is in S. If neither are in S, add both ue^i_1 and ve^i_1 to M. If one of them is in S, without loss of generality let it be u, then add ve^i_1 and $e^i_1 e^i_2$ to M. Furthermore, add $e^i_3 e^i_4$ and $e^i_3 e^i_5$ to M. Set $y_{e^i_1} = 1$, $y_{e^i_2} = 0$, $y_{e^i_3} = y_{e^i_4} = y_{e^i_5} = 0.5$, and $z_f = 0$ for all edges f in the gadget.

Let x be the indicator vector of M. One can verify that x and (y, z) satisfy the complementary slackness conditions for $\nu^c_f(G' \setminus S)$ and $\tau^c_f(G' \setminus S)$. Since x is integral, this implies that $G' \setminus S$ is stable.

($\Leftarrow$) Let S be a vertex-stabilizer of $(G', 1, c)$ of size k. We show that: (i) S contains at least one vertex of each gadget Γ_v; (ii) without loss of generality, one can assume that at most one of u and v is in S for each edge $uv \in E$.

(i) Suppose for the sake of contradiction that there is some $v \in V$ such that S contains no vertices of Γ_v. Since $G' \setminus S$ is stable, there is a maximum-cardinality fractional c-matching x^*, that is integral. Define for each $e \in E(G' \setminus S)$

$$x_e = \begin{cases} x^*_e & \text{if } e \in E(G' \setminus S) \setminus E[\Gamma_v], \\ 1 & \text{if } e \in \{uv_1 : u \in N^+(v)\}, \\ 0 & \text{if } e = v_1 v_2, \\ 0.5 & \text{if } e \in \{v_2 v_3, v_3 v_4, v_2 v_4\}. \end{cases}$$

Note that x is a fractional c-matching in $G' \setminus S$, since x^* is. Furthermore $\sum_{e \in E[\Gamma_v]} x_e = d^E_v + 2.5$. However, the total capacity that edges in Γ_v can use in any matching is at most $2d^E_v + 5$. So, $\sum_{e \in E[\Gamma_v]} x_e = d^E_v + 2.5 > \sum_{e \in E[\Gamma_v]} x^*_e$, since x^* is integral. Hence, $1^\top x > 1^\top x^*$, contradicting the optimality of x^*.

(ii) Suppose there is some $e = uv \in E$ such that S contains both u and v. All gadgets Γ^i_{uv} are then components in $G' \setminus S$. If u and v are the only vertices in S from some component Γ^i_{uv}, then a maximum-cardinality fractional c-matching in this components is given by $x_{e^i_1 e^i_2} = x_{e^i_1 e^i_3} = 1$ and $x_{e^i_3 e^i_4} = x_{e^i_4 e^i_5} = x_{e^i_3 e^i_5} = 0.5$. Which means this component is not stable, and thus $G' \setminus S$ is not stable, a contradiction. Hence, S must contain at least one vertex of each Γ^i_{uv} that is neither u nor v. Consequently, $k = |S| \geq n + 2$. Since G has only n vertices, it obviously has an independent dominating set of size at most n, and hence of size at most k. Such a set can for example be obtained by a greedy approach. Hence, for the remainder of the proof we can assume that at most one of u and v is in S for each $uv \in E$.

We now create a set $S' \subseteq V$ from S, that is an independent dominating set of G of size at most k, as follows. Iterate over $v \in V$. Let $S_v = S \cap V(\Gamma_v)$. Note that $S_v \neq \emptyset$ by (i). Define

$$S'_v = \begin{cases} (S_v \cup S') \cap N^+(v) & \text{if this is nonempty,} \\ v & \text{otherwise.} \end{cases}$$

Set $S' = S' \cup S'_v$, and repeat for the next vertex.

Clearly, all S'_v's are nonempty, which means that S' contains at least one vertex from $N^+(v)$ for all $v \in V$, which means S' is dominating.

Suppose for the sake of contradiction that S' contains both u and v for some edge $uv \in E$. We know S does not contain both of them, by (ii). If S contains exactly one of them, without loss of generality let it be u. Then, when v is considered by the iterative process, $(S_v \cup S') \cap N^+(v)$ contains u, but not v. In particular, this means that v is not added to S'_v and consequently also not to S', a contradiction. If S contains neither of them, then because it is an iterative process, one of them is added first to S'. Without loss of generality let it be u. Then again, when v is considered by the iterative process, $(S_v \cup S') \cap N^+(v)$ contains u but not v, so we reach a contradiction in the same way. In conclusion, S' is independent.

For all $v \in V$, before we add S'_v to S', we have $|S'_v \setminus S'| \leq |S_v|$. Consequently, $|S'| \leq \cup_{v \in V} |S_v| \leq |S| = k$. □

By this claim, any minimum-cardinality vertex-stabilizer of $(G', 1, c)$ is of the same size as any minimum independent dominating set of G. Further, note that the number of vertices V' of G' is $O(|V|^3)$, and any efficient $O(|V'|)$-approximation algorithm for the vertex-stabilizer problem translates into an efficient $O(|V|^3)$-approximation algorithm for the MIDS problem. Hence, the result follows from the inapproximability of the MIDS problem. □

4.2 Key Polyhedral Tools

Before we can state the capacity- and edge-stabilizer algorithms, we need some new polyhedral tools. First, we have a theorem for a general polytope $\mathcal{P}$, which afterwards we apply to $\mathcal{P}_{\text{FCM}}$.

Theorem 4.3. *Let $\mathcal{P}$ be any polytope, $a^\top x \leq b$ be an inequality of the description of $\mathcal{P}$, and $\delta \in \mathbb{R}_{>0}$. Let $\overline{x}$ be an optimal solution of the LP $\max\{c^\top x : x \in \mathcal{P}, a^\top x \leq b - \delta\}$, such that (i) $\overline{x}$ is a vertex of $\mathcal{P}$, (ii) $\overline{x}$ is not an optimal solution of the LP $\max\{c^\top x : x \in \mathcal{P}\}$, and (iii) there is no vertex $\widetilde{x}$ of $\mathcal{P}$ satisfying $b - \delta < a^\top \widetilde{x} < b$. Then it is possible to move to an optimal solution x^* of $\max\{c^\top x : x \in \mathcal{P}\}$ from $\overline{x}$ in one step over the edges of $\mathcal{P}$ (that is, there is an optimal vertex of $\mathcal{P}$ adjacent to $\overline{x}$). (See Figure 4.3.)*

Proof. Let x^* be the optimal solution of $\max\{c^\top x : x \in \mathcal{P}\}$ that is the *closest* vertex to $\overline{x}$ on $\mathcal{P}$ (that is, such that we need a minimum number of steps over the edges of $\mathcal{P}$ to reach x^* from $\overline{x}$). Note that $a^\top \overline{x} = b - \delta$ and $a^\top x^* = b$,

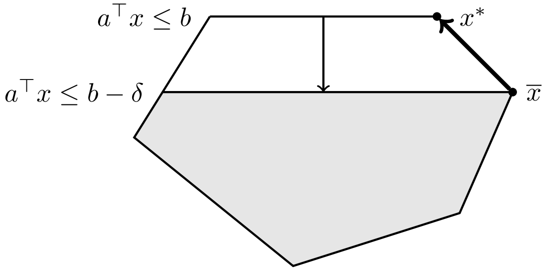


Figure 4.3: Example of the situation described in Theorem 4.3.

otherwise $\overline{x} + \lambda(x^* - \overline{x})$, for some small $\lambda > 0$, and x^*, respectively, contradict the optimality of $\overline{x}$. We need to show that $\overline{x}$ and x^* are adjacent on $\mathcal{P}$.

Let $\mathcal{P}' = \{x \in \mathcal{P} : a^\top x \geq b - \delta\}$. Then $\overline{x}, x^* \in \mathcal{P}'$. Note that $\overline{x}$ and x^* are adjacent on $\mathcal{P}$ if and only if they are adjacent on $\mathcal{P}'$. So for the remainder of the proof we restrict ourselves to $\mathcal{P}'$.

For the sake of contradiction, assume that $\overline{x}$ and x^* are not adjacent on $\mathcal{P}'$. Then, the line segment of all their convex combinations, $\lambda\overline{x} + (1-\lambda)x^*$ for $0 \leq \lambda \leq 1$, is not an edge of $\mathcal{P}'$. Hence, any point $\lambda'\overline{x} + (1-\lambda')x^*$ for a fixed $0 < \lambda' < 1$ is also a convex combination of other vertices of $\mathcal{P}'$: $\lambda'\overline{x} + (1-\lambda')x^* = \sum_i \alpha_i \hat{x}_i + \sum_j \beta_j \widetilde{x}_j$, where $\alpha_i \geq 0$ for all i, $\beta_j \geq 0$ for all j, $\sum_i \alpha_i + \sum_j \beta_j = 1$, $\hat{x}_i$ is a vertex of $\mathcal{P}'$ with $a^\top \hat{x}_i = b - \delta$ for all i, and $\widetilde{x}_j$ is a vertex of $\mathcal{P}'$ with $a^\top \widetilde{x}_j = b$ for all j. If we multiply both sides by a, we get

$$
\begin{aligned}
& & a^\top \left(\lambda'\overline{x} + (1-\lambda')x^*\right) &= a^\top \left(\textstyle\sum_i \alpha_i \hat{x}_i + \sum_j \beta_j \widetilde{x}_j\right), \\
\iff & & \lambda'(b-\delta) + (1-\lambda')b &= \textstyle\sum_i \alpha_i (b-\delta) + \sum_j \beta_j b, \\
\iff & & b - \lambda'\delta &= \left(\textstyle\sum_i \alpha_i + \sum_j \beta_j\right) b - \sum_i \alpha_i \delta,
\end{aligned}
$$

hence, $\lambda' = \sum_i \alpha_i$, and consequently, $1 - \lambda' = \sum_j \beta_j$. We can also multiply both sides by c. Here we use that $\overline{x}$ is an optimal solution of $\max\{c^\top x : x \in \mathcal{P}, a^\top x \leq b - \delta\}$, and that x^* is an optimal solution of $\max\{c^\top x : x \in \mathcal{P}\}$.

$$
\begin{aligned}
c^\top \left(\lambda'\overline{x} + (1-\lambda')x^*\right) = c^\top \left(\textstyle\sum_i \alpha_i \hat{x}_i + \sum_j \beta_j \widetilde{x}_j\right) &= \textstyle\sum_i \alpha_i c^\top \hat{x}_i + \sum_j \beta_j c^\top \widetilde{x}_j \\
&\leq \textstyle\sum_i \alpha_i c^\top \overline{x} + \sum_j \beta_j c^\top x^* = \lambda' c^\top \overline{x} + (1-\lambda') c^\top x^*
\end{aligned}
$$

So we must have equality throughout. In particular, $c^\top \widetilde{x}_j = c^\top x^*$, that is, all $\widetilde{x}_j$ are optimal solutions to $\max\{c^\top x : x \in \mathcal{P}\}$. Note that all $\widetilde{x}_j$'s are also vertices of $\mathcal{P}$. We show that we can choose some $\widetilde{x}_j$ to be adjacent to $\overline{x}$ on $\mathcal{P}'$, and hence, also on $\mathcal{P}$, contradicting that x^* is the optimal solution closest to x.

Let x' be a vertex of $\mathcal{P}'$ that is adjacent to $\overline{x}$, such that $ax' = b$ (such an x' must exist). Consider the line segment between x' and $\lambda'\overline{x} + (1-\lambda')x^*$: $\mu x' + (1-\mu)(\lambda'\overline{x} + (1-\lambda')x^*)$ for $0 \leq \mu \leq 1$. For $\mu < 0$, this line segment extends beyond $\lambda'\overline{x} + (1-\lambda')x^*$. If this line for $\mu < 0$ is still in $\mathcal{P}'$, then we can write $\lambda'\overline{x} + (1-\lambda')x^*$ as a convex combination of x' and some other $\hat{x}_i$'s and $\widetilde{x}_j$'s. Since $ax' = b$, by our previous discussion we find that x' is optimal, reaching our desired contradiction. Otherwise, $\lambda'\overline{x}+(1-\lambda')x^*$ must be at the boundary, a face, of $\mathcal{P}'$. Because $\lambda'\overline{x}+(1-\lambda')x^*$ is in this face, the whole line segment $\lambda\overline{x}+(1-\lambda)x^*$ for $0 \leq \lambda \leq 1$ must be in this face. We can then repeat the argument, replacing $\mathcal{P}'$ by this face. Since this face has strictly smaller dimension than $\mathcal{P}'$, we either find a contradiction in one of the iterations, or we reach a face of dimension one, that is, an edge of $\mathcal{P}'$. Since this edge contains the whole line segment $\lambda\overline{x} + (1-\lambda)x^*$ for $0 \leq \lambda \leq 1$, the line segment is the edge, a contradiction. □

We make use of this theorem for $\mathcal{P}_{\text{FCM}}$ in two settings: to analyze what happens when we reduce the capacity of a vertex, and when we remove an edge. For the first setting, we have the following.

Theorem 4.4. *Let $\overline{x}$ be a maximum-weight fractional c-matching in $G[c_v-1]$ for some $v \in V$. If $\overline{x}$ is basic in G, but does not have maximum weight in G, then it is possible to move to a basic maximum-weight fractional c-matching in G in one step over the edges of $\mathcal{P}_{FCM}(G,c)$.*

Proof. Let $\mathcal{P} = \mathcal{P}_{\text{FCM}}(G,c)$, $a^\top x \leq b$ be $x(\delta(v)) \leq c_v$, $\delta = 1$, and w be the objective function. It follows from Theorem 2.2 that $x(\delta(v))$ is integral for all basic fractional c-matchings. Consequently, there are no vertices $\widetilde{x}$ of $\mathcal{P}_{\text{FCM}}(G,c)$ that satisfy $c_v - 1 < \widetilde{x}(\delta(v)) < c_v$. The theorem now readily follows from Theorem 4.3. □

In the second setting, we need to do a bit of extra work.

Theorem 4.5. *Let $\overline{x}$ be a maximum-weight fractional c-matching in $G \setminus e$ for some $e \in E$. If $\overline{x}$ is basic in G, but does not have maximum weight in G, then it is possible to move to a basic maximum-weight fractional c-matching in G in at most two steps over the edges of $\mathcal{P}_{FCM}(G,c)$. If two steps are needed, the first one moves to a vertex with $x_e = \frac{1}{2}$, and the second one moves to a vertex with $x_e = 1$.*

Proof. Case 1: $x_e \in \{0,1\}$ for all vertices of $\mathcal{P}_{FCM}(G,c)$. It follows directly from Theorem 4.3 that only one step is needed, by letting $\mathcal{P} = \mathcal{P}_{\text{FCM}}(G,c)$, $a^\top x \leq b$ be $x_e \leq 1$, $\delta = 1$, and w the objective function.

Case 2: there are vertices of $\mathcal{P}_{FCM}(G,c)$ that satisfy $x_e = \frac{1}{2}$. In this case, let us consider two polytopes: $\mathcal{P}^{\leq} = \{x \in \mathcal{P}_{\mathrm{FCM}}(G,c) : x_e \leq \frac{1}{2}\}$ and $\mathcal{P}^{\geq} = \{x \in \mathcal{P}_{\mathrm{FCM}}(G,c) : x_e \geq \frac{1}{2}\}$. Let $\overline{x}$ be a maximum-weight fractional c-matching in $G \setminus e$, such that $\overline{x}$ is basic in G, but $\overline{x}$ does not have maximum weight in G. Since $\overline{x}$ does not have maximum weight over $\mathcal{P}_{\mathrm{FCM}}(G,c)$, there is an improving direction at $\overline{x}$ in $\mathcal{P}_{\mathrm{FCM}}(G,c)$. Moving in this direction a bit, we obtain a fractional c-matching with larger weight than $\overline{x}$, which is in $\mathcal{P}^{\leq}$. Hence, $\overline{x}$ does not have maximum weight over $\mathcal{P}^{\leq}$. In addition, since $\overline{x}$ is a vertex of $\mathcal{P}_{\mathrm{FCM}}(G,c)$, and feasible in $\mathcal{P}^{\leq}$, it is a vertex of $\mathcal{P}^{\leq}$.

Let $\mathcal{P} = \mathcal{P}^{\leq}$, $ax \leq b$ be $x_e \leq \frac{1}{2}$, $\delta = \frac{1}{2}$, and w be the objective function. We can then apply Theorem 4.3: it is possible to move to an optimal solution $\hat{x}$ of $\max\{w^\top x : x \in \mathcal{P}^{\leq}\}$ from $\overline{x}$ in one step over the edges of $\mathcal{P}^{\leq}$. Note that $\hat{x}_e = \frac{1}{2}$. If $\hat{x}$ is a basic maximum-weight fractional c-matching in G, we are done. So suppose that that is not the case.

Subcase 2a: $\hat{x}$ is a vertex of $\mathcal{P}_{FCM}(G,c)$. First, note that since $\hat{x}$ is a vertex of $\mathcal{P}_{\mathrm{FCM}}(G,c)$, then the edge of $\mathcal{P}^{\leq}$ that is used to move from $\overline{x}$ to $\hat{x}$, is also an edge of $\mathcal{P}_{\mathrm{FCM}}(G,c)$. Furthermore, since $\hat{x}$ is feasible in $\mathcal{P}^{\geq}$, it is also a vertex in $\mathcal{P}^{\geq}$. By assumption, $\hat{x}$ is optimal over $\mathcal{P}^{\leq}$, so also over $\mathcal{P}^{\geq}$ with the additional constraint $x_e \leq \frac{1}{2}$, but not optimal over $\mathcal{P}_{\mathrm{FCM}}(G,c)$, so also not over $\mathcal{P}^{\geq}$. Let $\mathcal{P} = \mathcal{P}^{\geq}$, $ax \leq b$ be $x_e \leq 1$, $\delta = \frac{1}{2}$, and w the objective function. We can then apply Theorem 4.3: it is possible to move to an optimal solution x^* of $\max\{w^\top x : x \in \mathcal{P}^{\geq}\}$ from $\hat{x}$ in one step over the edges of $\mathcal{P}^{\geq}$. Note that $x^*_e = 1$. Then, x^* is also an optimal solution of $\max\{w^\top x : x \in \mathcal{P}_{\mathrm{FCM}}(G,c)\}$, and a vertex of $\mathcal{P}_{\mathrm{FCM}}(G,c)$. Since $\hat{x}$ and x^* are both vertices of $\mathcal{P}_{\mathrm{FCM}}(G,c)$, the edge of $\mathcal{P}^{\geq}$ that is used, is also an edge of $\mathcal{P}_{\mathrm{FCM}}(G,c)$. All in all, we get that, starting from $\overline{x}$, it is possible to move to a basic maximum-weight fractional c-matching in G in two steps over the edges of $\mathcal{P}_{\mathrm{FCM}}(G,c)$, such that $x_e = \frac{1}{2}$ after the first step, and $x_e = 1$ after the second step.

Subcase 2b: $\hat{x}$ is not a vertex of $\mathcal{P}_{FCM}(G,c)$. In this case, we moved from $\overline{x}$ to $\hat{x}$ over an edge of $\mathcal{P}^{\leq}$ which is strictly contained in an edge of $\mathcal{P}_{\mathrm{FCM}}(G,c)$: it must therefore be that $\mathcal{P}^{\leq}$ and $\mathcal{P}^{\geq}$ split this edge in two, and the splitting point, $\hat{x}$, is a vertex of both polytopes. By assumption, $\hat{x}$ is optimal over $\mathcal{P}^{\leq}$, so also over $\mathcal{P}^{\geq}$ with the additional constraint $x_e \leq \frac{1}{2}$. Since we reached $\hat{x}$ by moving over just part of an edge of $\mathcal{P}_{\mathrm{FCM}}(G,c)$, and this increased the weight, moving further along this edge will increase the weight even further. Hence, $\hat{x}$ is not optimal over $\mathcal{P}^{\geq}$. Let $\mathcal{P} = \mathcal{P}^{\geq}$, $ax \leq b$ be $x_e \leq 1$, $\delta = \frac{1}{2}$, and w the objective function. We can again apply Theorem 4.3: it is possible to move to an optimal solution x^* of $\max\{w^\top x : x \in \mathcal{P}^{\geq}\}$ from $\hat{x}$ in one step over the edges of $\mathcal{P}^{\geq}$. Note that $x^*_e = 1$. Then, x^* is also an optimal solution of $\max\{w^\top x : x \in \mathcal{P}_{\mathrm{FCM}}(G,c)\}$, and a vertex of $\mathcal{P}_{\mathrm{FCM}}(G,c)$. Since x^* is a vertex of $\mathcal{P}_{\mathrm{FCM}}(G,c)$, but $\hat{x}$ is not, the edge of $\mathcal{P}^{\geq}$ that is used, is only part of an edge of $\mathcal{P}_{\mathrm{FCM}}(G,c)$. In particular, it must be the remainder of the edge over which we moved in the first step. All in all, we get that, starting from $\overline{x}$, it is possible to move to a basic maximum-weight fractional c-matching in G in one step over the edges of $\mathcal{P}_{\mathrm{FCM}}(G, c)$. $\square$

The following theorem describes the relation between adjacent vertices on $\mathcal{P}_{\mathrm{FCM}}$. The theorem is based on methods used in Section III-G in Sanità [46].

Theorem 4.6. *If x and y are adjacent vertices of $\mathcal{P}_{FCM}(G, c)$, then $y = x + \alpha g$, where $g \in \mathcal{C}(\mathcal{P}_{FCM})$ and $\alpha \in \left\{\frac{1}{2}, 1\right\}$. Furthermore,*

- *if $\alpha = 1$, then $g \in \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3$ and $|\mathscr{C}_y| = |\mathscr{C}_x|$.*
- *if $\alpha = \frac{1}{2}$, then $g \in \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_4 \cup \mathcal{C}_5$, and*
  - *if $g \in \mathcal{C}_1$, then $|\mathscr{C}_y| = |\mathscr{C}_x|$.*
  - *if $g \in \mathcal{C}_2 \cup \mathcal{C}_4$, then $|\mathscr{C}_y| = |\mathscr{C}_x| \pm 1$, and the odd cycle in g belongs to either $\mathscr{C}_x$ or $\mathscr{C}_y$.*
  - *if $g \in \mathcal{C}_5$, then $|\mathscr{C}_y| = |\mathscr{C}_x| \pm \{0, 2\}$, and the odd cycles in g both belong to $\mathscr{C}_x$, or both to $\mathscr{C}_y$, or exactly one belongs to $\mathscr{C}_x$ and the other to $\mathscr{C}_y$.*

Before we go into the proof, we introduce a fractional perfect c-matching polytope, which will be helpful. Consider $\mathcal{P}_{\mathrm{FCM}}$. Add a nonnegative slack variable for each inequality of the form $x(\delta(v)) \leq c_v$. We get a polytope that naturally corresponds to the set of fractional perfect c-matchings on a modified graph $\overline{G} = (V, E \cup L)$, obtained from G by adding a loop edge $uv \in L$ for each vertex $v \in V$. We define

$$\mathcal{P}_{\mathrm{FPCM}}(\overline{G}, c) = \left\{x \in \mathbb{R}^{E \cup L} : x(\delta(v)) = c_v \ \forall v \in V, x \geq 0, x_e \leq 1 \ \forall e \in E\right\},$$

as the polytope of fractional perfect c-matchings in $\overline{G}$.

Proof of Theorem 4.6. Let g be the edge direction of the edge between x and y, scaled in such a way that the components of g are co-prime. Then, clearly, $y = x + \alpha g$ for some $\alpha \neq 0$. Without loss of generality, we can assume that $\alpha > 0$, since $-g$ is also an edge direction of the same edge. All edge directions are circuits, hence $g \in \mathcal{C}(\mathcal{P}_{\mathrm{FCM}})$, and in particular, $g \in \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3 \cup \mathcal{C}_4 \cup \mathcal{C}_5$ by Proposition 2.2. That means that the components of g have a magnitude of at least 1. Then it follows from $0 \leq x \leq 1$, that $\alpha \leq 1$. Likewise, for the circuits with components that have a magnitude of 2, it follows that $\alpha \leq \frac{1}{2}$. Finally, since x and y are vertices, that is, they are basic, their components are equal to 0, $\frac{1}{2}$, or 1, which implies $\alpha \in \left\{\frac{1}{2}, 1\right\}$.

Case 1: $\alpha = 1$. As discussed, for circuits with components that have a magnitude of 2, $\alpha \leq \frac{1}{2}$. Hence, in this case, $g \in \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3$. Furthermore, all components of αg are integral, which means that fractional edges, and in particular the fractional odd cycles, are not affected. It follows that $|\mathscr{C}_y| = |\mathscr{C}_x|$.

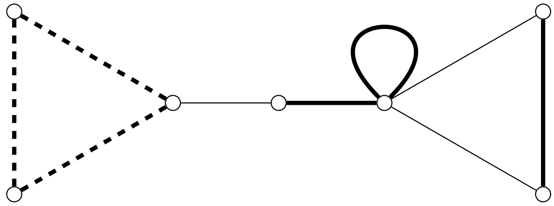
Case 2: $\alpha = \frac{1}{2}$. Circuits in $\mathcal{C}_3$ correspond with paths. For either endpoint of this path, applying the circuit, that is, adding αg to a fractional c-matching, results in $\alpha \cdot \pm 1 = \pm\frac{1}{2}$ on a single edge incident with the vertex. Since x and y are both basic, this is not possible, so $g \in \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_4 \cup \mathcal{C}_5$.

We extend x and y to fractional perfect c-matchings $\overline{x}$ and $\overline{y}$ in $\overline{G}$. This extension is uniquely obtained by setting $\overline{x}_{vv} = c_v - x(\delta(v))$ for each $uv \in L$, and likewise for $\overline{y}$. Since x and y are adjacent vertices of $\mathcal{P}_{\mathrm{FCM}}(G, c)$, it follows that $\overline{x}$ and $\overline{y}$ are adjacent vertices of $\mathcal{P}_{\mathrm{FPCM}}(\overline{G}, c)$. We extend g to $\overline{g}$ such that $\overline{y} = \overline{x} + \frac{1}{2}\overline{g}$.

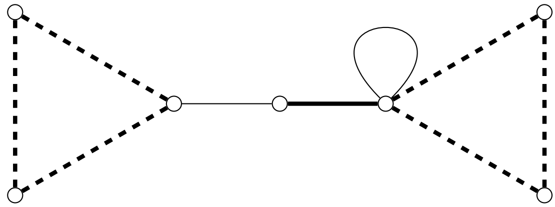
Let $E^1 = \{e \in E : \overline{x}_e = \overline{y}_e = 1\}$, and $\mathcal{G}$ be the graph induced by the supports of $\overline{x}$ and $\overline{y}$, minus the edges in E^1. See Figure 4.4 for an example of $\overline{x}$, $\overline{y}$, $\overline{g}$, and the graph $\mathcal{G}$ obtained from them. We claim that there is exactly one component of $\mathcal{G}$ that contains an edge e with $\overline{x}_e \neq \overline{y}_e$. Clearly there is at least one, since $x \neq y$ and hence $\overline{x} \neq \overline{y}$. Actually, $\overline{x} \neq \overline{y}$ only on the support of $\overline{g}$. Since $\overline{x}_e \neq \overline{y}_e$ for every edge e in the support of $\overline{g}$, we have that at least one of $\overline{x}_e$ and $\overline{y}_e$ is not zero, and at least one is not one. Hence, all those edges are in $\mathcal{G}$, and in particular they all are in the same component, since $\overline{g}$ is connected.

Let $\mathcal{K}$ be the component of $\mathcal{G}$ that contains an edge e with $\overline{x}_e \neq \overline{y}_e$, and let k be the number of vertices in this component. Let K be a subgraph of $\overline{G}$ induced by the vertices in $\mathcal{K}$, minus the edges in E^1. We change the capacities accordingly: let $c|_K$ be obtained from c by restricting to the vertices in K, and, for each vertex $v \in V(K)$, reducing the capacity of v by $|\delta(v) \cap E^1|$. Let $\overline{x}|_K$ be obtained from $\overline{x}$ by restricting to the edges in K, and likewise for $\overline{y}|_K$. Note that $\overline{x}|_K$ and $\overline{y}|_K$ are perfect c-matchings in K with respect to $c|_K$. In particular, they are adjacent vertices of $\mathcal{P}_{\mathrm{FPCM}}(K, c|_K)$, since $\overline{x}$ and $\overline{y}$ are adjacent vertices of $\mathcal{P}_{\mathrm{FPCM}}(\overline{G, c})$.

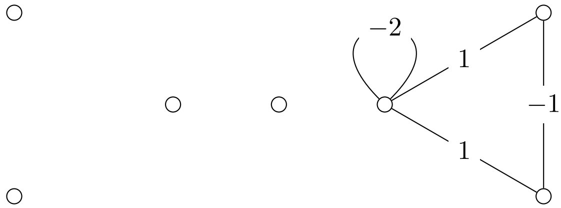
Let A be the incidence matrix of K. Since the columns associated to the loop edges form an identity matrix, the rank of A is k. Since $\overline{x}|_K$ and $\overline{y}|_K$ are adjacent vertices, there must be $|E(K)| - 1$ linearly independent constraints that are tight for both of them. Since the rank of A is k, and we removed the edges for which the "≤ 1" constraint is tight for both $\overline{x}$ and $\overline{y}$, this implies that there are at least $|E(K)| - 1 - k$ edges for which the "≥ 0" constraint is tight for both of them. Consequently, there are at most $k + 1$ edges in the union of the supports of $\overline{x}|_K$ and $\overline{y}|_K$. Note that the graph induced by the supports of $\overline{x}|_K$ and $\overline{y}|_K$ is $\mathcal{K}$. With $k + 1$ edges on k connected vertices, we have a spanning tree plus two additional (possibly loop) edges: it is easy to realize then that there can be at most two odd cycles in $\mathcal{K}$.



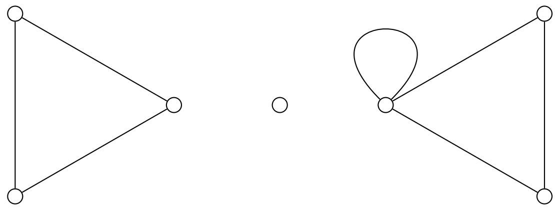
(a) Example of $\overline{x}$, with $\overline{x}_e = 1$ for bold edges (and the loop), $\overline{x}_e = \frac{1}{2}$ for dashed edges, and $\overline{x}_e = 0$ for normal edges.



(b) Example of $\overline{y}$, with $\overline{y}_e = 1$ for bold edges, $\overline{y}_e = \frac{1}{2}$ for dashed edges, and $\overline{y}_e = 0$ for normal edges (and the loop).



(c) The circuit $\overline{g}$ such that $\overline{y} = \overline{x} + \frac{1}{2}\overline{g}$ is given on the edges ($\overline{g}$ is the extension of a circuit $g \in \mathcal{C}_2$).



(d) Example of $\mathcal{G}$. The component on the right is the unique component that contains an edge e with $\overline{x}_e \neq \overline{y}_e$.

Figure 4.4: Example of a graph G with unit-weights and unit-capacities, except for the vertex that has a loop edge, which has a capacity of two. The other loops are not drawn, as they are not relevant for the example. Figure 4.4a shows $\overline{x}$, Figure 4.4b shows $\overline{y}$, Figure 4.4c shows $\overline{g}$, and Figure 4.4d shows $\mathcal{G}$.

Subcase 2a: $g \in \mathcal{C}_1$. The support of $\overline{g}$ is an even cycle, say C. If $\overline{x}|_K = \frac{1}{2}$ for all edges on C, then $\overline{x}|_K$, and therefore also x, contains a fractional even cycle, which contradicts that x is basic. Similarly, if $\overline{x}|_K$ is integral for all edges on C, $\overline{y}|_K = \frac{1}{2}$ for all edges on C, contradicting that y is basic. Hence, $\overline{x}|_K$ has edges on C with integral value, and also edges with value $\frac{1}{2}$. The fractional edges imply that $\overline{x}|_K$ has at least one fractional odd cycle. The integral edges become fractional in $\overline{y}|_K$, which means $\overline{y}|_K$ also has at least one fractional odd cycle, different from the one of $\overline{x}|_K$. These odd cycles are distinct, and both in $\mathcal{K}$, and we have already shown that $\mathcal{K}$ contains at most two odd cycles. Hence, $\left|\mathscr{C}_{\overline{y}|_K}\right| = \left|\mathscr{C}_{\overline{x}|_K}\right|$.

Subcase 2b: $g \in \mathcal{C}_2 \cup \mathcal{C}_4$. The support of $\overline{g}$ is a (possibly empty) path, an odd cycle and a loop edge, of which only the odd cycle can influence the fractional odd cycles in $\overline{x}|_K$ and $\overline{y}|_K$. If this odd cycle in the support of $\overline{g}$ is a fractional odd cycle in $\overline{x}|_K/\overline{y}|_K$, then note that $\overline{y}|_K$ contains exactly one less/more fractional odd cycle than $\overline{x}|_K$. Otherwise, both $\overline{x}|_K$ and $\overline{y}|_K$ have fractional and integral edges on the odd cycle of g. That means that $\overline{x}|_K$ and $\overline{y}|_K$ both have at least one odd cycle in the component, different from the odd cycle in g, and different from each other. But then there are at least three odd cycles in the component, a contradiction. Hence, the odd cycle in g belongs to either $\mathscr{C}_{\overline{x}|_K}$ or $\mathscr{C}_{\overline{y}|_K}$, and $\left|\mathscr{C}_{\overline{y}|_K}\right| = \left|\mathscr{C}_{\overline{x}|_K}\right| \pm 1$.

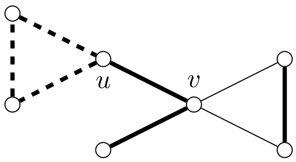
Subcase 2c: $g \in \mathcal{C}_5$. The support of $\overline{g}$ is two odd cycles connected by a (possibly empty) path. Since $\mathcal{K}$ contains at most two odd cycles, and $\overline{g}$ already contains two odd cycles, these are the only odd cycles. Similar to the previous subcase, for both the odd cycles separately we can argue that not both $\overline{x}|_K$ and $\overline{y}|_K$ can have fractional and integral edges on the odd cycle, that is, each odd cycle belongs to either $\mathscr{C}_{\overline{x}|_K}$ or $\mathscr{C}_{\overline{y}|_K}$. There are three options: both odd cycles belong to $\mathscr{C}_{\overline{x}|_K}$, or both to $\mathscr{C}_{\overline{y}|_K}$, or one to $\mathscr{C}_{\overline{x}|_K}$ and one to $\mathscr{C}_{\overline{y}|_K}$. It follows that $\left|\mathscr{C}_{\overline{y}|_K}\right| = \left|\mathscr{C}_{\overline{x}|_K}\right| \pm \{0, 2\}$.

Since $\overline{x}$ equals $\overline{y}$ outside of K, our conclusions carry over from $\overline{x}|_K$ and $\overline{y}|_K$ to $\overline{x}$ and $\overline{y}$. In addition, removing loop edges, that is, going back from $\overline{x}, \overline{y}$ to x, y, does not affect fractional odd cycles. Hence, our conclusions also hold for x and y: If $g \in \mathcal{C}_1$, then $|\mathscr{C}_y| = |\mathscr{C}_x|$. If $g \in \mathcal{C}_2 \cup \mathcal{C}_4$, then $|\mathscr{C}_y| = |\mathscr{C}_x| \pm 1$ and the odd cycle in g belongs to either $\mathscr{C}_x$ or $\mathscr{C}_y$. If $g \in \mathcal{C}_5$, then $|\mathscr{C}_y| = |\mathscr{C}_x| \pm \{0, 2\}$, and the odd cycles in g both belong to $\mathscr{C}_x$, or both to $\mathscr{C}_y$, or exactly one to $\mathscr{C}_x$ and the other to $\mathscr{C}_y$. $\square$

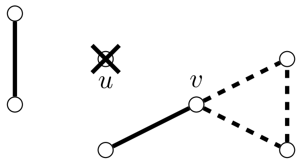
4.3 Capacity-Stabilizer

Our algorithm for the capacity-stabilizer problem is based on the unit-capacity vertex-stabilizer algorithm of Koh and Sanità [36]. So before we state our algorithm, we explain the idea of their algorithm.

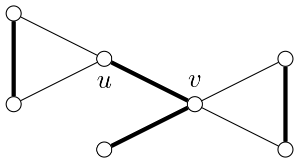
Recall from Proposition 3.1 that a graph is stable if and only if there are no fractional odd cycles. The vertex-stabilizer algorithm of Koh and Sanità [36] is based on this: compute a basic maximum-weight fractional matching with the minimum number of fractional odd cycles ($\gamma(G)$), and remove one vertex from every fractional odd cycle. This same approach does not work in the capacitated case: if we remove a vertex from a fractional odd cycle, even though that specific fractional odd cycle is removed, another one might be created (see Figure 4.5). Interestingly, if instead of removing the vertex u in Figure 4.5, we reduce its capacity from two to one, we do obtain a stable graph. Our results confirm what this example suggests: As we have seen in Section 4.1, the vertex-stabilizer problem is APX-hard in capacitated instances, so there is no polynomial-time exact algorithm, unless $P = NP$. And as we will discuss now, the algorithm of Koh and Sanità [36] generalizes to capacitated graphs when reducing the capacity of vertices.



(a) The graph G with a basic maximum-weight fractional c-matching x with $|\mathscr{C}_x| = \gamma(G) = 1$.



(b) The graph $G \setminus u$ with a basic maximum-weight fractional c-matching x with $|\mathscr{C}_x| = \gamma(G \setminus u) = 1$.



(c) The graph $G[c_u - 1]$ with a basic maximum-weight fractional c-matching x with $|\mathscr{C}_x| = \gamma(G[c_u - 1]) = 0$.

Figure 4.5: A graph $(G, 1, c)$ with $c_u = c_v = 2$ and $c = 1$ otherwise. Bold lines indicate edges with $x_e = 1$, bold dashed lines edges with $x_e = \frac{1}{2}$ and normal lines edges with $x_e = 0$. Figure 4.5b shows that removing a vertex from a fractional odd cycle might create a new fractional odd cycle, and Figure 4.5c shows that instead reducing the capacity of that same vertex stabilizes the graph.

We first exploit the polyhedral results from Section 4.2 to prove that a lower bound on the size of a capacity-stabilizer is given by the minimum number of fractional odd cycles in the support of any basic maximum-weight fractional c-matching.

Lemma 4.1. *For every capacity-stabilizer S, $|S| \geq \gamma(G)$.*

Proof. To prove the lemma, by Proposition 3.1, it is enough to show that reducing the capacity of any vertex by one decreases the number of fractional odd cycles by at most one. Therefore, from now on, we concentrate on proving the following statement:

$$\text{for all } v \in V,\ \gamma(G[c_v - 1]) \geq \gamma(G) - 1. \tag{4.1}$$

Let x be a basic maximum-weight fractional c-matching in $G[c_v - 1]$ with $\gamma(G[c_v - 1])$ fractional odd cycles. Let (y, z) be an optimal fractional vertex cover in $G[c_v - 1]$, satisfying complementary slackness (see Equation (2.2)) with x. Note that increasing the capacity does not influence the feasibility of x and (y, z), hence they are a fractional c-matching and vertex cover in G, respectively.

If x has maximum weight in G, then $\gamma(G) \leq |\mathscr{C}_x| = \gamma(G[c_v - 1])$, and hence (4.1) holds.

Assume now that x does not have maximum weight in G. Then, x and (y, z) cannot satisfy complementary slackness in G. The change from $G[c_v - 1]$ to G only influences the complementary slackness condition $y_v = 0 \vee x(\delta(v)) = c_v - 1$, so we must have $y_v > 0$ and $x(\delta(v)) = c_v - 1 < c_v$. We distinguish two cases.

Case 1: $v \in V(C)$ for some $C \in \mathscr{C}_x$. Create a new fractional c-matching $\hat{x}$ by alternate rounding C covering v. One can check that $\hat{x}$ and (y, z) satisfy complementary slackness in G. Consequently, $\hat{x}$ is a maximum-weight fractional c-matching in G, and given that x is basic, so is $\hat{x}$. Then, $\gamma(G) \leq |\mathscr{C}_{\hat{x}}| = |\mathscr{C}_x| - 1 = \gamma(G[c_v - 1]) - 1$, and hence (4.1) holds.

Case 2: $v \notin V(\mathscr{C}_x)$. Since x is basic in $G[c_v - 1]$ and v is not part of any fractional odd cycle, by Theorem 2.2, x is also basic in G. Then, by Theorem 4.4, we can move to a basic maximum-weight fractional c-matching x^* in G in one step over the edges of $\mathcal{P}_{\text{FCM}}(G, c)$. By Theorem 4.6, $x^* = x + \alpha g$ for $\alpha \in \left\{\frac{1}{2}, 1\right\}$ and $g \in \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3 \cup \mathcal{C}_4 \cup \mathcal{C}_5$. Because $w^\top x^* > w^\top x$, x^* cannot be feasible in $G[c_v - 1]$, so we must have $x^*(\delta(v)) = c_v = x(\delta(v)) + 1$. Consequently, $g \in \mathcal{C}_2 \cup \mathcal{C}_3 \cup \mathcal{C}_4$. Then, by Theorem 4.6, we have $|\mathscr{C}_{x^*}| = |\mathscr{C}_x| \pm \{0, 1\}$. Therefore $|\mathscr{C}_{x^*}| \leq |\mathscr{C}_x| + 1$, and consequently, $\gamma(G) \leq |\mathscr{C}_{x^*}| \leq |\mathscr{C}_x| + 1 = \gamma(G[c_v - 1]) + 1$, yielding (4.1). $\square$

We can now state the polynomial-time algorithm to solve the stabilization problem via capacity reduction.

Algorithm 1 stabilization by capacity reduction

1: initialize $S \leftarrow \emptyset$
2: compute a basic maximum-weight fractional c-matching x in G with $\gamma(G)$ fractional odd cycles $C_1, \ldots, C_{\gamma(G)}$, and a minimum fractional vertex cover (y, z) in G
3: **for** $i = 1$ to $\gamma(G)$ **do**
4: $\quad S \leftarrow S + \arg\min_{v \in V(C_i)} y_v$
5: **end for**
6: **return** $G[c_S - 1]$

Theorem 4.7. *Algorithm 1 is a polynomial-time algorithm that computes a minimum capacity-stabilizer S for G. Moreover,*

(a) The solution S reduces the capacity of each vertex by at most one unit.

(b) The solution S preserves the weight of a maximum-weight matching by a factor of $\frac{2}{3}$, that is, $\nu^c(G[c_S - 1]) \geq \frac{2}{3}\nu^c(G)$.

Proof. Let $S = \{v_1, \ldots, v_{\gamma(G)}\}$ be the set of vertices whose capacity is reduced in Algorithm 1. Let $\hat{x}$ be obtained from x by alternate rounding C_i exposing v_i, for all $i \in \{1, \ldots, \gamma(G)\}$. Clearly, $\hat{x}$ is a fractional c-matching in $G[c_S - 1]$. In addition, (y, z) is still a fractional vertex cover in $G[c_S - 1]$. One can check that they satisfy complementary slackness with respect to $G[c_S - 1]$. Hence, they are optimal in $G[c_S - 1]$. Note that $\hat{x}$ is an integral matching. Hence, $G[c_S - 1]$ is stable. Moreover, $|S| = \gamma(G)$, which is minimum by Lemma 4.1.

Since all cycles in $\mathscr{C}_x$ are vertex-disjoint, the set S is *not* a multiset. Hence, (a) holds. To see (b), note that, since $v_i = \arg\min_{v \in V(C_i)} y_v$, we have

$$y_{v_i} \leq \frac{y(V(C_i))}{|C_i|} \leq \tfrac{1}{3} y(V(C_i)).$$

Then, using stability of $G[c_S - 1]$ and optimality of (y, z) in $G[c_S - 1]$, we find

$$\begin{aligned}
\nu^c(G[c_S - 1]) = \tau_f^c(G[c_S - 1]) &= (c^{S-1})^\top y + 1^\top z \\
&= c^\top y - \sum_{i=1}^{\gamma(G)} y_{v_i} + 1^\top z \geq c^\top y - \sum_{i=1}^{\gamma(G)} \tfrac{1}{3} y(V(C_i)) + 1^\top z \\
&\geq \tfrac{2}{3}(c^\top y + 1^\top z) = \tfrac{2}{3}\tau_f^c(G) \geq \tfrac{2}{3}\nu^c(G).
\end{aligned}$$

In the above chain of inequalities, we use the fact that $c_v - \frac{1}{3} \geq \frac{2}{3} c_v$ for all $v \in V(\mathscr{C}_x)$. This is true since for all these vertices $c_v \geq 1$. $\square$

The previous theorem shows that there always exists a capacity-stabilizer of minimum size that preserves the total value that the players can get by a factor of $\frac{2}{3}$. We note that, for arbitrary weighted graphs, this factor is asymptotically best possible, as shown by Koh and Sanità [36] already in the unit-capacity case. However, for unit-capacities *and* unit-weights, Ahmadian et al. [1] proved a stronger statement: namely, that inclusion-wise minimal stabilizers completely preserve the total value that the players can get (that is, up to a factor of 1). Using our polyhedral tools, we can show that this statement still holds in the capacitated setting (and note that it is satisfied by the solution provided by our algorithm).

Theorem 4.8. *In $(G, 1, c)$, for any inclusion-wise minimal capacity-stabilizer S, we have $\nu^c(G[c_S - 1]) = \nu^c(G)$.*

Proof. Let M be a maximum-cardinality c-matching in $G[c_S - 1]$.

Claim 4.4. *M is maximum in $G[c_{S\setminus v} - 1]$, for any $v \in S$.*

Proof. For the sake of contradiction, suppose that $|M| < \nu^c(G[c_{S\setminus v} - 1])$.

Since S is a stabilizer, $G[c_S - 1]$ is stable, and hence there exists a fractional vertex cover (y, z) that satisfies complementary slackness with M in $G[c_S - 1]$. Increasing the capacity of a vertex does not change feasibility of (y, z), hence, (y, z) is a fractional vertex cover in $G[c_{S\setminus v} - 1]$. Observe that, since the edges have unit-weight, we can assume without loss of generality that $y \leq 1$. Then

$$\tau_f^c(G[c_{S\setminus v} - 1]) \leq (c^{S-1})^\top y + y_v + 1^\top z = |M| + y_v \leq |M| + 1 \leq \nu^c(G[c_{S\setminus v} - 1]),$$

that is, $G[c_{S\setminus v} - 1]$ is stable, contradicting the minimality of S. $\square$

Claim 4.5. *M is maximum in $G[c_{S\setminus\{u,v\}} - 1]$, for any $u, v \in S$.*

Proof. For the sake of contradiction, suppose that $|M| < \nu^c(G[c_{S\setminus\{u,v\}} - 1])$.

Let x be the indicator vector of M, then by Theorem 2.2, x is basic. Since S is inclusion-wise minimal, $G[c_{S\setminus v} - 1]$ is not stable, and thus x is not a maximum fractional c-matching in $G[c_{S\setminus v} - 1]$. We can apply Theorem 4.4 to x, $G[c_S - 1]$ and $G[c_{S\setminus v} - 1]$, and conclude that there exists a basic maximum-weight fractional c-matching $\hat{x}$ in $G[c_{S\setminus v} - 1]$, which is adjacent to x on $\mathcal{P}_{\mathrm{FCM}}$. By Theorem 4.6, $\hat{x} = x + \alpha g$, where $\alpha \in \{\frac{1}{2}, 1\}$ and $g \in \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3 \cup \mathcal{C}_4 \cup \mathcal{C}_5$. Since M is maximum in $G[c_{S\setminus v} - 1]$ by the previous claim, $\hat{x}$ cannot be integral, so $\alpha = \frac{1}{2}$, and consequently, by Theorem 4.6, $g \notin \mathcal{C}_3$. Furthermore, the circuits in $\mathcal{C}_1 \cup \mathcal{C}_5$ are not augmenting in cardinality, so $g \in \mathcal{C}_2 \cup \mathcal{C}_4$, and necessarily v must be the only vertex with $g(\delta(v)) \neq 0$. Observing now that $|M|$ and

$\nu^c(\cdot)$ are integral, we find that $\hat{x}$ is not a maximum fractional c-matching in $G[c_{S\setminus\{u,v\}} - 1]$, since

$$1^\top \hat{x} = 1^\top x + \tfrac{1}{2} = |M| + \tfrac{1}{2} < \nu^c(G[c_{S\setminus\{u,v\}} - 1]) \leq \nu^c_f(G[c_{S\setminus\{u,v\}} - 1]).$$

Applying Theorem 4.4 again, we get that there exists a basic maximum-weight fractional c-matching x^* in $G[c_{S\setminus\{u,v\}} - 1]$, which is adjacent to $\hat{x}$ on $\mathcal{P}_{\mathrm{FCM}}$. By Theorem 4.6, $x^* = \hat{x} + \beta h$, where $\beta \in \{\frac{1}{2}, 1\}$ and $h \in \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3 \cup \mathcal{C}_4 \cup \mathcal{C}_5$. As before, the circuits in $\mathcal{C}_1 \cup \mathcal{C}_5$ are not augmenting in cardinality, so $h \in \mathcal{C}_2 \cup \mathcal{C}_3 \cup \mathcal{C}_4$. Since S is an inclusion-wise minimal stabilizer, $G[c_{S\setminus\{u,v\}} - 1]$ is not stable, and consequently, $1^\top x^* > \nu^c(G[c_{S\setminus\{u,v\}} - 1])$. Using integrality of $\nu^c(\cdot)$ and half-integrality of x^* and $\hat{x}$, this implies $1^\top x^* \geq 1^\top \hat{x} + 1$. So we must have $\beta = 1$, hence, $h \in \mathcal{C}_2 \cup \mathcal{C}_3$ by Theorem 4.6, and $1^\top x^* = 1^\top \hat{x} + 1$. Furthermore, u must be one of the (possibly two) vertices with $h(\delta(u)) > 0$.

Assume first that $h \in \mathcal{C}_2$. Then, u is the only vertex with $h(\delta(u)) > 0$. However, as components of βh have a magnitude of one, it means that u is not saturated in $G[c_{S\setminus v} - 1]$. Therefore, $\hat{x} + \frac{1}{2}h$ is a fractional c-matching in $G[c_{S\setminus v} - 1]$ with higher objective value than $\hat{x}$, contradicting its optimality.

We are left with $h \in \mathcal{C}_3$. It follows that the support of h is a path P_h with endpoints u and some vertex $t \neq u$. Note that, necessarily, $t \neq v$, as v is saturated in $\hat{x}$, while t is not. In particular, neither u nor t are saturated in $G[c_{S\setminus u} - 1]$. However, we know that $x + \beta h$ cannot be a fractional c-matching in $G[c_{S\setminus u} - 1]$, because M is of maximum cardinality in $G[c_{S\setminus u} - 1]$ by the previous claim. Since $x + \beta h$ does not violate any capacity bound in $G[c_{S\setminus u} - 1]$, the reason why $x + \beta h$ is not a fractional c-matching must be the fact that either $0 \leq x + \beta h$ or $x + \beta h \leq 1$ does *not* hold. Since instead $0 \leq x + \alpha g + \beta h \leq 1$ holds, it follows that the supports of h and g must share some edge. Note that since all components of βh have a magnitude of one, the support of h cannot overlap with the cycle in the support of g. Let ℓ be the last vertex on the ut-path P_h that is an endpoint of a shared edge between the supports of h and g. By construction, the subpath P_1 from ℓ to t in P_h is then an M-alternating path. Let P_g denote the edges in the support of g, and let P_2 be the path from v to ℓ in P_g. Note that P_2 is also an M-alternating path. Then, one observes that either $P_2 \cup P_1$ is a proper M-augmenting tv-path in $G[c_{S\setminus v} - 1]$ (contradicting the previous claim), or $P_1 \cup (P_g \setminus P_2)$ is a circuit that we can apply to (fractionally) increase the cardinality of x in $G[c_S - 1]$, contradicting the stability of $G[c_S - 1]$. $\square$

Suppose for the sake of contradiction that $|M| < \nu^c(G)$. Then there exists a proper M-augmenting st-trail T in G, by Theorem 2.1 (note that, possibly, $s = t$). Since M is maximum in $G[c_S - 1]$, T cannot be proper in $G[c_S - 1]$, by Theorem 2.1. Therefore, $|S \cap \{s, t\}| \geq 1$. We distinguish two cases.

Case 1: $|S \cap \{s,t\}| = 1$. Without loss of generality, let s be the vertex whose capacity gets reduced by S. If $s \neq t$, then $c_s^{S\setminus s-1} = c_s^{S-1} + 1 \geq d_s^M + 1$ and $c_t^{S\setminus s-1} = c_t$. If $s = t$ then $c_s^{S\setminus s-1} = c_s$. In both cases, T is a proper M-augmenting trail in $G[c_{S\setminus s} - 1]$, because T is proper in G, contradicting the first claim.

Case 2: $|S \cap \{s,t\}| = 2$. If $s \neq t$, then $c_s^{S\setminus\{s,t\}-1} = c_s^{S-1} + 1 \geq d_s^M + 1$ and $c_t^{S\setminus\{s,t\}-1} = c_t^{S-1} + 1 \geq d_t^M + 1$. If $s = t$ then $c_s^{S\setminus\{s,s\}-1} = c_s^{S-1} + 2 \geq d_s^M + 2$. In both cases, T is a proper M-augmenting trail in $G[c_{S\setminus\{s,t\}} - 1]$, contradicting the second claim. $\square$

4.3.1 Increasing the Capacity

Let us consider a variant of the capacity-stabilizer problem where we instead increase the capacity of each vertex in S, so:

Given a graph $G = (V,E)$ with edge weights $w \in \mathbb{R}^E_{\geq 0}$ and vertex capacities $c \in \mathbb{Z}^V_{\geq 0}$, find a minimum-cardinality multiset S of vertices V such that $\nu_f^c(G[c_S + 1]) = \nu^c(G[c_S + 1])$.

We will show that all our results also work for this case. First, we need a different version of Theorem 4.3.

Theorem 4.9. *Let $\mathcal{P}$ be any polytope, $a^\top x \leq b$ be an inequality of the description of $\mathcal{P}$, and $\delta \in \mathbb{R}_{>0}$. Let $\overline{x}$ be an optimal vertex of the LP $\max\{c^\top x : x \in \mathcal{P}\}$, such that (i) $a^\top \overline{x} = b$, and (ii) there is no vertex $\widetilde{x}$ of $\mathcal{P}$ satisfying $b - \delta < a^\top \widetilde{x} < b$. Then it is possible to move to an optimal solution x^* of $\max\{c^\top x : x \in \mathcal{P}, a^\top x \leq b - \delta\}$ from $\overline{x}$ in one (partial) step over the edges of $\mathcal{P}$ (that is, there is an optimal vertex of $\{x \in \mathcal{P} : a^\top x \leq b - \delta\}$ that is (1) also a vertex of $\mathcal{P}$ and adjacent to $\overline{x}$, or (2) on an edge of $\mathcal{P}$ that is incident with $\overline{x}$). (See Figure 4.6.)*

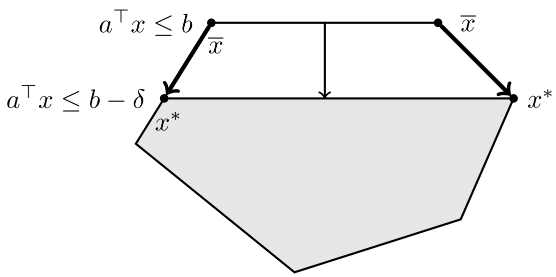


Figure 4.6: Example of the situation described in Theorem 4.3, with two options for $\overline{x}$ and x^*.

Proof. Let x^* be the optimal vertex of $\max\{c^\top x : x \in \mathcal{P}, a^\top x \leq b - \delta\}$ that is the *closest* to $\overline{x}$ on $\mathcal{P}$ (that is, such that we need a minimum number of (partial) steps over the edges of $\mathcal{P}$ to reach x^* from $\overline{x}$). Note that $a^\top x^* = b - \delta$, otherwise $x^* + \lambda(\overline{x} - x^*)$, for some small $\lambda > 0$, contradicts the optimality of x^*. We need to show that $\overline{x}$ and x^* are adjacent on $\mathcal{P}$.

Let $\mathcal{P}' = \{x \in \mathcal{P} : a^\top x \geq b - \delta\}$. Then $\overline{x}, x^* \in \mathcal{P}'$, and in particular x^* is a vertex of $\mathcal{P}'$. Note that $\overline{x}$ and x^* are adjacent on $\mathcal{P}$ or x^* is on an edge of $\mathcal{P}$ incident to $\overline{x}$ if and only if they are adjacent on $\mathcal{P}'$. So for the remainder of the proof we restrict ourselves to $\mathcal{P}'$.

For the sake of contradiction assume that $\overline{x}$ and x^* are not adjacent on $\mathcal{P}'$. Then, the line segment of all their convex combinations: $\lambda\overline{x} + (1 - \lambda)x^*$ for $0 \leq \lambda \leq 1$, is not an edge of $\mathcal{P}'$. Hence, any point $\lambda'\overline{x} + (1 - \lambda')x^*$ for a fixed $0 < \lambda' < 1$ is also a convex combination of other vertices of $\mathcal{P}'$: $\lambda'\overline{x} + (1 - \lambda')x^* = \sum_i \alpha_i \hat{x}_i + \sum_j \beta_j \widetilde{x}_j$, where $\alpha_i \geq 0$ for all i, $\beta_j \geq 0$ for all j, $\sum_i \alpha_i + \sum_j \beta_j = 1$, $\hat{x}_i$ is a vertex of $\mathcal{P}'$ with $a^\top \hat{x}_i = b - \delta$ for all i, and $\widetilde{x}_j$ is a vertex of $\mathcal{P}'$ with $a^\top \widetilde{x}_j = b$ for all j. If we multiply both sides by a we get

$$\begin{aligned}
& & a^\top\left(\lambda'\overline{x} + (1 - \lambda')x^*\right) &= a^\top\left(\textstyle\sum_i \alpha_i \hat{x}_i + \sum_j \beta_j \widetilde{x}_j\right), \\
&\iff & \lambda' b + (1 - \lambda')(b - \delta) &= \textstyle\sum_i \alpha_i (b - \delta) + \sum_j \beta_j b, \\
&\iff & b - (1 - \lambda')\delta &= \left(\textstyle\sum_i \alpha_i + \sum_j \beta_j\right) b - \sum_i \alpha_i \delta,
\end{aligned}$$

hence $1 - \lambda' = \sum_i \alpha_i$, and consequently $\lambda' = \sum_j \beta_j$. We can also multiply both sides by c. Here we use that $\overline{x}$ is an optimal solution of $\max\{c^\top x : x \in \mathcal{P}\}$, and that x^* is an optimal solution of $\max\{c^\top x : x \in \mathcal{P}, a^\top x \leq b - \delta\}$.

$$\begin{aligned}
c^\top\left(\lambda'\overline{x} + (1 - \lambda')x^*\right) &= c^\top\left(\textstyle\sum_i \alpha_i \hat{x}_i + \sum_j \beta_j \widetilde{x}_j\right) = \sum_i \alpha_i c^\top \hat{x}_i + \sum_j \beta_j c^\top \widetilde{x}_j \\
&\leq \textstyle\sum_i \alpha_i c^\top x^* + \sum_j \beta_j c^\top \overline{x} = (1 - \lambda') c^\top x^* + \lambda' c^\top \overline{x}
\end{aligned}$$

So we must have equality throughout. In particular, $c^\top \hat{x}_j = c^\top x^*$, that is, all $\hat{x}_j$ are optimal solutions to $\max\{c^\top x : x \in \mathcal{P}, a^\top x \leq b - \delta\}$. We show that we can choose some $\hat{x}_j$ to be adjacent to $\overline{x}$ on $\mathcal{P}'$, contradicting that x^* is the optimal solution closest to x.

Let x' be a vertex of $\mathcal{P}'$ that is adjacent to $\overline{x}$, such that $ax' = b - \delta$ (such an x' must exist). Consider the line segment between x' and $\lambda'\overline{x} + (1 - \lambda')x^*$: $\mu x' + (1 - \mu)(\lambda'\overline{x} + (1 - \lambda')x^*)$ for $0 \leq \mu \leq 1$. For $\mu < 0$, this line segment extends beyond $\lambda'\overline{x} + (1 - \lambda')x^*$. If this line for $\mu < 0$ is still in $\mathcal{P}'$, then we can write $\lambda'\overline{x} + (1 - \lambda')x^*$ as a convex combination of x' and some other $\hat{x}_i$'s and $\widetilde{x}_j$'s. Since $ax' = b - \delta$, by our previous discussion we find that x' is an optimal solution to $\max\{c^\top x : x \in \mathcal{P}, a^\top x \leq b - \delta\}$, reaching our desired contradiction. Otherwise, $\lambda'\overline{x} + (1 - \lambda')x^*$ must be at the boundary, a face, of $\mathcal{P}'$. Because

$\lambda'\overline{x} + (1-\lambda')x^*$ is in this face, the whole line segment $\lambda\overline{x} + (1-\lambda)x^*$ for $0 \le \lambda \le 1$ must be in this face. We can then repeat the argument, replacing $\mathcal{P}'$ by this face. Since this face has strictly smaller dimension than $\mathcal{P}'$, we either find a contradiction in one of the iterations, or we reach a face of dimension one, that is, an edge of $\mathcal{P}'$. Since this edge contains the whole line segment $\lambda\overline{x}+(1-\lambda)x^*$ for $0 \le \lambda \le 1$, the line segment is the edge, a contradiction. ☐

This also gives us an alternate version of Theorem 4.4.

Theorem 4.10. *Let $\overline{x}$ be a basic maximum-weight fractional c-matching in G, such that $\overline{x}(\delta(v)) = c_v$ for some $v \in V$. Then it is possible to move to a basic maximum-weight fractional c-matching in $G[c_v - 1]$ in one (partial) step over the edges of $\mathcal{P}_{FCM}(G, c)$.*

Proof. Let $\mathcal{P} = \mathcal{P}_{\text{FCM}}(G, c)$, $a^\top x \le b$ be $x(\delta(v)) \le c_v$, $\delta = 1$, and w be the objective function. It follows from Theorem 2.2 that $x(\delta(v))$ is integral for all basic fractional c-matchings. Consequently, there are no vertices $\widetilde{x}$ of $\mathcal{P}_{\text{FCM}}(G, c)$ that satisfy $c_v - 1 < \widetilde{x}(\delta(v)) < c_v$. The theorem now readily follows from Theorem 4.3. ☐

We use this result to show that $\gamma(G)$ is still a lower bound on the size of a(n increase) capacity-stabilizer.

Lemma 4.2. *For every (increase) capacity-stabilizer S, $|S| \ge \gamma(G)$.*

Proof. To prove the lemma, by Proposition 3.1, it is enough to show that increasing the capacity of any vertex by one decreases the number of fractional odd cycles by at most one. Therefore, from now on, we concentrate on proving the following statement:

$$\text{for all } v \in V,\ \gamma(G[c_v + 1]) \ge \gamma(G) - 1. \tag{4.2}$$

Let x be a basic maximum-weight fractional c-matching in $G[c_v + 1]$ with $\gamma(G[c_v + 1])$ fractional odd cycles. Let (y, z) be an optimal fractional vertex cover in $G[c_v + 1]$, satisfying complementary slackness (see Equation (2.2)) with x. Note that decreasing the capacity does not influence the feasibility of (y, z), hence (y, z) is a fractional vertex cover in G. However, x is only a fractional c-matching in G if $x(\delta_v) \le c_v$, and not if $x(\delta_v) = c_v + 1$.

Assume that $x(\delta_v) \le c_v$, that is, x is a fractional c-matching in G. By complementary slackness in $G[c_v+1]$ we have $y_v = 0$. The change from $G[c_v+1]$ to G only influences the complementary slackness condition $y_v = 0 \vee x(\delta(v)) = c_v+1$, which is satisfied by $y_v = 0$ in both graphs. Hence, x has maximum weight in G, which means that $\gamma(G) \le |\mathscr{C}_x| = \gamma(G[c_v + 1])$, and hence (4.2) holds.

Assume now that $x(\delta(v)) = c_v + 1$. Then, by Theorem 4.10, we can move to a basic maximum-weight fractional c-matching x^* in G in one (partial) step over the edges of $\mathcal{P}_{\mathrm{FCM}}(G, c)$. If we take a complete step, by Theorem 4.6, $x^* = x + \alpha g$ for $\alpha \in \left\{\frac{1}{2}, 1\right\}$ and $g \in \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3 \cup \mathcal{C}_4 \cup \mathcal{C}_5$. If we take only a partial step, then let x' be the vertex obtained by taking the complete step. Then, by Theorem 4.6, $x' = x + \alpha g$ for $\alpha \in \left\{\frac{1}{2}, 1\right\}$ and $g \in \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3 \cup \mathcal{C}_4 \cup \mathcal{C}_5$. To reach x^* we only take part of this step, so $x^* = x + \alpha' g$ for $0 < \alpha' < \alpha$. Now, since x^* is basic in G, we know that x^* is half-integral, so we must have $\alpha' = \frac{1}{2}$ (and $\alpha = 1$). Finally, in both cases we found that $x^* = x + \alpha g$ for $\alpha \in \left\{\frac{1}{2}, 1\right\}$ and $g \in \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3 \cup \mathcal{C}_4 \cup \mathcal{C}_5$. Since $x^*(\delta(v)) \leq c_v = x(\delta(v)) - 1$, we must have $g \in \mathcal{C}_2 \cup \mathcal{C}_3 \cup \mathcal{C}_4$. Then, by Theorem 4.6, we have $|\mathscr{C}_{x^*}| = |\mathscr{C}_x| \pm \{0, 1\}$. Therefore $|\mathscr{C}_{x^*}| \leq |\mathscr{C}_x| + 1$, and consequently, $\gamma(G) \leq |\mathscr{C}_{x^*}| \leq |\mathscr{C}_x| + 1 = \gamma(G[c_v + 1]) + 1$, yielding (4.2). □

Like in the capacity-reduction case, we stabilize by selecting one vertex per fractional odd cycle, except now we increase their capacity.

Algorithm 2 stabilization by capacity increase

1: initialize $S \leftarrow \emptyset$
2: compute a basic maximum-weight fractional c-matching x in G with $\gamma(G)$ fractional odd cycles $C_1, \ldots, C_{\gamma(G)}$, and a minimum fractional vertex cover (y, z) in G
3: **for** $i = 1$ to $\gamma(G)$ **do**
4: $\quad S \leftarrow S + v_i$ for any $v_i \in V(C_i)$
5: **end for**
6: **return** $G[c_S + 1]$

Theorem 4.11. *Algorithm 2 is a polynomial-time algorithm that computes a minimum (increase) capacity-stabilizer S for G. Moreover,*

(a) The solution S increases the capacity of each vertex by at most one unit.

(b) The solution S preserves the weight of a maximum-weight matching, that is, $\nu^c(G[c_S + 1]) \geq \nu^c(G)$.

(c) In unit-weight graphs, the solution S increases the size of a maximum matching by $\gamma(G)$, that is, $\nu^c(G[c_S + 1]) = \nu^c(G) + \gamma(G)$.

Proof. Let $S = \{v_1, \ldots, v_{\gamma(G)}\}$ be the set of vertices whose capacity is increased in Algorithm 2. Let $\hat{x}$ be obtained from x by alternate rounding C_i covering v_i, for all $i \in \{1, \ldots, \gamma(G)\}$. Clearly, $\hat{x}$ is a fractional c-matching in $G[c_S + 1]$. In addition, (y, z) is still a fractional vertex cover in $G[c_S + 1]$. One can check that they satisfy complementary slackness with respect to $G[c_S + 1]$. Hence, they are optimal in $G[c_S + 1]$. Note that $\hat{x}$ is an integral matching. Hence, $G[c_S + 1]$ is stable. Moreover, $|S| = \gamma(G)$, which is minimum by Lemma 4.2.

Since all cycles in $\mathscr{C}_x$ are vertex-disjoint, the set S is *not* a multiset. Hence, (a) holds. Using stability of $G[c_S + 1]$, optimality of (y, z) in $G[c_S + 1]$, and $y \geq 0$, we find

$$
\begin{aligned}
\nu^c(G[c_S + 1]) = \tau_f^c(G[c_S + 1]) &= (c^{S+1})^\top y + 1^\top z \\
&= c^\top y + \sum_{i=1}^{\gamma(G)} y_{v_i} + 1^\top z \geq c^\top y + 1^\top z \\
&= \tau_f^c(G) \geq \nu^c(G).
\end{aligned}
$$

So, (b) holds. Finally, to see (c), note that in unit-weight graphs we have $y_v = \frac{1}{2}$ for all $v \in V(\mathscr{C}_x)$. So $\sum_{i=1}^{\gamma(G)} y_{v_i} = \frac{1}{2}\gamma(G)$ in the equation above, which gives us $\nu^c(G[c_S+1]) = \tau_f^c(G) + \frac{1}{2}\gamma(G)$. In the proof of Theorem 4.7 we found that

$$
\nu^c(G[C_S - 1]) = c^\top y - \sum_{i=1}^{\gamma(G)} + 1^\top z = \tau_f^c(G) - \sum_{i=1}^{\gamma(G)} y_{v_i},
$$

and from Theorem 4.8 we know that $\nu^c(G[C_S - 1]) = \nu^c(G)$. Here we also have $y_v = \frac{1}{2}$ for all $v \in V(\mathscr{C}_x)$ because of the unit-weights, and so we find that $\tau_f^c(G) = \nu^c(G) + \frac{1}{2}\gamma(G)$. In total this gives us $\nu^c(G[c_S+1]) = \nu^c(G) + \gamma(G)$. □

4.4 Edge-Stabilizer

In this section we state our results for the edge-stabilizer problem. First, we generalize a lower bound on the size of an edge-stabilizer, provided in the unit-capacity setting.

Lemma 4.3. *For every edge-stabilizer F, $|F| \geq \frac{1}{2}\gamma(G)$.*

Proof. To prove the lemma, by Proposition 3.1, it is enough to show that removing one edge decreases the number of fractional odd cycles by at most two. Therefore, from now on, we concentrate on proving the following statement:

$$
\text{for all } e \in E,\ \gamma(G \setminus e) \geq \gamma(G) - 2. \tag{4.3}
$$

Let x be a basic maximum-weight fractional c-matching in $G \setminus e$ with $\gamma(G \setminus e)$ fractional odd cycles. Extend x to G by setting $x_e = 0$. Then x is a basic fractional c-matching in G.

If x has maximum weight in G, then $\gamma(G) \leq |\mathscr{C}_x| = \gamma(G \setminus e)$, and hence (4.3) holds.

If x does not have maximum weight in G, we can apply Theorem 4.5: we can move to a basic maximum-weight fractional c-matching x^* in G in at most two steps over the edges of $\mathcal{P}_{\mathrm{FCM}}(G, c)$, and if two steps are needed, the first one moves to a vertex with $x_e = \frac{1}{2}$, and the second one to a vertex with $x_e = 1$.

Suppose only one step was needed. By Theorem 4.6, $x^* = x + \alpha g$ for $\alpha \in \{\frac{1}{2}, 1\}$ and $g \in \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3 \cup \mathcal{C}_4 \cup \mathcal{C}_5$. Then, by Theorem 4.6, we have $|\mathscr{C}_{x^*}| = |\mathscr{C}_x| \pm \{0, 1, 2\}$. So definitely, $|\mathscr{C}_{x^*}| \leq |\mathscr{C}_x| + 2$, and consequently, $\gamma(G) \leq |\mathscr{C}_{x^*}| \leq |\mathscr{C}_x| + 2 = \gamma(G \setminus e) + 2$, yielding (4.3).

Suppose two steps were needed. Let $\hat{x}$ be the vertex reached after the first step. By Theorem 4.6, $\hat{x} = x + \alpha g$ and $x^* = \hat{x} + \beta h$ for $\alpha, \beta \in \{\frac{1}{2}, 1\}$ and $g, h \in \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3 \cup \mathcal{C}_4 \cup \mathcal{C}_5$. Now note that we must have $x_e = 0$, $\hat{x}_e = \frac{1}{2}$ and $x^*_e = 1$. So $\alpha = \beta = \frac{1}{2}$, g creates at least one cycle, and h breaks at least one cycle. Again looking at Theorem 4.6, this gives the following options for g:

- $g \in \mathcal{C}_1$ and $|\mathscr{C}_{\hat{x}}| = |\mathscr{C}_x|$,
- $g \in \mathcal{C}_2 \cup \mathcal{C}_4$ and $|\mathscr{C}_{\hat{x}}| = |\mathscr{C}_x| + 1$, since the odd cycle in g must belong to $\hat{x}$,
- $g \in \mathcal{C}_5$ and $|\mathscr{C}_{\hat{x}}| = |\mathscr{C}_x| + \{0, 2\}$, since at least one of the odd cycles in g must belong to $\hat{x}$.

Hence, $|\mathscr{C}_{\hat{x}}| = |\mathscr{C}_x| + \{0, 1, 2\}$. Similarly for h:

- $h \in \mathcal{C}_1$ and $|\mathscr{C}_{x^*}| = |\mathscr{C}_{\hat{x}}|$,
- $h \in \mathcal{C}_2 \cup \mathcal{C}_4$ and $|\mathscr{C}_{x^*}| = |\mathscr{C}_{\hat{x}}| - 1$, since the odd cycle in h must belong to $\hat{x}$,
- $h \in \mathcal{C}_5$ and $|\mathscr{C}_{x^*}| = |\mathscr{C}_{\hat{x}}| - \{0, 2\}$, since at least one of the odd cycles in h must belong to $\hat{x}$.

Hence, $|\mathscr{C}_{x^*}| = |\mathscr{C}_{\hat{x}}| - \{0, 1, 2\}$. So definitely, $|\mathscr{C}_{x^*}| \leq |\mathscr{C}_x| + 2$, and consequently, as before, $\gamma(G) \leq \gamma(G \setminus e) + 2$, yielding (4.3). □

Koh and Sanità [36] provided an example that shows this bound is tight, already in the unit-capacity setting. However, in the unit-weight, capacitated setting we can get a stronger bound. Repeating the proof above, but replacing the "-2" in (4.3) by "-1" and using that circuits in $\mathcal{C}_1 \cup \mathcal{C}_5$ are not augmenting in cardinality, we can prove the following.

Lemma 4.4. *In $(G, 1, c)$, for every edge-stabilizer F, $|F| \geq \gamma(G)$.*

Koh and Sanità [36] give a $O(\Delta)$-approximation algorithm, where Δ denotes the maximum degree in the graph, based on their algorithm for the vertex-stabilizer problem: instead of removing the vertices, all edges incident to those vertices are removed. This removes at most Δ edges per odd cycle, and hence results in a $O(\Delta)$-approximation algorithm. Similarly, we use our algorithm for the capacity-stabilizer problem (Algorithm 1), and instead of reducing the capacity of the vertices, remove all edges incident to those vertices, except the edges e such that $e \in \mathscr{M}_x$.

Theorem 4.12. *The edge-stabilizer problem admits an efficient $O(\Delta)$-approximation algorithm.*

Proof. Let $S = \{v_1, \ldots, v_{\gamma(G)}\}$ be the set of vertices found by Algorithm 1. Set $F = \{\delta(v) \setminus \mathscr{M}_x : v \in S\}$. The size of F is at most $\Delta\gamma(G)$. We claim that $G \setminus F$ is stable, hence this gives us an $O(\Delta)$-approximation by Lemma 4.3.

Let $\hat{x}$ be obtained from x by alternate rounding C_i exposing v_i, for all $i \in \{1, \ldots, \gamma(G)\}$. Note that $\hat{x}$ is a basic fractional c-matching in $G \setminus F$ with $|\mathscr{C}_{\hat{x}}| = 0$. Let $(\hat{y}, \hat{z})$ be obtained from (y, z) as follows:

$$\hat{y}_v = \begin{cases} y_v & \text{if } v \notin S, \\ 0 & \text{if } v \in S, \end{cases} \qquad \hat{z}_{uv} = \begin{cases} z_{uv} + y_u + y_v & \text{if } u, v \in S, uv \in E \setminus F, \\ z_{uv} + y_u & \text{if } u \in S, v \notin S, uv \in E \setminus F, \\ z_{uv} + y_v & \text{if } u \notin S, v \in S, uv \in E \setminus F, \\ z_{uv} & \text{if } u, v \notin S, uv \in E \setminus F. \end{cases}$$

One can check that $\hat{x}$ and $(\hat{y}, \hat{z})$ satisfy complementary slackness in $G \setminus F$. Consequently, $\hat{x}$ is a basic maximum-weight fractional c-matching in $G \setminus F$ with $|\mathscr{C}_{\hat{x}}| = 0$, and so $G \setminus F$ is stable. □

If we restrict ourselves to unit-weight instances, like for capacity-stabilizers, we can show that any *inclusion-wise minimal* edge-stabilizer preserves the size of a maximum-cardinality c-matching.

Theorem 4.13. *In $(G, 1, c)$, for any inclusion-wise minimal edge-stabilizer F, we have $\nu^c(G \setminus F) = \nu^c(G)$.*

Proof. Let M be a maximum-cardinality c-matching in G such that the overlap with F is minimum, that is, such that $|M \cap F|$ is minimum. Suppose for the sake of contradiction that $|M \cap F| > 0$.

Consider the graph $G \setminus (F \setminus M)$. Since M is avoided, M is a maximum-cardinality c-matching in $G \setminus (F \setminus M)$. Let x be the indicator vector of M, then by Theorem 2.2, x is basic. Since F is inclusion-wise minimal, $G \setminus (F \setminus M)$ is not stable, and thus x is not a maximum fractional c-matching in $G \setminus (F \setminus M)$.

So there must be a vertex x^* of $\mathcal{P}_{\text{FCM}}$, adjacent to x, with $1^\top x^* > 1^\top x$. By Theorem 4.6, $x^* = x + \alpha g$, where $\alpha \in \{\frac{1}{2}, 1\}$ and $g \in \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3 \cup \mathcal{C}_4 \cup \mathcal{C}_5$. Since M is maximum, x^* cannot be integral, so $\alpha = \frac{1}{2}$, and consequently, by Theorem 4.6, $g \notin \mathcal{C}_3$. Furthermore, the circuits in $\mathcal{C}_1 \cup \mathcal{C}_5$ are not augmenting in cardinality, so $g \in \mathcal{C}_2 \cup \mathcal{C}_4$. Let $v \in V$ be the only vertex with $g(\delta(v)) \neq 0$. The circuits in $\mathcal{C}_2 \cup \mathcal{C}_4$ are only augmenting in cardinality if $g(\delta(v)) > 0$. Consequently, $x^*(\delta(v)) = x(\delta(v)) + 1 = d_v^M + 1$. Clearly, we have $x^*(\delta(v)) \leq c_v$, and so, $d_v^M < c_v$.

The support of g consists of a (possibly empty) path P_g and an odd cycle C_g, intersecting at only one vertex, such that the sign of g_e alternates. Since it is feasible to apply g to x, it must be that, if $sgn(g_e) = -$, then $x_e = 1$ ($e \in M$), and if $sgn(g_e) = +$, then $x_e = 0$ ($e \notin M$). Since $g(\delta(v)) > 0$, the first and last edge of g satisfy $sgn(g_e) = +$. Recall that $d_v^M < c_v$. Consider the closed vv-walk $W = (P_g, C_g, P_g^{-1})$. Then W is a feasible M-augmenting walk in $G \setminus (F \setminus M)$.

Claim 4.6. *$W \cap F = \emptyset$.*

Proof. For the sake of contradiction, suppose that there is some $e \in W \cap F$. Then $e \in M$, since all other edges of F were removed. Then there exists an even-length M-alternating trail T: the part of W starting from v, to (and including) the edge e. Since $d_v^M < c_v$ and $e \in M$, T is proper. Then $M' = M \triangle T$ is a maximum-cardinality c-matching in G with $|M' \cap F| < |M \cap F|$, contradicting our assumption. $\square$

By this claim, W is in $G \setminus F$. In addition, we have $d_v^{M \setminus F} \leq d_v^M < c_v$. Thus, W is a feasible $M \setminus F$-augmenting walk in $G \setminus F$. By Theorem 5.3, since $G \setminus F$ is stable, it follows that $M \setminus F$ is not maximum in $G \setminus F$. Then, by Theorem 2.1, there is a proper $M \setminus F$-augmenting st-trail T in $G \setminus F$. Note that, since T is augmenting in cardinality, the first and last edge of T are not in $M \setminus F$. Since W cannot exist for a maximum-cardinality c-matching in $G \setminus F$, by Theorem 5.3, there must be such a trail T that either makes W infeasible, or overlaps with the edges of W. Note that T is also an M-augmenting trail in G, but M is maximum in G, so by Theorem 2.1, T is not proper for M in G.

Since T is proper for $M \setminus F$ in $G \setminus F$, we have, if $s \neq t$, $d_s^{M \setminus F} \leq c_s - 1$ and $d_t^{M \setminus F} \leq c_t - 1$, if instead $s = t$, then $d_s^{M \setminus F} \leq c_s - 2$. Since T is not proper for M in G, we have, if $s \neq t$, $d_s^M = c_s$ or $d_t^M = c_t$, if instead $s = t$, then $d_s^M \geq c_s - 1$. This gives us five cases:

1. $s \neq t$, $d_s^M = c_s$ and $d_t^M < c_t$,
2. $s \neq t$, $d_s^M < c_s$ and $d_t^M = c_t$,

3. $s \neq t$, $d_s^M = c_s$ and $d_t^M = c_t$,
4. $s = t$ and $d_s^M = c_s - 1$,
5. $s = t$ and $d_s^M = c_s$.

In the first case, since $d_s^M = c_s$ and $d_s^{M \setminus F} \leq c_s - 1$, there is at least one $e \in \delta(s) \cap M \cap F$. Since $e \in F$, we have $e \notin T$, which means $T \cup e$ is a trail. Note that $T \cup e$ is M-alternating and has even length. Since $d_t^M < c_t$ and $e \in M$, $T \cup e$ is proper. (If $T \cup e$ is closed, it is still proper, because it has even-length.) Therefore, $M' = M \triangle (T \cup e)$ is a maximum-cardinality c-matching in G with $|M' \cap F| < |M \cap F|$, contradicting our assumption. Similar arguments can be made in the second and fourth case.

For the third and fifth case we take a look at the overlap of T and W. We know that $d_v^{M \setminus F} \leq d_v^M < c_v$. So, to make W infeasible in $G \setminus F$, T would need to increase the degree of v with respect to the matching. This is only possible if $v \in \{s, t\}$. However, s and t are both saturated by M (in both the third and fifth case) and v is not. Hence, T and W must overlap in at least one edge.

We create a new walk W' by combining W and T: follow W starting from v to the first edge e that is also on T, traverse e, then switch to T and follow T from here to one of its endpoints. Since T and W are both M-alternating and -augmenting, so is W'. The part of W from v to and including e is a trail, this part of W does not overlap with T, and T is a trail, hence W' is a trail. Since T exists in $G \setminus F$, and $W \cap F = \emptyset$ by Claim 4.6, we have $W' \cap F = \emptyset$.

In both the third and fifth case, the degree of both s and t with respect to M is strictly larger than their degree with respect to $M \setminus F$. Hence, $\delta(s) \cap M \cap F$ and $\delta(t) \cap M \cap F$ are nonempty. Without loss of generality assume W' ends at s. Let $e \in \delta(s) \cap M \cap F$. Since $e \in F$, we have $e \notin W'$, which means $W' \cup e$ is a trail. Note that $W' \cup e$ is M-alternating and has even length. Since $d_v^M < c_v$ and $e \in M$, $W' \cup e$ is proper. Therefore, $M' = M \triangle (W' \cup e)$ is a maximum-cardinality c-matching in G with $|M' \cap F| < |M \cap F|$, contradicting our assumption. $\square$

We conclude this section with some additional remarks. Note that, as opposed to the capacity-stabilizer case, when dealing with edge-removal operations it is always possible to stabilize a graph without decreasing the weight of a maximum-weight matching: for example, one could take any maximum-weight c-matching M in G and remove all edges in $E \setminus M$. The previous theorem shows that, for unit-weight instances, this property comes essentially for free, as any edge-stabilizer of minimum cardinality is weight-preserving. However, for general weighted instances, this is not the case, and we can show that the size of a minimum weight-preserving edge-stabilizer and the size of a minimum edge-stabilizer can differ by a very large factor, namely $\Omega(|V|)$, already for unit-capacities.

Theorem 4.14. *There exist graphs $(G, w, 1)$ where the sizes of a minimum edge-stabilizer and a minimum weight-preserving edge-stabilizer differ by $\Omega(|V|)$.*

Proof. Let G and M be the graph and matching given in Figure 4.7, respectively. Any matching in G can contain (i) at most one edge of each 3-cycle,

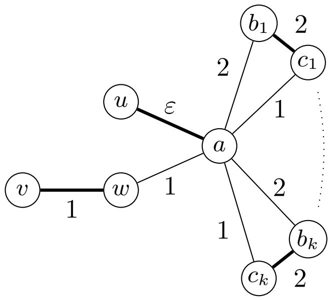


Figure 4.7: Let $k \geq 3$ be an integer, and $0 < \varepsilon < 0.5$. The figure shows a graph $(G, w, 1)$ with k 3-cycles of the form (a, b_i, c_i) for $i = 1, \ldots, k$. Edge weights are given next to the edges. The bold edges indicate a matching M.

which all have a weight of at most 2, (ii) at most one of the edges incident to w, which both have weight 1, and (iii) the edge ua of weight ε. This gives us $\nu^c(G) \leq 2k + 1 + \varepsilon$. In particular, M is the only matching in G that attains this weight. So, M is the unique maximum-weight matching in G. There are k feasible M-augmenting walks: (u, a, b_i, c_i, a, u) for $i = 1, \ldots, k$, hence G is not stable by Theorem 5.3.

We can stabilize G by removing only two edges: ua and vw. We verify the stability by giving a matching and a vertex cover of the same value. Let the matching be $wa \cup \{b_i c_i : i \in \{1, \ldots, k\}\}$. The weight of this matching is $2k+1$. Note that the weight decreased by ε. We set the vertex cover y equal to 0 for u, v, w and equal to 1 for a and all k pairs b_i, c_i. The value of this vertex cover is $2k + 1$. So indeed, the graph is stable. This shows that the size of a minimum edge-stabilizer is at most two.

As mentioned, M is the unique maximum-weight matching in G. Consequently, any weight-preserving edge-stabilizer has to avoid M. Because the k feasible M-augmenting walks are edge-disjoint with respect to the edges from $E \setminus M$, any weight-preserving edge-stabilizer has to remove at least k edges.

We have $|V| = 3k + 3$, or equivalently, $k = |V|/3 - 1$. The difference in size of a minimum edge-stabilizer and a minimum weight-preserving edge-stabilizer in this graph is at least $k-2$. So the difference in sizes is $\Omega(k) = \Omega(|V|)$. $\square$

Theorem 9 of Koh and Sanità [36] shows that there is no constant factor approximation for the minimum edge-stabilizer problem in $(G, w, 1)$, unless $P = NP$. We note that their proof actually also shows inapproximability for the minimum weight-preserving edge-stabilizer problem.

Our last remark is the following. For the vertex-stabilizer problem, the setting of removing the vertices completely has been analyzed in Section 4.1, and that of reducing the capacity has been investigated in Section 4.3. One may wonder whether a similar setting of "partial reduction" also makes sense for the edge-stabilizer problem: what if one reduces the weight of the edges, instead of completely removing them? The next theorem suggests that reducing edge weights might not be that interesting from a bargaining perspective: if the weight of an edge is decreased, it will not be part of any maximum-weight c-matching, so the edge could just as well be removed.

Theorem 4.15. *Let (G, w, c) be a graph, and let $\widetilde{w} \in \mathbb{R}^E_{\geq 0}$ such that $(G, w - \widetilde{w}, c)$ is stable and $1^\top \widetilde{w}$ is minimum. For every $f \in E$ such that $\widetilde{w}_f > 0$, the edge f is not in any maximum-weight c-matching in $(G, w - \widetilde{w}, c)$.*

Proof. For the sake of contradiction let M be a maximum-weight c-matching in $(G, w - \widetilde{w}, c)$ with $f \in M$. Let x be the indicator vector of M. Since $(G, w - \widetilde{w}, c)$ is stable, there is a fractional vertex cover (y, z) that satisfies complementary slackness with x:

$$\begin{aligned}
&x_e = 0 \vee y_u + y_v + z_e = w_e - \widetilde{w}_e && \forall e = uv \in E \\
&y_v = 0 \vee x(\delta(v)) = c_v && \forall v \in V \\
&z_e = 0 \vee x_e = 1 && \forall e \in E
\end{aligned}$$

Now, let $x' = x$, $y' = y$, $z'_f = z_f + \widetilde{w}_f$, $z'_e = z_e$ for all $e \neq f$, $w'_f = 0$, and $w'_e = \widetilde{w}_e$ for all $e \neq f$. One can check that x' and (y', z') are a c-matching and fractional vertex cover in $(G, w - w', c)$, respectively, and that they satisfy complementary slackness. Hence, $(G, w - w', c)$ is stable, but $1^\top w' < 1^\top \widetilde{w}$, contradicting that $1^\top \widetilde{w}$ is minimum.

In case of unit-capacities there are no z variables in the dual. To make the proof still valid, it is enough to increase the y-value of one of the vertices of f with $\widetilde{w}_f$. $\square$

Chapter 5

The M-Stabilizer Problem

In this chapter we discuss the M-vertex- and M-edge-stabilizer problem, in Sections 5.1 and 5.2, respectively. We defined these problems before as follows.

The M-vertex-stabilizer problem: Given a graph $G = (V, E)$ with edge weights $w \in \mathbb{R}^E_{\geq 0}$ and vertex capacities $c \in \mathbb{Z}^V_{\geq 0}$, and a c-matching M in G, find a minimum-cardinality subset $S \subseteq V$ of vertices such that $\nu^c_f(G \setminus S) = \nu^c(G \setminus S)$ and M is a maximum-weight c-matching in $G \setminus S$.

The M-edge-stabilizer problem: Given a graph $G = (V, E)$ with edge weights $w \in \mathbb{R}^E_{\geq 0}$ and vertex capacities $c \in \mathbb{Z}^V_{\geq 0}$, and a c-matching M in G, find a minimum-cardinality subset $F \subseteq E$ of edges such that $\nu^c_f(G \setminus F) = \nu^c(G \setminus F)$ and M is a maximum-weight c-matching in $G \setminus F$.

Section 5.1 is based on (part of) [V1].

Notation. In this chapter we denote a weighted, capacitated graph (G, w, c) together with a c-matching M as $[(G, w, c), M]$. For $X \subseteq V$, we denote by $(G, w, c) \setminus X$ the graph $G \setminus X$ with the edge weights w restricted to the edges in $G \setminus X$ and the vertex capacities c restricted to the vertices $V \setminus X$. We say that $[(G, w, c), M]$ is stable if $\nu^c_f(G) = \nu^c(G)$ and M is a maximum-weight c-matching in G.

5.1 M-Vertex-Stabilizer

In this section we give our M-vertex-stabilizer results. First, in Section 5.1.1 we describe an auxiliary construction to reduce a given problem instance to a unit-capacity instance. Then, in Section 5.1.2 we give our M-vertex-stabilizer algorithm.

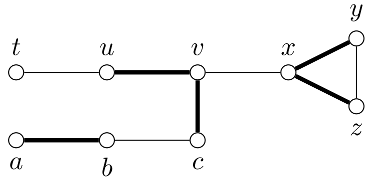
5.1.1 Auxiliary Construction

We use a construction given in Section 4.1 of Farczadi et al. [21], to transform a weighted, capacitated graph G and c-matching M into a weighted, unit-capacity auxiliary graph G' and matching M'.

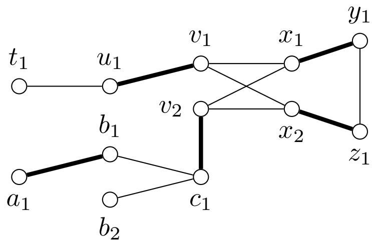
Construction: $[(G, w, c), M] \to [(G', w', 1), M']$

1. For each $v \in V$, create the set $C_v = \{v_1, \dots, v_{c_v}\}$ of c_v copies of v, add C_v to $V(G')$, and initialize $J(v) = \{1, \dots, c_v\}$.
2. For each $uv \in M$, add a single edge $u_i v_j$ to both $E(G')$ and M' with edge-weight w_{uv}, where $i \in J(u)$ and $j \in J(v)$ are chosen arbitrarily. Remove i and j from $J(u)$ and $J(v)$, respectively.
3. For each edge $uv \in E \setminus M$, add an edge $u_i v_j$ to $E(G')$ with edge-weight w_{uv}, for all $u_i \in C_u$ and $v_j \in C_v$.

See Figure 5.1 for an example. In this figure it is easy to see that the matching M' in G' is not maximum, even though M is maximum in G.[1]



(a) Original graph.



(b) Auxiliary graph.

Figure 5.1: Example of the auxiliary construction on an instance $[(G, w, c), M]$. Capacities are all 1 except for $c_v = c_x = c_b = 2$. Weights are all 1 except for $w_{bc} = 0.5$. The matching is displayed as bold edges.

Remark 5.1. *If $[(G, w, c), M]$ has auxiliary $[(G', w', 1), M']$, and $X \subseteq V$ is any set of vertices which avoids M, then $[(G, w, c) \setminus X, M]$ has auxiliary $[(G', w', 1) \setminus X', M']$, where $X' = \cup_{v \in X} C_v$.*

We define a map η to go back from the auxiliary graph G' to the original graph G. Specifically, if $u_i \in V(G') \cap C_u$ for some $u \in V$, then $\eta(u_i) = u$, and if $u_i v_j \in E(G')$ such that $u_i \in C_u$, $v_j \in C_v$ for some $u, v \in V$, then $\eta(u_i v_j) = uv$. This extends in the obvious way to paths, cycles, walks, and so on.

We need the following theorem.

[1] It was stated in Corollary 1 in Farczadi et al. [21] that M is maximum if and only if M' is maximum, but this example shows the forward direction to be false.

Theorem 5.1. *$[(G, w, c), M]$ is not stable if and only if the graph G' in the auxiliary $[(G', w', 1), M']$ contains at least one of the following: (i) an M'-augmenting flower; (ii) an M'-augmenting bi-cycle; (iii) a proper M'-augmenting path; (iv) an M'-augmenting cycle.*

Proof. It was proven in Theorem 2 in Farczadi et al. [21] that $[(G, w, c), M]$ is not stable if and only if the graph G' in the auxiliary $[(G', w', 1), M']$ is not stable, or the matching M' in the auxiliary $[(G', w', 1), M']$ does not have maximum weight. If G' is not stable (and M' has maximum weight), then G' contains an M'-augmenting flower or bi-cycle, see Theorem 1 in Koh and Sanità [36]. If M' does not have maximum-weight, G' must contain a proper M'-augmenting path or cycle, by standard matching theory. $\square$

We refer to an augmenting structure of type $(i) - (iv)$ in Theorem 5.1 as a *basic* augmenting structure.

The following lemma will be useful.

Lemma 5.1. *Given $[(G, w, c), M]$ and auxiliary $[(G', w', 1), M']$, let P be a feasible M'-augmenting walk. Then, $\eta(P)$ is a feasible M-augmenting walk.*

Proof. Let $e_1 = uv$ and $e_2 = vw$ be two consecutive edges on P. Then $\eta(e_1)$ and $\eta(e_2)$ are the corresponding edges on $\eta(P)$, and they are both incident with $\eta(v)$. Hence, $\eta(P)$ is a walk. For an edge e on P, we have $e \in M'$ if and only if $\eta(e) \in M$. In addition, $w'_e = w_{\eta(e)}$. So, $\eta(P)$ is an M-augmenting walk. Suppose $P = (u; e_1, \ldots, e_k; v)$. Feasibility of P means that either $e_1 \in M'$, or u is M'-exposed. Likewise for e_k and v. It follows that either $\eta(e_1) \in M$, or $\eta(u)$ is M-unsaturated. Likewise for $\eta(e_k)$ and $\eta(v)$. This means $\eta(P)$ is feasible. $\square$

5.1.2 Algorithm

The goal of this section is to prove the following theorem.

Theorem 5.2. *The M-vertex-stabilizer problem on weighted, capacitated graphs admits an efficient 2-approximation algorithm. Furthermore, the algorithm is exact if the given c-matching M has maximum weight.*

A natural strategy would be to first apply the auxiliary construction described in Section 5.1.1 to reduce to unit-capacity instances, and then apply the algorithm proposed by Koh and Sanità [36] which solves the problem exactly. However, there is a critical issue with this strategy. Namely, the auxiliary construction applied to unstable instances does *not* always preserve maximality of the corresponding matchings, as shown in Figure 5.1. In that example, the matching M' is not maximum in G'. The algorithm of Koh and Sanità [36], if applied to an instance where the given matching is not maximum, is not guaranteed to find an optimal solution, but only a 2-approximate one (see Theorem 12 in Koh and Sanità [36]). In addition, since the auxiliary construction splits a vertex into multiple ones, we may even get infeasible solutions. As a concrete example of this, the algorithm of Koh and Sanità [36] applied to the instance of Figure 5.1b includes b_2 in its proposed solution. Mapping this solution to our capacitated instance would imply to remove b, which is clearly not allowed as b is M-covered.

To overtake this issue, we do not apply the algorithm of Koh and Sanità [36] as a black-box, but use parts of it (highlighted in Lemma 5.2 below) in a careful way. In particular, we use it to compute a sequence of feasible augmenting walks in G'. We actually show that the walks in G' which might create the issue described before when mapped backed to G, are the walks in which at least one edge of G is traversed more than once in opposite directions, and that have two distinct endpoints. When this happens, we prove that we can modify the walk and get one where the endpoints coincide, which is still feasible and augmenting. In this latter case, we can then either correctly identify a vertex to remove (the unique endpoint), or determine that the instance cannot be stabilized.

Lemma 5.2. *Let G' be a unit-capacity graph, and M' a matching of G'.*

(a) For a given M'-exposed vertex u, one can compute a feasible M'-augmenting walk starting at u of length at most $3|V(G')|$, or determine that none exists, in polynomial time.

(b) A feasible M'-augmenting uv-walk contains a feasible M'-augmenting uv-path (proper if $u \neq v$), an M'-augmenting cycle, an M'-augmenting flower rooted at u or v, or an M'-augmenting bi-cycle. Furthermore, this augmenting structure can be computed in polynomial time.

Proof. (a) When given a graph G', a matching M', a vertex u, and an integer k, Algorithm 3 in Koh and Sanità [36] computes a feasible M'-augmenting uv-walk of length at most k, or determines none exist, for all $v \in V(G')$. Correctness is shown in Lemmas 7 and 8 in Koh and Sanità [36]. The algorithm runs in time that is polynomial in k, $|V(G')|$, and $|E(G')|$. We use this algorithm and select an arbitrary v for which a uv-walk is returned, or determine that no such walk starting at u exists. Since we set $k = 3|V(G')|$, this procedure terminates in polynomial time.

(b) Lemma 9 in Koh and Sanità [36] states that a feasible M'-augmenting uv-walk contains a feasible M'-augmenting uv-path, an M'-augmenting cycle, an M'-augmenting flower rooted at u or v, or an M'-augmenting bi-cycle. By

Proposition 2.1 the path is proper if $u \neq v$. Lemma 9 in Koh and Sanità [36] is proven in a constructive way, hence it also gives a way to compute the augmenting structure in polynomial time. $\square$

We next define *ties*.

Definition 5.1. Given $[(G, w, c), M]$ with auxiliary $[(G', w', 1), M']$, and an M'-alternating path P', a *tie* in P' is a pair of unmatched edges $\{ab, cd\}$ on P' such that for some distinct $u, v \in V$, either (i) $\{a, c\} \subseteq C_u$ and $\{b, d\} \subseteq C_v$ or (ii) $\{a, d\} \subseteq C_u$ and $\{b, c\} \subseteq C_v$. We say P' is *tieless* if it does not contain a tie.

We now show that if the auxiliary construction does not preserve maximality of the c-matching M, then we must have ties in all proper M'-augmenting paths and cycles.

Lemma 5.3. *Given $[(G, w, c), M]$ with auxiliary $[(G', w', 1), M']$, if M is a maximum-weight c-matching in G, then all proper M'-augmenting paths and cycles contain ties.*

Proof. We prove this by contraposition. Suppose that there is a proper M'-augmenting path or cycle P' that is tieless. Note that P' is also feasible. By Lemma 5.1, $P = \eta(P')$ is a feasible M-augmenting walk. Since P' is tieless, there is a bijection between $E(P')$ and $E(P)$, and so, as P' does not repeat edges, neither does P. Hence P is a feasible M-augmenting trail. We show that P is proper.

If P' is an M'-augmenting cycle, P is a closed M-augmenting trail of even length. It follows that $d_v^{P \triangle M} = d_v^M \leq c_v$ for all vertices v on P, and hence P is proper.

Now suppose P' is a proper M'-augmenting path. Let $P' = (u_i; e'_1, \ldots, e'_k; v_j)$ and $u = \eta(u_i)$, $v = \eta(v_j)$, $e_1 = \eta(e'_1)$ and $e_k = \eta(e'_k)$. Note that, because P' is proper, $e'_1 \notin M$ if and only if u_i is M'-exposed. Likewise for e'_k and v_j.

Case 1: $u = v$. If at most one of u_i and v_j is M'-exposed, then at least one of e'_1 and e'_k is in M' and hence at least one of e_1 and e_k is in M. Therefore, $d_u^{P \triangle M} \leq d_u^M \leq c_u$. If both u_i and v_j are M'-exposed, then $e'_1, e'_k \notin M'$ and hence $e_1, e_k \notin M$. Therefore, $d_u^{P \triangle M} = d_u^M + 2$. By construction there are c_u copies of u, and since u_i and v_j are already two of those copies, and they are exposed, we have $d_u^M \leq c_u - 2$. Thus $d_u^{P \triangle M} \leq c_u$.

Case 2: $u \neq v$. If $e'_1 \in M'$, then $e_1 \in M$, and so we have $d_u^{P \triangle M} = d_u^M - 1 \leq c_u$. If $e'_1 \notin M'$, then $e_1 \notin M$, and so we have $d_u^{P \triangle M} = d_u^M + 1$. Using the same reasoning as in case 1, we can conclude that $d_u^M \leq c_u - 1$ because u_i is M'-exposed, and therefore $d_u^{P \triangle M} \leq c_u$. The argument is analogous for v.

In all cases P is a proper M-augmenting trail. It follows by Theorem 2.1 that M is not a maximum-weight c-matching in G. $\square$

We now define the operation of *traceback*, which we will use to modify the feasible augmenting walks, when needed.

Definition 5.2. Given $[(G, w, c), M]$ and an M-alternating walk $P = (u; e_1, \dots, e_k; v)$ which repeats an edge in opposite directions, let t be the least index such that $e_t = e_s$ for some $s < t$, and e_s and e_t are traversed in opposite directions by P. Then the *u-traceback* and *v-traceback* of P are defined as the walks $\mathrm{tb}(P, u) = (e_1, \dots, e_t, e_{s-1}, e_{s-2}, \dots, e_1)$ and $\mathrm{tb}(P, v) = (e_k, e_{k-1}, \dots, e_s, e_{t+1}, e_{t+2}, \dots, e_k)$.

The next lemma explains how to use the traceback operation.

Lemma 5.4. *Given $[(G, w, c), M]$ such that M has maximum weight, and auxiliary $[(G', w', 1), M']$, let $P' = (u_i; e'_1, \dots, e'_k; v_j)$ be a proper M'-augmenting path such that both u_i and v_j are M'-exposed and $\eta(u_i) \neq \eta(v_j)$. Then, (i) $\eta(P')$ is an M-alternating walk that repeats an edge in opposite directions, and (ii) $tb(\eta(P'), \eta(u_i))$ and $tb(\eta(P'), \eta(v_j))$ are feasible M-alternating walks and at least one of them is M-augmenting.*

Proof. Let $P = \eta(P') = (u; e_1, \dots, e_k; v)$. By Lemma 5.1, P is a feasible M-augmenting walk, and also proper by Proposition 2.1, since $u \neq v$. In order to use the traceback operation, we must show that P traverses some edge in opposite directions. By Lemma 5.3 we already have that P' contains a tie, and hence that P traverses some edge twice. We now show that there must exist at least one edge that is traversed in opposite direction. Suppose not, let t be the least index such that $e_t = e_s$ for some $s < t$. Decompose P as $(P_1, e_s, P_2, e_t, P_3)$.

Claim 5.1. *If P traverses e_s and e_t in the same direction, then (P_1, e_s, P_3) is a shorter proper M-augmenting walk.*

Proof. For notation, define $P_2^+ = (P_2, e_t)$. By choice of t, P_2^+ is an M-alternating closed trail of even length. It follows that $d_v^{P_2^+ \triangle M} = d_v^M \leq c_v$ for all vertices v on P_2^+, and hence P_2^+ is proper. Since M has maximum weight, Theorem 2.1 implies that P_2^+ cannot be M-augmenting. However, P is M-augmenting, which means the augmenting part must come from $P \backslash P_2^+$. Hence, (P_1, e_s, P_3) is an M-augmenting walk. It is proper because P is proper. $\square$

By this claim $W = (P_1, e_s, P_3)$ is a shorter proper M-augmenting walk. W also necessarily repeats an edge, because otherwise W is a proper M-augmenting trail, contradicting that M has maximum weight, by Theorem 2.1. Then we can apply the claim again, to find an even shorter proper M-augmenting walk. This argument can be repeated until eventually we reach a contradiction.

Thus there is at least one edge traversed in opposite direction, hence we can use the traceback operation. Since P is M-alternating (because it is M-augmenting), $\mathrm{tb}(P,u)$ and $\mathrm{tb}(P,v)$ are also M-alternating. Furthermore, since u_i and v_j are M'-exposed, u and v are M-unsaturated. It follows that $\mathrm{tb}(P,u)$ and $\mathrm{tb}(P,v)$ are feasible.

That leaves to show that at least one of them is M-augmenting. For notation, let t be the least index such that $e_t = e_s$ for some $s < t$ and e_t and e_s are traversed in opposite direction. As before, decompose P as $(P_1, e_s, P_2, e_t, P_3)$. Define $P_2^{++} = (e_s, P_2, e_t)$, $P_u = \mathrm{tb}(P,u)$, and $P_v = \mathrm{tb}(P,v)$. Note that $P_u = (P_1, P_2^{++}, P_1^{-1})$ and $P_v = (P_3^{-1}, (P_2^{++})^{-1}, P_3)$.

Case 1: $w(P_1 \setminus M) - w(P_3 \setminus M) > w(P_1 \cap M) - w(P_3 \cap M)$. Because P is M-augmenting, we know that

$$\begin{aligned} &w(P_1 \setminus M) + w(P_2^{++} \setminus M) + w(P_3 \setminus M) \\ &\qquad > w(P_1 \cap M) + w(P_2^{++} \cap M) + w(P_3 \cap M). \end{aligned} \tag{5.1}$$

Adding these inequalities, we obtain

$$\begin{aligned} w(P_u \setminus M) &= 2w(P_1 \setminus M) + w(P_2^{++} \setminus M) \\ &> 2w(P_1 \cap M) + w(P_2^{++} \cap M) = w(P_u \cap M). \end{aligned} \tag{5.2}$$

Hence, P_u is M-augmenting.

Case 2: $w(P_1 \setminus M) - w(P_3 \setminus M) < w(P_1 \cap M) - w(P_3 \cap M)$. Subtracting this inequality from Equation (5.1), we obtain

$$\begin{aligned} w(P_v \setminus M) &= w(P_v^{++} \setminus M) + 2w(P_3 \setminus M) \\ &> w(P_2^{++} \cap M) + 2w(P_3 \cap M) = w(P_v \cap M). \end{aligned} \tag{5.3}$$

Hence, P_v is M-augmenting.

Case 3: $w(P_1 \setminus M) - w(P_3 \setminus M) = w(P_1 \cap M) - w(P_3 \cap M)$. Adding this inequality to Equation (5.1) we obtain Equation (5.2) again, and subtracting it from Equation (5.1), we obtain Equation (5.3) again. Hence, both P_u and P_v are M-augmenting. ☐

The next theorem is standard.

Theorem 5.3. *$[(G, w, c), M]$ is stable if and only if G does not contain a feasible M-augmenting walk.*

Proof. ($\Rightarrow$) Assume there exists a feasible M-augmenting walk W. Since W is augmenting, $w(W \setminus M) > w(W \cap M)$, and since W is feasible, $x^{M/W}(\varepsilon)$ is a fractional c-matching. Together they imply

$$\nu_f^c(G) \geq w^\top x^{M/W}(\varepsilon) = w(M) - \varepsilon w(W \cap M) + \varepsilon w(W \setminus M) > w(M),$$

that is, the instance $[(G, w, c), M]$ is not stable.

($\Leftarrow$) Assume the instance is not stable. Then by Theorem 5.1, the graph G' from the auxiliary $[(G', w', 1), M']$ contains a basic augmenting structure, which clearly is a feasible M'-augmenting walk P. Then $\eta(P)$ is a feasible M-augmenting walk, by Lemma 5.1. $\square$

The algorithm is stated in Algorithm 3. We are now ready to prove Theorem 5.2.

Proof of Theorem 5.2. Let $[(G, w, c), M]$ be the input for the M-vertex-stabilizer problem, with auxiliary $[(G', w', 1), M']$. Algorithm 3 iteratively considers an M'-exposed vertex u_i, and computes a feasible M'-augmenting walk U starting at u_i, if one exists. Lemma 5.1 implies that $\eta(U)$ is a feasible M-augmenting walk in G. Theorem 5.3 implies that we need to remove at least one vertex of the walk $\eta(U)$ to stabilize the instance. Note that every vertex $a \neq u_i, v_j$ of U is M'-covered, and hence, $\eta(a)$ is M-covered. Therefore, the only vertices we can potentially remove are $\eta(u_i)$ or $\eta(v_j)$. Hence, if both $\eta(u_i)$ and $\eta(v_j)$ are M-covered, the instance cannot be stabilized and Algorithm 3 checks this in line 10. If only one among $\eta(u_i)$ and $\eta(v_j)$ is M-covered, then necessarily we have to remove the M-exposed vertex among the two. Algorithm 3 checks this in line 12 and 14. Note that, by Remark 5.1, instead of computing a new auxiliary for the modified G, we can just remove $C_{\eta(u_i)}$ (resp. $C_{\eta(v_j)}$) from G'. Similarly, if $\eta(u_i) = \eta(v_j)$ and $\eta(u_i)$ is M-exposed, we necessarily have to remove $\eta(u_i)$. Algorithm 3 checks this in line 17. If instead $\eta(u_i) \neq \eta(v_j)$, and both are M'-exposed, we apply Lemma 5.2(b) to find a basic augmenting structure W contained in U. Once again, we know by Lemma 5.1 and Theorem 5.3 that we need to remove a vertex in $\eta(W)$. In case W is a cycle or bi-cycle, all vertices of $\eta(W)$ are M-covered so the instance cannot be stabilized and Algorithm 3 checks this in line 21. In case W is a M'-augmenting flower with base u_i or v_j, Algorithm 3 accordingly removes $\eta(u_i)$ or $\eta(v_j)$ as all other vertices in $\eta(W)$ are M-covered, in line 24 and 26. If W is a proper (because $\eta(u_i) \neq \eta(v_j)$) M'-augmenting path and M has maximum weight in G, by Lemma 5.4 we know that we can find a feasible M-augmenting walk, where the only M-exposed vertex is either $\eta(u_i)$ or $\eta(v_j)$. Once again, this implies that this vertex must be removed. Algorithm 3 does so in lines 31 and 33. Finally, if W is a proper M'-augmenting path and M does not have

Algorithm 3 finding an M-vertex-stabilizer

input: $[(G, w, c), M]$
let $M_{\max}$ indicate if M has maximum weight in G
compute the auxiliary $[(G', w', 1), M']$
initialize $S \leftarrow \emptyset$, $L \leftarrow M'$-exposed vertices
while $L \neq \emptyset$ **do**
  select $u_i \in L$ and compute a feasible M'-augmenting walk starting at u_i using Lemma 5.2(a)
  if no such walk exists **then**
    $L \leftarrow L \setminus \{u_i\}$
  else
    consider the computed feasible M'-augmenting $u_i v_j$-walk
    if both $\eta(u_i)$ and $\eta(v_j)$ are M-covered **then**
      return infeasible
    else if $\eta(u_i)$ is M-covered and $\eta(v_j)$ is not **then**
      $S \leftarrow S \cup \eta(v_j)$, $G \leftarrow G \setminus \eta(v_j)$, $G' \leftarrow G' \setminus C_{\eta(v_j)}$, $L \leftarrow L \setminus C_{\eta(v_j)}$
    else if $\eta(v_j)$ is M-covered and $\eta(u_i)$ is not **then**
      $S \leftarrow S \cup \eta(u_i)$, $G \leftarrow G \setminus \eta(u_i)$, $G' \leftarrow G' \setminus C_{\eta(u_i)}$, $L \leftarrow L \setminus C_{\eta(u_i)}$
    else
      if $\eta(u_i) = \eta(v_j)$ **then**
        $S \leftarrow S \cup \eta(u_i)$, $G \leftarrow G \setminus \eta(u_i)$, $G' \leftarrow G' \setminus C_{\eta(u_i)}$, $L \leftarrow L \setminus C_{\eta(u_i)}$
      else
        find a basic M'-augmenting structure W contained in the $u_i v_j$-walk using Lemma 5.2(b)
        if W is an M'-augmenting cycle or bi-cycle **then**
          return infeasible
        else if W is an M'-augmenting flower rooted at u_i **then**
          $S \leftarrow S \cup \eta(u_i)$, $G \leftarrow G \setminus \eta(u_i)$, $G' \leftarrow G' \setminus C_{\eta(u_i)}$, $L \leftarrow L \setminus C_{\eta(u_i)}$
        else if W is an M'-augmenting flower rooted at v_j **then**
          $S \leftarrow S \cup \eta(v_j)$, $G \leftarrow G \setminus \eta(v_j)$, $G' \leftarrow G' \setminus C_{\eta(v_j)}$, $L \leftarrow L \setminus C_{\eta(v_j)}$
        else if W is a proper M'-augmenting $u_i v_j$-path **then**
          if $M_{\max}$ **then**
            compute $\mathrm{tb}(\eta(W), \eta(u_i))$ and $\mathrm{tb}(\eta(W), \eta(v_j))$
            if $\mathrm{tb}(\eta(W), \eta(u_i))$ is M-augmenting **then**
              $S \leftarrow S \cup \eta(u_i)$, $G \leftarrow G \setminus \eta(u_i)$, $G' \leftarrow G' \setminus C_{\eta(u_i)}$, $L \leftarrow L \setminus C_{\eta(u_i)}$
            else if $\mathrm{tb}(\eta(W), \eta(v_j))$ is M-augmenting **then**
              $S \leftarrow S \cup \eta(v_j)$, $G \leftarrow G \setminus \eta(v_j)$, $G' \leftarrow G' \setminus C_{\eta(v_j)}$, $L \leftarrow L \setminus C_{\eta(v_j)}$
            end if

35: **else**
36: $S \leftarrow S \cup \{\eta(u_i), \eta(v_j)\}$, $G \leftarrow G \setminus \{\eta(u_i), \eta(v_j)\}$, $G' \leftarrow G' \setminus \left(C_{\eta(u_i)} \cup C_{\eta(v_j)}\right)$, $L \leftarrow L \setminus \left(C_{\eta(u_i)} \cup C_{\eta(v_j)}\right)$
37: **end if**
38: **end if**
39: **end if**
40: **end if**
41: **end if**
42: **end while**
43: **if** $w(M) < \nu_f^c(G)$ **then**
44: **return** infeasible
45: **else**
46: **return** S
47: **end if**

maximum weight in G, Algorithm 3 removes both $\eta(u_i)$ and $\eta(v_j)$ in line 36, even though it might only be necessary to remove one of them.

From the discussion so far, it follows that when we exit the while loop each vertex in S is either a necessary vertex to be removed from G, in order to stabilize the instance, or it is one of two vertices for which it is necessary to remove at least one. Therefore, for any M-vertex-stabilizer S^* we have $|S^*| \geq \frac{1}{2}|S|$. It follows that Algorithm 3 is a 2-approximation algorithm. Furthermore, if M has maximum weight in G, then each vertex in S is a necessary vertex to be removed from G, in order to stabilize the instance. It follows that Algorithm 3 is exact in this case.

We now argue that either removing all vertices in S is also sufficient, or the instance cannot be stabilized. Suppose that the M-vertex-stabilizer instance given by $G \setminus S$ and M is not stable. Theorem 5.1 implies that $(G \setminus S)'$ contains a basic augmenting structure Q. Note that Q cannot be an M'-augmenting flower with exposed root, or a proper M'-augmenting path with at least one exposed endpoint. To see this, observe that a flower and path are feasible M'-augmenting walks of length at most $3|V(G')|$ and $|V(G')|$, respectively. Hence, they would have been found by Algorithm 3 in line 5, contradicting that Q exists in $(G \setminus S)'$. It follows that Q is a basic augmenting structure where all vertices are M'-covered. By Lemma 5.1 $\eta(Q)$ is a feasible M-augmenting walk where all vertices are M-covered. This implies that the instance cannot be stabilized. Furthermore, using the ε-augmentation of $\eta(Q)$ we can obtain a fractional c-matching whose value is strictly greater than $w(M)$. Hence, $w(M) < \nu_f^c(G \setminus S)$. Algorithm 3 correctly determines this in line 43. This proves correctness of our algorithm.

Finally, we argue about the running time of the algorithm. Note that each operation that the algorithm performs can be done in polynomial time. Furthermore, after each iteration of the while loop, we either determine that the instance cannot be stabilized, or remove at least one vertex from L. Therefore, the while loop can be executed at most $|V(G')| \leq n^2$ times. The result follows. $\square$

5.2 M-Edge-Stabilizer

In this section we give our M-edge-stabilizer results. First we give a sketch of the 2-approximation algorithm of Bock et al. [11]. Then, we generalize this algorithm to capacitated graphs and arbitrary given (c-)matchings. On the other hand, we show that a straightforward generalization of this algorithm to weighted graphs does not work.

Let us sketch the idea of the 2-approximation algorithm. Given a graph $G = (V, E)$ and a maximum matching M in G, Bock et al. [11] formulate the following covering linear program to find an M-edge-stabilizer, where $V(M) \subseteq V$ is the set of vertices covered by M.

$$\begin{aligned}
\min \quad & \sum_{e \in E \setminus M} a_e \\
\text{s.t.} \quad & y_u + y_v = 1 \quad \forall uv \in M \\
& y_u + y_v + a_e \geq 1 \quad \forall e = uv \in E \setminus M \text{ and } u, v \in V(M) \\
& y_v + a_e \geq 1 \quad \forall e = uv \in E \setminus M \text{ and } v \in V(M), u \notin V(M) \\
& y \in \mathbb{R}_{\geq 0}^{V(M)}, a \in \mathbb{R}_{\geq 0}^{E \setminus M}
\end{aligned} \tag{5.4}$$

To obtain the 2-approximation algorithm, Bock et al. [11] construct an auxiliary bipartite instance. They then solve (5.4) on this auxiliary instance, and map it back to a 2-approximate solution of the original instance.

Let us take a closer look at (5.4). The first constraint ensures that all edges in M are covered by y. The subsequent two constraints ensure that all edges not in M, but with at least one of its endpoints covered by M, are covered by y and a. Since M is a maximum, and hence maximal, matching, for each edge at least one of its endpoints is covered by M. So these three constraints cover all edges. We can easily allow for arbitrary matchings, by adding a constraint for the edges not in M and with both endpoints exposed by M:

$$a_e \geq 1 \quad \forall e = uv \in E \setminus M \text{ and } u, v \notin V(M).$$

The vector y in (5.4) represents a fractional vertex cover. To include vertex capacities, we replace y by a fractional vertex cover (y, z). Note that y is defined on $V(M)$ instead of on V. This is because the goal is to have y satisfy complementary slackness with M, which means y_v should be zero for all vertices v exposed by M. Similarly, we want that (y,z) satisfies complementary slackness with M, which means y_v should be zero for all vertices v unsaturated by M, and z_e should be zero for all edges not in M. Hence, we define y on $V(M)$, where $V(M)$ is now redefined as the set of vertices saturated by M, and z on M.

So, given a graph $G=(V,E)$ with vertex capacities $c \in \mathbb{Z}^V_{\geq 0}$, and a c-matching M in G, we obtain the following linear program, where $V(M) \subseteq M$ is the set of vertices saturated by M.

$$
\begin{aligned}
\min \quad & \sum_{e \in E \setminus M} a_e \\
\text{s.t.} \quad & y_u + y_v + z_e = 1 \quad \forall e = uv \in M \text{ and } u, v \in V(M) \\
& y_v + z_e = 1 \quad \forall e = uv \in M \text{ and } v \in V(M), u \notin V(M) \\
& z_e = 1 \quad \forall e = uv \in M \text{ and } u, v \notin V(M) \\
& y_u + y_v + a_e \geq 1 \quad \forall e = uv \in E \setminus M \text{ and } u, v \in V(M) \\
& y_v + a_e \geq 1 \quad \forall e = uv \in E \setminus M \text{ and } v \in V(M), u \notin V(M) \\
& a_e \geq 1 \quad \forall e = uv \in E \setminus M \text{ and } u, v \notin V(M) \\
& y \in \mathbb{R}^{V(M)}_{\geq 0}, z \in \mathbb{R}^M_{\geq 0}, a \in \mathbb{R}^{E \setminus M}_{\geq 0}
\end{aligned}
\tag{5.5}
$$

Theorem 5.4. *The M-edge-stabilizer problem on unit-weight, capacitated graphs admits an efficient 2-approximation algorithm.*

Our proof follows the proof of Proposition 3 of Bock et al. [11].

Proof. Consider the linear program (5.5). The first three constraints ensure that each edge in M is covered by y and z and that $|M| = \sum_{v \in V(M)} c_v y_v + \sum_{e \in M} z_e$. The subsequent three constraints ensure that all edges not in M are covered by y and a. Observe that there is one covering constraint for every edge.

If a feasible solution (y,z,a) of (5.5) satisfies $a \in \{0,1\}^{E \setminus M}$, then $F = \{e \in E : a_e = 1\}$ is an M-edge-stabilizer. This is because (y,z) is a fractional vertex cover in $G \setminus F$, M is a c-matching in $G \setminus F$ and $|M| = \sum_{v \in V(M)} c_v y_v + \sum_{e \in M} z_e$.

Likewise, if we have an M-edge-stabilizer F, then we can construct a feasible solution (y,z,a) of (5.5) satisfying $a \in \{0,1\}^{E \setminus M}$ as follows. Set $a_e = 1$ if $e \in F$ and $a_e = 0$ otherwise. Take (y,z) to be a minimum fractional vertex cover in $G \setminus F$. Since $G \setminus F$ is stable, (y,z) and M form a primal-dual optimal pair for $\nu^c_f(G \setminus F)$ and $\tau^c_f(G \setminus F)$. Hence, by complementary slackness, $y_v = 0$ if v is not saturated by M and $z_e = 0$ if e is not in M, and so $\operatorname{supp}(y) \subseteq V(M)$ and $\operatorname{supp}(z) \subseteq M$. Thus, (y,z,a) is a feasible solution to (5.5).

Claim 5.2. *For a bipartite graph $G = (V, E)$ and c-matching M in G, the linear program (5.5) has an integral optimal solution (y^*, z^*, a^*).*

Proof. Let A denote the coefficient matrix of the constraints in (5.5) for G. Then A has the form

$$A = \begin{bmatrix} A' & I_M & I_{E\setminus M} \end{bmatrix},$$

where A' is an $|E| \times |V(M)|$ submatrix of the edge-vertex incidence matrix A_G of G, and I_M and $I_{E\setminus M}$ are $|E|\times|M|$ and $|E|\times|E\setminus M|$ submatrices, respectively, of the $|E| \times |E|$ identity matrix I, after removing the columns corresponding to the edges not in M and the edges in M, respectively. Observe that the matrix $\begin{bmatrix} A_G & I & I \end{bmatrix}$ is totally unimodular since G is bipartite and thus A_G is totally unimodular. Since A is a (column-indexed) submatrix of $\begin{bmatrix} A_G & I & I \end{bmatrix}$, we conclude that A is totally unimodular as well. In addition, the right hand sides of the constraints in (5.5) are integral. Together these imply that there is an integral optimal solution (y^*, z^*, a^*) of (5.5). (See for example Schrijver [47] for properties of totally unimodular matrices.) □

We use the above claim to find an M-edge-stabilizer in G that is at most twice as large as the minimum M-edge-stabilizer by constructing a bipartite graph $G' = (V_1 \cup V_2, E')$, where $V_i = \{v_i : v \in V\}$ and $E' = \{u_1v_2, u_2v_1 : uv \in E\}$, and a c-matching $M' = \{u_1v_2, u_2v_1 : uv \in M\}$ in G'. Let (5.5)' and (5.5) denote the corresponding linear programs for G' and G, respectively.

We first show that a minimum M-edge-stabilizer F in G induces a solution (y', z', a') of (5.5)' with cost $2|F|$ and integral a'. Since $G\setminus F$ is stable, there exists a fractional vertex cover (y, z) in $G \setminus F$ that satisfies complementary slackness with M. Like before, this means that $\mathrm{supp}(y) \subseteq V(M)$ and $\mathrm{supp}(z) \subseteq M$. We set $y'_{v_i} = y_v$ for all $v \in V(M)$ and $i = 1, 2$, $z'_{u_1v_2} = z'_{u_2v_1} = z_{uv}$ for all $uv \in M$, and $a'_{u_1v_2} = a'_{u_2v_1} = 1$ for all $uv \in F$ and zero otherwise. Then (y', z', a') is a feasible solution of (5.5)' with cost $2|F|$ and integral a'.

Next, we show that an optimal integral solution of (5.5)' can be used to find a half-integral solution of (5.5). Let (y^*, z^*, a^*) be an optimal integral solution of (5.5)'. By the previous paragraph, the cost of (y^*, z^*, a^*) is bounded by $2|F|$. We define (y, z, a) by $y_v = \frac{1}{2}(y^*_{v_1} + y^*_{v_2})$ for all $v \in V(M)$, $z_{uv} = \frac{1}{2}(z^*_{u_1v_2} + z^*_{u_2v_1})$ for all $uv \in M$, and $a_{uv} = \max\{a^*_{u_1v_2}, a^*_{u_2v_1}\}$ for all $uv \in E \setminus M$. This defines a feasible solution for (5.5), which can be seen as follows. For $uv \in M$ and $u, v \in V(M)$, we have

$$\begin{aligned} y_u + y_v + z_{uv} &= \tfrac{1}{2}\left(y^*_{u_1} + y^*_{u_2}\right) + \tfrac{1}{2}\left(y^*_{v_1} + y^*_{v_2}\right) + \tfrac{1}{2}\left(z^*_{u_1v_2} + z^*_{u_2v_1}\right) \\ &= \tfrac{1}{2}\left(y^*_{u_1} + y^*_{v_2} + z^*_{u_1v_2}\right) + \tfrac{1}{2}\left(y^*_{u_2} + y^*_{v_1} + z^*_{u_2v_1}\right) \\ &= \tfrac{1}{2} + \tfrac{1}{2} = 1, \end{aligned}$$

and for $uv \in E \setminus M$ and $u, v \in V(M)$, we have

$$\begin{aligned}
y_u + y_v + a_{uv} &= \tfrac{1}{2}\left(y^*_{u_1} + y^*_{u_2}\right) + \tfrac{1}{2}\left(y^*_{v_1} + y^*_{v_2}\right) + \max\left\{a^*_{u_1v_2}, a^*_{u_2v_1}\right\} \\
&\geq \tfrac{1}{2}\left(y^*_{u_1} + y^*_{u_2}\right) + \tfrac{1}{2}\left(y^*_{v_1} + y^*_{v_2}\right) + \tfrac{1}{2}\left(a^*_{u_1v_2} + a^*_{u_2v_1}\right) \\
&= \tfrac{1}{2}\left(y^*_{u_1} + y^*_{v_2} + a^*_{u_1v_2}\right) + \tfrac{1}{2}\left(y^*_{u_2} + y^*_{v_1} + a^*_{u_2v_1}\right) \\
&\geq \tfrac{1}{2} + \tfrac{1}{2} = 1.
\end{aligned}$$

The cases for $uv \in M$ and $uv \in E \setminus M$ with $v \in V(M)$, $u \notin V(M)$ and with $u, v \notin V(M)$ follow in an analogous manner. As the cost of (y^*, z^*, a^*) is at most $2|F|$, the cost of the solution (y, z, a) of (5.5) is also bounded by $2|F|$. Moreover, a is integral and thus defines an M-edge-stabilizer in G of size at most $2|F|$, for any minimum M-edge-stabilizer F. □

Now we take a look at what happens if we include edge weights. We observed that the vector y in (5.4) represents a fractional vertex cover. In unit-weight graphs fractional vertex covers have to satisfy $y_u + y_v \geq 1$ for all edges $uv \in E$. In weighted graphs this becomes $y_u + y_v \geq w_{uv}$. We can incorporate a similar change in (5.4) to include edge-weights. In particular, we change the right hand side from 1 to w_{uv}, and we multiply a_e by w_e in the constraints, as we still want to have $0 \leq a \leq 1$.

$$\begin{aligned}
\min \quad & \sum_{e \in E \setminus M} a_e \\
\text{s.t.} \quad & y_u + y_v = w_{uv} \quad \forall uv \in M \\
& y_u + y_v + w_e a_e \geq w_e \quad \forall e = uv \in E \setminus M \text{ and } u, v \in V(M) \\
& y_v + w_e a_e \geq w_e \quad \forall e = uv \in E \setminus M \text{ and } v \in V(M), u \notin V(M) \\
& w_e a_e \geq w_e \quad \forall e = uv \in E \setminus M \text{ and } u, v \notin V(M) \\
& y, a \geq 0 \\
& y \in \mathbb{R}^{V(M)}, a \in \mathbb{R}^{E \setminus M}
\end{aligned} \tag{5.6}$$

For the 2-approximation algorithm to work, it is important that on the auxiliary bipartite graph the linear program has an integral optimal solution. This integral solution then maps to a solution in the original graph with integral a, which means we can obtain an M-edge-stabilizer from it. However, now that we have included edge weights, the linear program does not have an integral optimal solution in general, also not on bipartite graphs. In particular, (5.6) has an unbounded integrality gap, which we show with the next example. Note that the matching in the example has maximum weight.

Consider the graph G and matching M given in Figure 5.2. The linear program

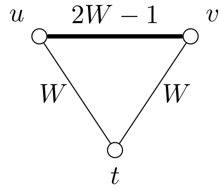


Figure 5.2: Graph G with vertex names next to the vertices, edge weights next to the edges, and a maximum-weight matching $M = \{uv\}$ indicated by the bold edge. W can be any positive (large) real number.

with respect to this graph is given in (5.7).

$$\begin{aligned} \min \quad & a_{tu} + a_{tv} \\ \text{s.t.} \quad & y_u + y_v = 2W - 1 \\ & y_u + W a_{tu} \geq W \\ & y_v + W a_{tv} \geq W \\ & y, a \geq 0 \\ & y \in \mathbb{R}^2, a \in \mathbb{R}^2 \end{aligned} \tag{5.7}$$

Using the constraints, we can find a lower bound on the objective value:

$$\begin{aligned} a_{tu} + a_{tv} &\geq 1 - \tfrac{1}{W} y_u + 1 - \tfrac{1}{W} y_v = 2 - \tfrac{1}{W} \left(y_u + y_v \right) \\ &= 2 - \tfrac{1}{W} \left(2W - 1 \right) = \tfrac{1}{W}. \end{aligned}$$

The solution given by $y_u = y_v = W - \frac{1}{2}$, and $a_{tu} = a_{tv} = \frac{1}{2W}$, is feasible and achieves this lower bound. Hence, fractionally, the optimal objective value is $\frac{1}{W}$. However, when we require integrality, the lower bound of $\frac{1}{W}$ still needs to be satisfied, and so we get an objective value of at least one. Thus, the integrality gap is at least $\frac{1}{1/W} = W$, and we can choose W to be arbitrarily large.

If we create an auxiliary bipartite graph for this instance, we get a lower bound of $\frac{2}{W}$ on the objective. And again, fractionally this can be attained, but integrally this results in a lower bound of one. Hence, the integrality gap is at least $\frac{1}{2}W$.

Part III

Cooperative Matching Games

Chapter 6

Core Separation of 2-Matching Games

In this chapter we discuss the problem of separating over the core of 2-matching games, which we defined before as follows.

Determine if a given allocation $y \in \mathbb{R}^V$ belongs to the core, or find a coalition that violates the corresponding constraint in

$$y(S) \geq \nu^c(G[S]) \text{ for all } S \subset V, \quad y(V) = \nu^c(G). \tag{6.1}$$

In Section 6.1 we prove that separating over the core of 2-matching games is solvable in polynomial time. In Section 6.2 we show that there exists a compact extended formulation that describes the core of 2-matching games.

This chapter is based on [V2].

6.1 Separating over the Core

Theorem 6.1. *Separating over the core of 2-matching games is solvable in polynomial time.*

The first important observation in Biró et al. [10] is that for any $S \subseteq V$, a maximum-weight 2-matching in $G[S]$ is composed of cycles and paths, which means the core of 2-matching games can alternatively be described by the following (smaller) set of constraints:

$$y(V) = \nu^c(G), \tag{6.2a}$$

$$y(C) \geq w(C), \quad \text{for all cycles } C \in \boldsymbol{C}, \tag{6.2b}$$

$$y(P) \geq w(P), \quad \text{for all paths } P \in \boldsymbol{P}. \tag{6.2c}$$

Here, $\boldsymbol{C}$ stands for the set of cycles $C \subseteq E$ in G with $c_v = 2$ for all $v \in V(C)$, and $\boldsymbol{P}$ stands for the set of paths $P \subseteq E$ with $c_v = 2$ for all inner vertices on P. We shortened $y(V(C))$ and $y(V(P))$ to $y(C)$ and $y(P)$, respectively. With this observation, separating over the core for a given vector y reduces to checking whether $y(V) = \nu^c(G)$, which can be done in polynomial time (a maximum-weight c-matching in a graph G can be computed in polynomial time, see for example Letchford et al. [40]), and to separating over the set of constraints for cycles, and the set of constraints for paths.

Biró et al. [10] show how to separate over the set of cycle constraints, by reducing the problem to the *tramp steamer* problem (also known as the minimum cost-to-time ratio problem), which we introduce now. Let $G = (V, E)$ be a graph with edge weights $p, w \in \mathbb{R}^E_{\geq 0}$. The tramp steamer problem is to find a cycle $C \subseteq E$ of G that maximizes the ratio $w(C)/p(C)$. The tramp steamer problem is well-known to be solvable in polynomial time (see for example Dantzig et al. [15], Eiselt and Sandblom [19], and Lawler [38]).

The following lemma is proved in Biró et al. [10] (see the proof of their Theorem 12). We report a proof for completeness.

Lemma 6.1 (Biró et al. [10])**.** *Separating over the constraints for cycles in Equation* (6.2b) *is solvable in polynomial time.*

Proof. Let $V_2 = \{v \in V : c_v = 2\}$ and $G_2 = G[V_2]$. In $G_2 = (V_2, E_2)$ we transfer the given allocations y_v to the edges by setting $p_{uv} = (y_u + y_v)/2$ for all $uv \in E_2$. This defines edge weights $p \in \mathbb{R}^{E_2}$ such that the core constraints for cycles are equivalent to

$$\max_{C \in \boldsymbol{C}} \frac{w(C)}{p(C)} \leq 1. \tag{6.3}$$

Hence we obtained an instance of the tramp steamer problem, which is polynomial-time solvable as mentioned before. Note that by solving the above maximization problem we either find that all the constraints for cycles in Equation (6.2b) are satisfied or we end up with a particular cycle C with $y(C) = p(C) < w(C)$. □

Next, we discuss the flaw related to the separation of the path constraints in Equation (6.2c).

Path separation of Biró et al. [10]. Assuming that all the constraints for cycles in (6.2b) are satisfied by the given vector $y \in \mathbb{R}^V$, Biró et al. [10] process the path constraints separately for all possible endpoints $u_0, v_0 \in V$ (with $u_0 \neq v_0$) and all possible lengths $k = 1, \ldots, n-1$. Let $\mathcal{P}_k(u_0, v_0) \subseteq \mathcal{P}$

denote the set of $u_0 - v_0$-paths of length k in G. They construct an auxiliary graph $G_k(u_0, v_0)$, that is a subgraph of $G \times P_{k+1}$, the product of G with a path of length k. To this end, let $V_2^{(1)}, \dots, V_2^{(k-1)}$ be $k-1$ copies of V_2. The vertex set of $G_k(u_0, v_0)$ is then $\{u_0, v_0\} \cup V_1^{(1)} \cup \dots \cup V_2^{(k-1)}$. Denote the copy of $v \in V_2$ in $V_2^{(r)}$ by $v^{(r)}$. The edges of $G_k(u_0, v_0)$ and their weights $\overline{w}$ are defined as

$$\begin{array}{lll} u_0 v^{(1)} & \text{for } u_0 v \in E & \text{with weight } \overline{w}_{u_0 v} = y_{u_0} + y_v/2 - w_{u_0 v}, \\ u^{(r-1)} v^{(r)} & \text{for } uv \in E & \text{with weight } \overline{w}_{uv} = (y_u + y_v)/2 - w_{uv}, \\ u^{(k-1)} v_0 & \text{for } uv_0 \in E & \text{with weight } \overline{w}_{uv_0} = y_u/2 + y_{v_0} - w_{uv_0}. \end{array}$$

They claim that $y(P) \geq w(P)$ holds for all $P \in \mathcal{P}$ if and only if the shortest $u_0 - v_0$-path in $G_k(u_0, v_0)$ (w.r.t. $\overline{w}$) has weight ≥ 0 for all $u_0 \neq v_0$ and $k = 1, \dots, n-1$.

However, the next example shows that this claim is not true. Consider the graph in Figure 6.1. One can check that $\nu^c(G) = 12$ and that the given

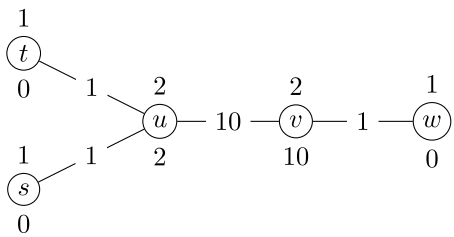


Figure 6.1: Graph with edge weights w on the edges, vertex labels in the vertices, vertex capacities c above the vertices, and an allocation y below the vertices.

allocation y is in the core. In the auxiliary graph $G_4(s, t)$, defined above, there is a path of total weight strictly less than zero: $P = (s, u^{(1)}, v^{(2)}, u^{(3)}, t)$ has weight

$$\overline{w}(P) = y_s + 2y_u + y_v + y_t - w_{su} - 2w_{uv} - w_{ut} = 14 - 22 < 0.$$

According to their claim, this should mean that $y(P) \geq w(P)$ does not hold for all $P \in \mathcal{P}$, that is, y is not a core allocation.

Our path separation. We now show how to fix the above issue, again by relying on the tramp steamer problem.

Lemma 6.2. *Separating over the constraints for paths in Equation (6.2c) is solvable in polynomial time.*

Proof. We assume that all the constraints for cycles in Equation (6.2b) are satisfied by the given vector $y \in \mathbb{R}^V$. We first process all paths of length zero and one separately, that is, the vertices and edges, by checking

$$\begin{aligned} y_v &\geq 0, \quad \text{for all } v \in V, \\ y_u + y_v &\geq w_{uv}, \quad \text{for all } uv \in E. \end{aligned}$$

We process the remaining constraints for paths separately for all possible endpoints $s \neq t \in V$. We create an auxiliary graph $G(s,t)$ as the subgraph of G induced by the vertex set $V_2 \cup \{s,t\}$. We add the edge st to $G(s,t)$ with $w_{st} = 0$, replacing the original edge st if it exists. Like we did for the constraints for cycles, we transfer the given allocations y_v to the edges by setting $p_{uv} = (y_u + y_v)/2$ for all edges uv in $G(s,t)$. If $c_s = c_t = 2$, then we solve the tramp steamer problem on $G(s,t)$ directly. If $c_s = 1$ and $c_t = 2$, then we solve $d_s - 1$ (where d_s is the degree of s in $G(s,t)$) instances of the tramp steamer problem: we remove all edges incident to s in $G(s,t)$ except for st and one other edge e, and solve the tramp steamer problem on this variant of $G(s,t)$. We repeat this for all possible edges e incident to s, unequal to st. The case $c_s = 2$ and $c_t = 1$ can be handled similarly. If $c_s = c_t = 1$, then we solve $(d_s - 1)(d_t - 1)$ instances of the tramp steamer problem: we remove all edges incident to both s and t except st, one other edge e incident to s and one other edge f incident to t, and solve the tramp steamer problem on this variant of $G(s,t)$. We repeat this for all possible combinations of the edges e and f.

Suppose there is a path $P \in \boldsymbol{P}$ of length at least two, such that $y(P) < w(P)$. Let $P = (s; e_1, \ldots, e_k; t)$, $k \geq 2$. There is a variant of $G(s,t)$ which contains both e_1 and e_k. We construct a cycle C in this graph from P: $C = (s; e_1, \ldots, e_k, ts; s)$. Then:

$$\begin{aligned} w(C) &= w_{e_1} + \cdots + w_{e_k} + w_{st} = w(P) + 0 = w(P), \\ p(C) &= p_{e_1} + \cdots + p_{e_k} + p_{st} = y(P), \end{aligned}$$

which means we have $w(C)/p(C) = w(P)/y(P) > 1$. So, solving the tramp steamer problem on this graph, we find that the maximum is > 1.

Suppose we solve the tramp steamer problem on some variant of $G(s,t)$, for some s and t, and find that the maximum is > 1, and that this is attained by the cycle C. It is straightforward to check that we must have $s, t \in V(C)$, and in particular $st \in C$, as otherwise $c_v = 2$ for all $v \in V(C)$, which means $C \in \boldsymbol{C}$, contradicting that all the constraints for cycles in Equation (6.2b) are satisfied. Let P be the st-path obtained from C by removing st. Then:

$$\begin{aligned} w(P) &= w(C) - w_{st} = w(C) - 0 = w(C), \\ y(P) &= p(C), \end{aligned}$$

which means we have $w(P)/y(P) = w(C)/p(C) > 1$. So the constraint for P is violated. $\square$

Note that Lemma 6.1 and Lemma 6.2, together with checking (6.2a), yield a proof of Theorem 6.1.

6.2 Compact Extended Formulation

Theorem 6.2. *There exists a compact extended formulation that describes the core of* 2*-matching games.*

To give a compact extended formulation of the core, we essentially need to rewrite the inequality in Equation (6.3) in a compact form.

Suppose we are given a graph $G' = (V', E')$, with edge weights $p, w \in \mathbb{R}^{E'}_{\geq 0}$, and that we want to check whether this graph satisfies the inequality in Equation (6.3). As a first step, define the edge costs $c_{uv} = p_{uv} - w_{uv}$ for all edges $uv \in E'$. Note that Equation (6.3) is violated if and only if G' contains a negative cost cycle C with respect to c, since

$$w(C)/p(C) > 1 \iff 0 > \sum_{uv \in C} c_{uv} = \sum_{uv \in C} (p_{uv} - w_{uv}) = p(C) - w(C).$$

We therefore focus on checking if a graph $G' = (V', E')$ with edge costs $c \in \mathbb{R}^{E'}$ contains a negative cost cycle. There are several efficient ways to detect negative cost cycles. For example one can rely on the notion of potential in undirected graphs with general edge weights, as described by Sebő [48]. A combinatorial algorithm also follows from Dudycz and Katarzyna [18] (that actually works for a broader class of generalized matching problems). For our extended formulation, we rely on the LP formulation designed by Barahona [6] for finding negative cost cycles. Recall that a cut $B \subseteq E'$ is a set of edges of the form $\delta(X)$ for some $\emptyset \neq X \subset V'$. We need the following theorem.

Theorem 6.3 (Seymour [49])**.** *The cone generated by the incidence vectors of the cycles of a graph is defined by the system*

$$\begin{aligned} &x_e - x(B \setminus e) \leq 0, \ \textit{for each cut } B, \ \textit{for every edge } e \in B, \\ &x \geq 0. \end{aligned}$$

Using the system of constraints from Theorem 6.3, we can design an LP formulation as in Section 3 in Barahona [6], where they define the LP below with the goal of minimizing the cost of a cycle. (In Barahona [6] they also add the constraint $\sum_{e\in E'} x_e = 1$, because they are interested in cycles of minimum mean weight, but here this constraint is not needed.)

$$\begin{aligned} \min \quad & \sum_{e\in E'} c_e x_e \\ \text{s.t.} \quad & x_e - x(B\setminus e) \le 0, \ \text{for each cut } B, \ \text{for every edge } e \in B \\ & x \ge 0 \end{aligned} \tag{6.4}$$

One observes that G' contains a negative cost cycle if and only if there exists a solution to this LP whose objective value is negative (indeed, the LP in this case is unbounded). This is easily seen as if C is a cycle with negative cost, its characteristic vector x^C yields an LP solution with negative objective value. Vice versa, if x^* is a feasible solution for the LP with negative objective value, by Theorem 6.3 x^* can be expressed as a conic combination of cycles, implying that at least one such cycle must have negative cost.

To make the above LP compact, we rely on flows. Recall that an st-cut is a set of edges of the form $\delta(X)$ where X contains exactly one of s and t. For a fixed edge $\overline{e} = st$, the system of inequalities consisting of $x \ge 0$ and the inequalities of (6.4) for $\overline{e}$, is then equivalent to

$$\begin{aligned} & x(B) \ge x_{\overline{e}}, \ \text{for each } st\text{-cut } B \text{ in } G'\setminus\overline{e}, \\ & x \ge 0, \end{aligned} \tag{6.5}$$

which tells us that all st-cuts in $G'\setminus\overline{e}$ have capacity (w.r.t. x) at least $x_{\overline{e}}$. By the max flow min cut theorem of Ford and Fulkerson [24], this is equivalent to the existence of a st-flow (w.r.t. the capacity function x) in $G'\setminus\overline{e}$ of value $x_{\overline{e}}$. Therefore there exists an x that satisfies the constraints in Equation (6.5) if and only if there exists a pair (x, y) that satisfies

$$\begin{aligned} & \sum_{v:uv\in E'\setminus\overline{e}} (y_{uv} - y_{vu}) = \begin{cases} 0, & \text{if } u \ne s, u \ne t, \\ x_{\overline{e}}, & \text{if } u = s, \\ -x_{\overline{e}}, & \text{if } u = t, \end{cases} \qquad \text{for all } u \in V', \\ & 0 \le y_{uv}, y_{vu} \le x_e, \qquad \text{for all } e = uv \in E'\setminus\overline{e}. \end{aligned}$$

Finally, we can rewrite the LP in Equation (6.4) as

$$\begin{aligned} \min \quad & \sum_{e\in E'} c_e x_e, \\ \text{s.t.} \quad & \sum_{v:uv\in E'\setminus\overline{e}} (y^{\overline{e}}_{uv} - y^{\overline{e}}_{vu}) = \begin{cases} 0, & \text{if } u \ne s, u \ne t, \\ x_{\overline{e}}, & \text{if } u = s, \\ -x_{\overline{e}}, & \text{if } u = t, \end{cases} \\ & \qquad\qquad \text{for all } u \in V' \text{ and } \overline{e} = st \in E', \\ & 0 \le y^{\overline{e}}_{uv}, y^{\overline{e}}_{vu} \le x_e, \qquad \text{for all } e = uv \in E'\setminus\overline{e} \text{ and } \overline{e} \in E'. \end{aligned}$$

The dual of this LP is

$$
\begin{aligned}
&\max \quad 0\\
&\text{s.t.} \quad \left.\begin{aligned} \gamma_u^{\overline{e}} - \gamma_v^{\overline{e}} - \lambda_{uv}^{\overline{e}} \\ \gamma_v^{\overline{e}} - \gamma_u^{\overline{e}} - \lambda_{vu}^{\overline{e}} \end{aligned}\right\} \leq 0, \qquad \text{for all } uv \in E' \setminus \overline{e} \text{ and } \overline{e} \in E',\\
&\qquad \gamma_t^{\overline{e}} - \gamma_s^{\overline{e}} + \sum_{uv \in E' \setminus \overline{e}} \left(\lambda_{uv}^{\overline{e}} + \lambda_{vu}^{\overline{e}}\right) \leq c_{\overline{e}}, \qquad \text{for all } \overline{e} = st \in E',\\
&\qquad \lambda_{uv}^{\overline{e}}, \lambda_{vu}^{\overline{e}} \geq 0, \qquad \text{for all } uv \in E' \setminus \overline{e} \text{ and } \overline{e} \in E'.
\end{aligned}
\tag{6.6}
$$

Combining all of the steps, a graph G' satisfies the constraint in Equation (6.3) if and only if the LP in Equation (6.6) attains a feasible solution.

We are now ready to state the compact extended formulation. For convenience, let $\mathcal{G}$ be the set of graphs consisting of G_2 and all variants of $G(s,t)$ for all $s \neq t \in V$. From Section 6.1, we know we need to check $y(V) = \nu^c(G)$, $y_v \geq 0$ for all $v \in V$, $y_u + y_v \geq w_{uv}$ for all $uv \in E$, and that all graphs $G' = (V', E') \in \mathcal{G}$ satisfy the constraint in Equation (6.3). So, in total we get

$$
\begin{aligned}
&y(V) = \nu^c(G),\\
&y_v \geq 0, \quad \text{for all } v \in V,\\
&y_u + y_v \geq w_{uv}, \quad \text{for all } uv \in E,\\
&\left.\begin{aligned}
&\left.\begin{aligned} \gamma_u^{\overline{e}} - \gamma_v^{\overline{e}} - \lambda_{uv}^{\overline{e}} \\ \gamma_v^{\overline{e}} - \gamma_u^{\overline{e}} - \lambda_{vu}^{\overline{e}} \end{aligned}\right\} \leq 0, \qquad \text{for all } uv \in E' \setminus \overline{e} \text{ and } \overline{e} \in E',\\
&\gamma_t^{\overline{e}} - \gamma_s^{\overline{e}} + \sum_{uv \in E' \setminus \overline{e}} \lambda_{uv}^{\overline{e}} + \lambda_{vu}^{\overline{e}} \leq (y_s + y_t)/2 - w_{\overline{e}}, \qquad \text{for all } \overline{e} = st \in E',\\
&\lambda_{uv}^{\overline{e}}, \lambda_{vu}^{\overline{e}} \geq 0, \qquad \text{for all } uv \in E' \setminus \overline{e} \text{ and } \overline{e} \in E'.
\end{aligned}\right\}\\
&\qquad\qquad\qquad\qquad \text{for all } \; G' = (V', E') \in \mathcal{G}.
\end{aligned}
\tag{6.7}
$$

The size of this formulation easily follows from the size of $\mathcal{G}$. Consider the graph $G(s,t)$ for some $s \neq t \in V$. In the worst case, $c_s = c_t = 1$, which means there are $(d_s - 1)(d_t - 1)$ variants of $G(s,t)$. Therefore

$$|\mathcal{G}| \leq 1 + \sum_{s \neq t \in V} (d_s - 1)(d_t - 1) = O(n^4).$$

The formulation in Equation (6.7) has

$$n + |\mathcal{G}| \cdot (O(m^2) + O(nm)) = O(n^4 m^2) + O(n^5 m)$$

variables, and

$$1 + n + m + |\mathcal{G}| \cdot (O(m^2) + O(m) + O(m^2)) = O(n^4 m^2)$$

constraints, that is, it has polynomial size. This proves Theorem 6.2.

Chapter 7

Two-Stage Assignment Games

In this chapter we discuss the two-stage stochastic assignment game, which we defined before as

$$\min_{y \in \mathrm{core}(G_0)} \mathbb{E}_{S \sim \mathcal{D}} \left[\min_{y^S \in \mathrm{core}(G_S)} \sum_{v \in V_0 \cap V_S} \lambda_v \left[y_v - y_v^S \right]^+ \right]. \tag{2SAG}$$

In Sections 7.1 and 7.2 we consider the two-stage stochastic assignment game when the probability distribution $\mathcal{D}$ is given explicitly and implicitly, respectively. Lastly, we discuss the multistage vertex cover problem in Section 7.3, which is related to the two-stage stochastic assignment game when the probability distribution is given explicitly.

This chapter is based on [V5].

Preliminaries Let $\tau(G)$ be the value of a minimum integral vertex cover, that is, $\tau_f(G)$ with the additional requirement $y \in \mathbb{Z}^V$. It follows from Königs theorem ($\nu(G) = \tau(G)$ for bipartite graphs) and LP theory ($\nu(G) \leq \nu_f(G) = \tau_f(G) \leq \tau(G)$) that $\tau_f(G) = \nu(G)$ for bipartite graphs. Shapley and Shubik [50] showed that the core of an assignment game is precisely the set of minimum fractional vertex covers, that is,

$$\mathrm{core}(G) = \left\{ y \in \mathbb{R}^V_{\geq 0} : y_u + y_v \geq 1 \ \forall uv \in E, 1^\top y = \nu(G) \right\}.$$

From this and $\tau_f(G) = \nu(G)$ it readily follows that the core of each assignment game is nonempty: there is always a minimum fractional vertex cover. This is important for our two-stage stochastic assignment game, because we are assured that in both stages the core is nonempty.

Observe that in any core element $y \leq 1$, because $y_u + y_v \geq 1$ for all edges in a maximum matching M, $1^\top y = \nu(G) = |M|$, and $y \geq 0$.

7.1 Explicit Distribution

Suppose the distribution $\mathcal{D}$ is given explicitly by a list of scenarios $\mathcal{S}$ and their respective probabilities of occurrence $\{p_S\}_{S\in\mathcal{S}}$. Here we consider the problem of minimizing the absolute difference, instead of the positive difference, that is, $|y_v - y_v^S|$ instead of $[y_v - y_v^S]^+$. We solve this in such a way that one can later choose either option, or even $[y_v^S - y_v]^+$. Using the scenarios $\mathcal{S}$, we can expand the expectation in (2SAG):

$$\min_{y\in\text{core}(G_0)} \sum_{S\in\mathcal{S}} p_S \left[\min_{y^S\in\text{core}(G_S)} \sum_{v\in V_0\cap V_S} \lambda_v \left| y_v - y_v^S \right| \right]. \tag{2SAG-expl}$$

We can rewrite this as the following LP.

$$\begin{aligned}
\min \quad & \sum_{S\in\mathcal{S}} p_S \sum_{v\in V_0\cap V_S} \lambda_v(\delta_v^S + d_v^S) \\
\text{s.t.} \quad & y_u + y_v \geq 1 \quad \forall uv \in E_0 \\
& 1^\top y = \nu(G_0) \\
& y \in \mathbb{R}_{\geq 0}^{V_0} \\
& y_u^S + y_v^S \geq 1 \quad \forall uv \in E_S, \forall S\in\mathcal{S} \\
& 1^\top y^S = \nu(G_S) \quad \forall S\in\mathcal{S} \\
& y^S \in \mathbb{R}_{\geq 0}^{V_S} \quad \forall S\in\mathcal{S} \\
& y_v - y_v^S \leq \delta_v^S \quad \forall v\in V_0\cap V_S, \forall S\in\mathcal{S} \\
& y_v^S - y_v \leq d_v^S \quad \forall v\in V_0\cap V_S, \forall S\in\mathcal{S} \\
& \delta^S \in \mathbb{R}_{\geq 0}^{V_0\cap V_S} \quad \forall S\in\mathcal{S} \\
& d^S \in \mathbb{R}_{\geq 0}^{V_0\cap V_S} \quad \forall S\in\mathcal{S}
\end{aligned} \tag{2SAG-LP}$$

Observe that if we change $\delta_v^S + d_v^S$ in the objective to δ_v^S or d_v^S, then the objective we consider is $[y_v - y_v^S]^+$ or $[y_v^S - y_v]^+$, respectively.

Let $V = V_0 \cup \bigcup_{S\in\mathcal{S}} V_S$. This LP has $O(|V||\mathcal{S}|)$ variables and $O(|V|^2|\mathcal{S}|)$ constraints. So it has size polynomial in the input size, which means we can solve (2SAG-expl) in polynomial time, by solving (2SAG-LP). Using an auxiliary linear program we show that the feasible region of (2SAG-LP) is an integral polyhedron.

Theorem 7.1. *The feasible region of* (2SAG-LP) *is an integral polyhedron.*

We consider (2SAG-LP) with an arbitrary objective, and show that for each objective we can find an integral optimal solution. Since for each extreme point of a polyhedron there is an objective such that the extreme point is the unique optimal solution, this proves that the polyhedron is integral. We consider the following objective, where α can take any value, $\beta \geq 0$ and $b \geq 0$. Note that if $\beta_v^S < 0$ or $b_v^S < 0$ for any $v \in V_0 \cap V_S$, $S \in \mathcal{S}$, then the LP becomes unbounded, and so there are no extreme points in those directions.

$$\min \quad \sum_{v \in V_0} \alpha_v y_v + \sum_{S \in \mathcal{S}} \sum_{v \in V_S} \alpha_v^S y_v^S + \sum_{S \in \mathcal{S}} \sum_{v \in V_0 \cap V_S} \beta_v^S \delta_v^S + b_v^S d_v^S$$

We will formulate a maximum flow problem on an auxiliary graph, and compute its dual LP. It is known that the constraint matrix of a maximum flow LP is totally unimodular (TU). The dual LP has the transpose as constraint matrix, which is then also TU. Since in addition the right hand side of the dual constraints are integral, this means the feasible region of the dual is an integral polyhedron. (See for example Schrijver [47] for properties of TU matrices.) Finally, we show that we can map an optimal dual solution to an optimal solution for (2SAG-LP).

Let $V_1, V_2 \subseteq V_0 \cup \bigcup_{S \in \mathcal{S}} V_S$ be the bipartition of $G_0 \cup \bigcup_{S \in \mathcal{S}} G_S$. Let

$$\varepsilon = \frac{1}{1 + \sum_{v \in V_0} |\alpha_v| + \sum_{S \in \mathcal{S}} \sum_{v \in V_S} |\alpha_v^S| + \sum_{S \in \mathcal{S}} \sum_{v \in V_0 \cap V_S} \beta_v^S + b_v^S}.$$

Note that $\varepsilon > 0$. We create the auxiliary graph $G' = (V', A)$ as follows. Let

$$V' = \{s, t\} \cup V_0 \cup \left\{ v^S : v \in V_S, S \in \mathcal{S} \right\}.$$

The arc set A contains the following arcs:

- sv for all $v \in V_0 \cap V_1$, with flow-capacity $1 + \varepsilon \alpha_v$;
- sv^S for all $v \in V_S \cap V_1$, $S \in \mathcal{S}$, with flow-capacity $1 + \varepsilon \alpha_v^S$;
- $v^S v$ for all $v \in (V_0 \cap V_S) \cap V_1$, $S \in \mathcal{S}$, with flow-capacity $\varepsilon \beta_v^S$;
- vv^S for all $v \in (V_0 \cap V_S) \cap V_1$, $S \in \mathcal{S}$, with flow-capacity εb_v^S;
- vt for all $v \in V_0 \cap V_2$, with flow-capacity $1 + \varepsilon \alpha_v$;
- $v^S t$ for all $v \in V_S \cap V_2$, $S \in \mathcal{S}$, with flow-capacity $1 + \varepsilon \alpha_v^S$;
- vv^S for all $v \in (V_0 \cap V_S) \cap V_2$, $S \in \mathcal{S}$, with flow-capacity $\varepsilon \beta_v^S$;
- $v^S v$ for all $v \in (V_0 \cap V_S) \cap V_2$, $S \in \mathcal{S}$, with flow-capacity εb_v^S;
- uv for all $uv \in E_0$, such that $u \in V_1$, $v \in V_2$, without upperbound on the flow-capacity;
- $u^S v^S$ for all $uv \in E_S$, $S \in \mathcal{S}$, such that $u \in V_1$, $v \in V_2$, without upperbound on the flow-capacity.

(a) Example of the source/V_1 side. (b) Example of the sink/V_2 side.

Figure 7.1: Example of part of the auxiliary graph, where dashed arcs indicate the edges corresponding with E_0 and E_S.

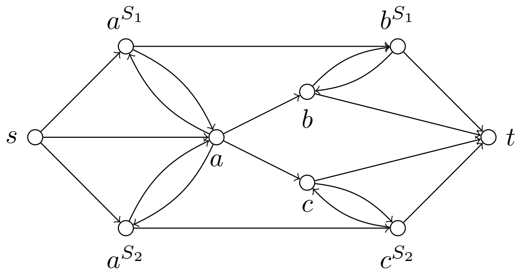


Figure 7.2: Example of the auxiliary graph for the instance given by $V_0 = \{a, b, c\}$, $E_0 = \{ab, ac\}$, and $\mathcal{S} = \{S_1, S_2\}$ where $V_{S_1} = \{a, b\}$, $E_{S_1} = \{ab\}$, and $V_{S_2} = \{a, c\}$, $E_{S_2} = \{ac\}$.

Figure 7.1 shows an example of the auxiliary graph in general, and Figure 7.2 shows an example of the auxiliary graph of a specific instance.

Let $\mathbb{1}[...]$ be 1 if the statement in between the brackets is true, and 0 if the statement is false. We formulate the maximum flow problem in G' as an LP in Equation (7.1). For this LP to have a feasible solution, the flow-capacities need to be nonnegative: $1 + \varepsilon\alpha_v \geq 0$ for all $v \in V_0$, $1 + \varepsilon\alpha_v^S \geq 0$ for all $v \in V_S$, $S \in \mathcal{S}$, $\varepsilon\beta_v^S \geq 0$ for all $v \in V_0 \cap V_S$, $S \in \mathcal{S}$, and $\varepsilon b_v^S \geq 0$ for all $v \in V_0 \cap V_S$, $S \in \mathcal{S}$. The latter two are satisfied as $\varepsilon > 0$ and we set $\beta, b \geq 0$. For any $v \in V_0$, we have

$$\alpha_v \geq -|\alpha_v| \geq -\left(1 + \sum_{v \in V_0} |\alpha_v| + \sum_{S \in \mathcal{S}} \sum_{v \in V_S} |\alpha_v^S| + \sum_{S \in \mathcal{S}} \sum_{v \in V_0 \cap V_S} \beta_v^S + b_v^S\right) = -\frac{1}{\varepsilon},$$

and so, since $\varepsilon > 0$, we have $1 + \varepsilon\alpha_v \geq 1 + \varepsilon\frac{-1}{\varepsilon} = 0$. The same argument works for α_v^S for any $v \in V_S$, $S \in \mathcal{S}$.

The dual of the flow LP in Equation (7.1) is given in Equation (7.2).

Lemma 7.1. *The feasible region of* (7.2) *is an integral polyhedron.*

$$
\begin{aligned}
\max \quad & \sum_{v \in V_0 \cap V_1} f_{sv} + \sum_{S \in \mathcal{S}} \sum_{v \in V_S \cap V_1} f_{sv^S} \\
\text{s.t.} \quad & f_{sv} + \sum_{S \in \mathcal{S} : v \in V_S} (f_{v^S v} - f_{vv^S}) - \sum_{u : uv \in E_0} f_{vu} = 0 \quad \forall v \in V_0 \cap V_1 \\
& f_{sv^S} + \mathbb{1}[v \in V_0](f_{vv^S} - f_{v^S v}) - \sum_{u : uv \in E_S} f_{v^S u^S} = 0 \\
& \hspace{20em} \forall v \in V_S \cap V_1, \forall S \in \mathcal{S} \\
& \sum_{u : uv \in E_0} f_{uv} + \sum_{S \in \mathcal{S} : v \in V_S} (f_{v^S v} - f_{vv^S}) - f_{vt} = 0 \quad \forall v \in V_0 \cap V_2 \\
& \sum_{u : uv \in E_S} f_{u^S v^S} + \mathbb{1}[v \in V_0](f_{vv^S} - f_{v^S v}) - f_{v^S t} = 0 \\
& \hspace{20em} \forall v \in V_S \cap V_2, \forall S \in \mathcal{S} \\
& f_{sv} \leq 1 + \varepsilon \alpha_v \quad \forall v \in V_0 \cap V_1 \\
& f_{sv^S} \leq 1 + \varepsilon \alpha_v^S \quad \forall v \in V_S \cap V_1, \forall S \in \mathcal{S} \\
& f_{vt} \leq 1 + \varepsilon \alpha_v \quad \forall v \in V_0 \cap V_2 \\
& f_{v^S t} \leq 1 + \varepsilon \alpha_v^S \quad \forall v \in V_S \cap V_2, \forall S \in \mathcal{S} \\
& f_{v^S v} \leq \varepsilon \beta_v^S \quad \forall v \in (V_0 \cap V_S) \cap V_1, \forall S \in \mathcal{S} \\
& f_{vv^S} \leq \varepsilon b_v^S \quad \forall v \in (V_0 \cap V_S) \cap V_1, \forall S \in \mathcal{S} \\
& f_{vv^S} \leq \varepsilon \beta_v^S \quad \forall v \in (V_0 \cap V_S) \cap V_2, \forall S \in \mathcal{S} \\
& f_{v^S v} \leq \varepsilon b_v^S \quad \forall v \in (V_0 \cap V_S) \cap V_2, \forall S \in \mathcal{S} \\
& f \in \mathbb{R}^A_{\geq 0}
\end{aligned}
\tag{7.1}
$$

$$
\begin{aligned}
\min \quad & \sum_{v \in V_0} y_v + \sum_{S \in \mathcal{S}} \sum_{v \in V_S} y_v^S \\
& \qquad + \varepsilon \left(\sum_{v \in V_0} \alpha_v y_v + \sum_{S \in \mathcal{S}} \sum_{v \in V_S} \alpha_v^S y_v^S + \sum_{S \in \mathcal{S}} \sum_{v \in V_0 \cap V_S} \beta_v^S \delta_v^S + b_v^S d_v^S \right) \\
\text{s.t.} \quad & \gamma_v + y_v \geq 1 \quad \forall v \in V_0 \cap V_1 \\
& \gamma_v^S + y_v^S \geq 1 \quad \forall v \in V_S \cap V_1, \forall S \in \mathcal{S} \\
& \gamma_v - \gamma_v^S + \delta_v^S \geq 0 \quad \forall v \in (V_0 \cap V_S) \cap V_1, \forall S \in \mathcal{S} \\
& \gamma_v^S - \gamma_v + d_v^S \geq 0 \quad \forall v \in (V_0 \cap V_S) \cap V_1, \forall S \in \mathcal{S} \\
& \gamma_v - \gamma_u \geq 0 \quad \forall uv \in E_0 \text{ such that } u \in V_1, v \in V_2 \\
& \gamma_v^S - \gamma_u^S \geq 0 \quad \forall uv \in E_S \text{ such that } u \in V_1, v \in V_2, \forall S \in \mathcal{S} \\
& -\gamma_v + y_v \geq 0 \quad \forall v \in V_0 \cap V_2 \\
& -\gamma_v^S + y_v^S \geq 0 \quad \forall v \in V_S \cap V_2, \forall S \in \mathcal{S} \\
& \gamma_v^S - \gamma_v + \delta_v^S \geq 0 \quad \forall v \in (V_0 \cap V_S) \cap V_2, \forall S \in \mathcal{S} \\
& \gamma_v - \gamma_v^S + d_v^S \geq 0 \quad \forall v \in (V_0 \cap V_S) \cap V_2, \forall S \in \mathcal{S} \\
& y \in \mathbb{R}_{\geq 0}^{V_0} \\
& y^S \in \mathbb{R}_{\geq 0}^{V_S} \quad \forall S \in \mathcal{S} \\
& \delta^S \in \mathbb{R}_{\geq 0}^{V_0 \cap V_S} \quad \forall S \in \mathcal{S} \\
& d^S \in \mathbb{R}_{\geq 0}^{V_0 \cap V_S} \quad \forall S \in \mathcal{S} \\
& \gamma \in \mathbb{R}^{V_0} \\
& \gamma^S \in \mathbb{R}^{V_S} \quad \forall S \in \mathcal{S}
\end{aligned}
\tag{7.2}
$$

Proof. The constraint matrix of a maximum flow LP is TU; in particular the constraint matrix of (7.1) is TU.

The transpose of a TU matrix is TU. The constraint matrix of a dual LP is the transpose of the constraint matrix of the primal LP. Together they imply that the constraint matrix of (7.2) is TU.

Finally, a polyhedron with a TU constraint matrix and integral right hand side is an integral polyhedron; in particular, the feasible region of (7.2) is an integral polyhedron. □

We can obtain an integral optimal solution for (7.2) by solving (7.2) directly, or combinatorially, as follows. First we find an optimal flow in the auxiliary graph. The flow can then be used to obtain an optimal solution for (7.2), by using complementary slackness. Finally, if this is not an integral solution, then in particular it is not an extreme point solution. So we can use this solution to go to an extreme point solution, which is integral by Lemma 7.1.

We map an integral optimal solution for (7.2) to an integral optimal solution for (2SAG-LP) in two steps. First we map it to the following LP.

$$
\begin{aligned}
\min \quad & 1^\top y + \sum_{S \in \mathcal{S}} 1^\top y^S \\
& \qquad + \varepsilon \left(\sum_{v \in V_0} \alpha_v y_v + \sum_{S \in \mathcal{S}} \sum_{v \in V_S} \alpha_v^S y_v^S + \sum_{S \in \mathcal{S}} \sum_{v \in V_0 \cap V_S} \beta_v^S \delta_v^S + b_v^S d_v^S \right) \\
\text{s.t.} \quad & y_u + y_v \geq 1 \quad \forall uv \in E_0 \\
& y \in \mathbb{R}_{\geq 0}^{V_0} \\
& y_u^S + y_v^S \geq 1 \quad \forall uv \in E_S, \forall S \in \mathcal{S} \\
& y^S \in \mathbb{R}_{\geq 0}^{V_S} \quad \forall S \in \mathcal{S} \\
& y_v - y_v^S \leq \delta_v^S \quad \forall v \in V_0 \cap V_S, \forall S \in \mathcal{S} \\
& y_v^S - y_v \leq d_v^S \quad \forall v \in V_0 \cap V_S, \forall S \in \mathcal{S} \\
& \delta^S \in \mathbb{R}_{\geq 0}^{V_0 \cap V_S} \quad \forall S \in \mathcal{S} \\
& d^S \in \mathbb{R}_{\geq 0}^{V_0 \cap V_S} \quad \forall S \in \mathcal{S}
\end{aligned}
\tag{7.3}
$$

Lemma 7.2. *We can obtain an optimal solution for (7.3) from an optimal solution for (7.2).*

Proof.

Claim 7.1. *Any feasible solution for (7.3) can be mapped to a feasible solution for (7.2) with the same objective value.*

Proof. Let $\hat{\mu} = (\hat{y}, \hat{y}^S \text{ for } S \in S, \hat{\delta}^S \text{ for } S \in S, \hat{d}^S \text{ for } S \in S)$ be a feasible solution for (7.3). We extend it to a solution $\hat{\mu}_{\text{ext}}$ for (7.2) by setting $\hat{\gamma}_v = 1-\hat{y}_v$ for all $v \in V_0 \cap V_1$, $\hat{\gamma}_v = \hat{y}_v$ for all $v \in V_0 \cap V_2$, $\hat{\gamma}_v^S = 1 - \hat{y}_v^S$ for all $v \in V_S \cap V_1$, $S \in \mathcal{S}$, and $\hat{\gamma}_v^S = \hat{y}_v^S$ for all $v \in V_S \cap V_2$, $S \in \mathcal{S}$. It is clear that $\hat{\mu}_{\text{ext}}$ has the same objective value as $\hat{\mu}$, as the objective functions of the two linear programs are the same and do not involve the γ variables. We next show that $\hat{\mu}_{\text{ext}}$ is feasible.

The constraints $\gamma_v + y_v \geq 1$, $\gamma_v^S + y_v^S \geq 1$, $-\gamma_v + y_v \geq 0$ and $-\gamma_v^S + y_v^S \geq 0$ of (7.2) are satisfied by $\hat{\mu}_{\text{ext}}$ by definition. Nonnegativity of y, y^S, δ^S and d^S for all $S \in \mathcal{S}$ are satisfied by $\hat{\mu}_{\text{ext}}$, because $\hat{\mu}$ is feasible for (7.3).

Let $S \in \mathcal{S}$ and $v \in (V_0 \cap V_S) \cap V_1$. We have

$$\hat{\gamma}_v - \hat{\gamma}_v^S + \hat{\delta}_v^S = 1 - \hat{y}_v - (1 - \hat{y}_v^S) + \hat{\delta}_v^S = \hat{y}_v^S - \hat{y}_v + \hat{\delta}_v^S \geq 0,$$

and

$$\hat{\gamma}_v^S - \hat{\gamma}_v + \hat{d}_v^S = 1 - \hat{y}_v^S - (1 - \hat{y}_v) + \hat{d}_v^S = \hat{y}_v - \hat{y}_v^S + \hat{d}_v^S \geq 0,$$

where both times the last inequality follows from $\hat{\mu}$'s feasibility for (7.3). Similarly, we have for $S \in \mathcal{S}$ and $v \in (V_0 \cap V_S) \cap V_2$

$$\hat{\gamma}_v^S - \hat{\gamma}_v + \hat{\delta}_v^S = \hat{y}_v^S - \hat{y}_v + \hat{\delta}_v^S \geq 0,$$

and

$$\hat{\gamma}_v - \hat{\gamma}_v^S + \hat{d}_v^S = \hat{y}_v - \hat{y}_v^S + \hat{d}_v^S \geq 0.$$

Let $uv \in E_0$ such that $u \in V_1$ and $v \in V_2$. We have

$$\hat{\gamma}_v - \hat{\gamma}_u = \hat{y}_v - (1 - \hat{y}_u) = \hat{y}_v + \hat{y}_u - 1 \geq 0.$$

Finally, let $S \in \mathcal{S}$ and $uv \in E_S$ such that $u \in V_1$ and $v \in V_2$. We have

$$\hat{\gamma}_v^S - \hat{\gamma}_u^S = \hat{y}_v^S - (1 - \hat{y}_u^S) = \hat{y}_v^S + \hat{y}_u^S - 1 \geq 0.$$

□

Claim 7.2. *Any feasible solution to (7.2) can be mapped to a feasible solution to (7.3) with the same objective value.*

Proof. Let $\hat{\mu} = (\hat{y}, \hat{y}^S \text{ for } S \in S, \hat{\delta}^S \text{ for } S \in S, \hat{d}^S \text{ for } S \in S, \hat{\gamma}, \hat{\gamma}^S \text{ for } S \in S)$ be a feasible solution for (7.2). We restrict it to a solution $\hat{\mu}_{\text{res}}$ for (7.3) by disregarding the γ variables. It is clear that $\hat{\mu}_{\text{res}}$ has the same objective value as $\hat{\mu}$, as the objective functions of the two linear programs are the same and do not involve the γ variables. We next show that $\hat{\mu}_{\text{res}}$ is feasible.

First observe that $\hat{\mu}_{\text{res}} \geq 0$.

Let $uv \in E_0$ such that $u \in V_1$ and $v \in V_2$. We have

$$\hat{y}_u + \hat{y}_v = (\hat{\gamma}_u + \hat{y}_u) + (\hat{\gamma}_v - \hat{\gamma}_u) + (-\hat{\gamma}_v + \hat{y}_v) \geq 1 + 0 + 0 = 1.$$

Similarly, let $S \in \mathcal{S}$ and $uv \in E_S$ such that $u \in V_1$ and $v \in V_2$. We have

$$\hat{y}^S_u + \hat{y}^S_v = (\hat{\gamma}^S_u + \hat{y}^S_u) + (\hat{\gamma}^S_v - \hat{\gamma}^S_u) + (-\hat{\gamma}^S_v + \hat{y}^S_v) \geq 1 + 0 + 0 = 1.$$

Let $S \in \mathcal{S}$, $v \in (V_0 \cap V_S) \cap V_1$. We will show that without loss of generality we can assume that $\hat{\gamma}_v + \hat{y}_v = 1$ and $\hat{\gamma}^S_v + \hat{y}^S_v = 1$. Consequently, we have

$$\begin{aligned}
\hat{\delta}^S_v + \hat{y}^S_v - \hat{y}_v &= \hat{\delta}^S_v + \hat{y}^S_v - \hat{y}_v + \hat{\gamma}^S_v - \hat{\gamma}^S_v + \hat{\gamma}_v - \hat{\gamma}_v \\
&= (\hat{\delta}^S_v + \hat{\gamma}_v - \hat{\gamma}^S_v) + (\hat{\gamma}^S_v + \hat{y}^S_v) - (\hat{\gamma}_v + \hat{y}_v) \\
&\geq 0 + 1 - 1 = 0,
\end{aligned}$$

and

$$\begin{aligned}
\hat{d}^S_v + \hat{y}_v - \hat{y}^S_v &= \hat{d}^S_v + \hat{y}_v - \hat{y}^S_v + \hat{\gamma}_v - \hat{\gamma}_v + \hat{\gamma}^S_v - \hat{\gamma}^S_v \\
&= (\hat{d}^S_v + \hat{\gamma}^S_v - \hat{\gamma}_v) + (\hat{\gamma}_v + \hat{y}_v) - (\hat{\gamma}^S_v + \hat{y}^S_v) \\
&\geq 0 + 1 - 1 = 0.
\end{aligned}$$

Now, let $S \in \mathcal{S}$, $v \in (V_0 \cap V_S) \cap V_2$. We will also show that without loss of generality we can assume that $-\hat{\gamma}_v + \hat{y}_v = 0$ and $-\hat{\gamma}^S_v + \hat{y}^S_v = 0$. Consequently, we have

$$\begin{aligned}
\hat{\delta}^S_v + \hat{y}^S_v - \hat{y}_v &= \hat{\delta}^S_v + \hat{y}^S_v - \hat{y}_v + \hat{\gamma}^S_v - \hat{\gamma}^S_v + \hat{\gamma}_v - \hat{\gamma}_v \\
&= (\hat{\delta}^S_v + \hat{\gamma}^S_v - \hat{\gamma}_v) + (-\hat{\gamma}^S_v + \hat{y}^S_v) - (-\hat{\gamma}_v + \hat{y}_v) \\
&\geq 0 + 0 - 0 = 0,
\end{aligned}$$

and

$$\begin{aligned}
\hat{d}^S_v + \hat{y}_v - \hat{y}^S_v &= \hat{d}^S_v + \hat{y}_v - \hat{y}^S_v + \hat{\gamma}_v - \hat{\gamma}_v + \hat{\gamma}^S_v - \hat{\gamma}^S_v \\
&= (\hat{d}^S_v + \hat{\gamma}_v - \hat{\gamma}^S_v) + (-\hat{\gamma}_v + \hat{y}_v) - (-\hat{\gamma}^S_v + \hat{y}^S_v) \\
&\geq 0 + 0 - 0 = 0,
\end{aligned}$$

This finishes the feasibility proof of $\hat{\mu}_{\text{res}}$.

To show that we can indeed assume without loss of generality that $\hat{\gamma}_v + \hat{y}_v = 1$ for $v \in V_0 \cap V_1$, suppose it does not hold, so: $\hat{\gamma}_v + \hat{y}_v > 1$. If $\hat{y}_v > 0$, then lower $\hat{y}_v$ by $\min\{\hat{y}_v, 1 - \hat{\gamma}_v\}$ (this does not affect feasibility as $\hat{y}_v$ is contained in only this one constraint, and it improves the objective). If now $\hat{\gamma}_v + \hat{y}_v = 1$, then we are done. If not, then it must be that $\hat{y}_v = 0$ and $\hat{\gamma}_v > 1$. Let $0 < \eta \leq \hat{\gamma}_v - 1$ and lower $\hat{\gamma}_v$ by η: $\hat{\gamma}'_v = \hat{\gamma}_v - \eta$ (this does not change the objective). By definition of η, we still have $\hat{\gamma}_v' + \hat{y}_v \geq 1$. Let $u \in V_2$ such that $uv \in E_0$, then we have

$$\hat{\gamma}_u - \hat{\gamma}_v' = \hat{\gamma}_u - (\hat{\gamma}_v - \eta) = \hat{\gamma}_u - \hat{\gamma}_v + \eta \geq \eta > 0.$$

If $v \notin V_S$ for all $S \in \mathcal{S}$, then the solution with $\hat{\gamma}_v$ replaced by $\hat{\gamma}_v'$ is feasible. If there is at least one $S \in \mathcal{S}$ such that $v \in V_S$, then we have

$$\hat{\gamma}_v^S - \hat{\gamma}_v' + \hat{d}_v^S = \hat{\gamma}_v^S - (\hat{\gamma}_v - \eta) + \hat{d}_v^S \geq \eta > 0,$$

for all $S \in \mathcal{S}$ such that $v \in V_S$. If also $\hat{\gamma}_v' - \hat{\gamma}_v^S + \hat{\delta}_v^S \geq 0$ for some choice of η, then the solution with $\hat{\gamma}_v$ replaced by $\hat{\gamma}_v'$ is again feasible.

So now suppose that there is some $S \in \mathcal{S}$ with $v \in V_S$, such that for all choices of η, this latter constraint is not satisfied. Then it must be that $\hat{\gamma}_v - \hat{\gamma}_v^S + \hat{\delta}_v^S = 0$. We can make this constraint work if we also decrease $\hat{\gamma}_v^S$ by η: $(\hat{\gamma}_v^S)' = \hat{\gamma}_v^S - \eta$ (this does not change the objective). Like before, it is clear that the constraint $\gamma_u^S - \gamma_v^S \geq 0$ is still satisfied, as we increase the left hand side. The constraint $\gamma_v^S - \gamma_v + d_v^S \geq 0$ is also still satisfied, as we decrease $\hat{\gamma}_v^S$ and $\hat{\gamma}_v$ by the same amount. Finally, using $\hat{\gamma}_v - \hat{\gamma}_v^S + \hat{\delta}_v^S = 0$ and $\hat{\gamma}_v - \eta \geq 1$, we have

$$(\hat{\gamma}_v^S)' + \hat{y}_v^S = \hat{\gamma}_v^S - \eta + \hat{y}_v^S = \hat{\gamma}_v + \hat{\delta}_v^S - \eta + \hat{y}_v^S \geq 1 + 0 + 0 = 1.$$

So again, we find that the solution with $\hat{\gamma}_v$ replaced by $\hat{\gamma}_v'$ is feasible. Consequently, we can decrease the value of $\hat{\gamma}_v$. By possibly repeating this argument, we can decrease $\hat{\gamma}_v$ to 1, so that $\hat{\gamma}_v + \hat{y}_v = 1$.

By similar arguments we can show that the other "without loss of generality"-assumptions hold as well. □

Now we can map an optimal solution for (7.2) to a solution for (7.3) with the same objective value, as described above. This solution is optimal for (7.3), as otherwise we could find a better solution, map the better solution back to a solution for (7.2) with the same objective value, contradicting the optimality of the starting solution. □

Lemma 7.3. *An integral optimal solution for* (7.3) *is also an integral optimal solution for* (2SAG-LP).

Proof. Let $\hat{\gamma} = (\hat{y}, \hat{y}^S \text{ for } S \in S, \hat{\delta}^S \text{ for } S \in S, \hat{d}^S \text{ for } S \in S)$ be an integral optimal solution for (7.3).

Suppose that $1^\top \hat{y} > \nu(G_0)$, then because $\hat{y}$ is integral, $1^\top \hat{y} \geq \nu(G_0) + 1$. Now replace $\hat{y}$ by a minimum vertex cover $\widetilde{y}$, that is, $1^\top \widetilde{y} = \nu(G_0)$, and set $\widetilde{\delta}^S$ and

$\widetilde{d}^S$ accordingly for all $S \in \mathcal{S}$. We have $\widetilde{y} \leq 1$ and $\hat{y}^S \leq 1$, and hence also $\widetilde{\delta}^S \leq 1$ and $\widetilde{d}^S \leq 1$. Therefore,

$$\begin{aligned}
&\varepsilon\left(\sum_{v\in V_0}\alpha_v\widetilde{y}_v + \sum_{S\in\mathcal{S}}\sum_{v\in V_S}\alpha_v^S\hat{y}_v^S + \sum_{S\in\mathcal{S}}\sum_{v\in V_0\cap V_S}\beta_v^S\widetilde{\delta}_v^S + b_v^S\widetilde{d}_v^S\right)\\
&\qquad\qquad \leq \varepsilon\left(\sum_{v\in V_0}\alpha_v + \sum_{S\in\mathcal{S}}\sum_{v\in V_S}\alpha_v^S + \sum_{S\in\mathcal{S}}\sum_{v\in V_0\cap V_S}\beta_v^S + b_v^S\right)\\
&\qquad\qquad \leq \varepsilon\left(\sum_{v\in V_0}|\alpha_v| + \sum_{S\in\mathcal{S}}\sum_{v\in V_S}|\alpha_v^S| + \sum_{S\in\mathcal{S}}\sum_{v\in V_0\cap V_S}\beta_v^S + b_v^S\right)\\
&\qquad\qquad < \varepsilon\left(1 + \sum_{v\in V_0}|\alpha_v| + \sum_{S\in\mathcal{S}}\sum_{v\in V_S}|\alpha_v^S| + \sum_{S\in\mathcal{S}}\sum_{v\in V_0\cap V_S}\beta_v^S + b_v^S\right)\\
&\qquad\qquad = 1,
\end{aligned}$$

which means we obtain a strictly better solution, contradicting that $\hat{\gamma}$ is optimal. So, $1^\top\hat{y} = \nu(G_0)$. Similarly, $1^\top\hat{y}^S = \nu(G_S)$ for all $S \in \mathcal{S}$. Hence, $\hat{\gamma}$ is feasible for (2SAG-LP).

Suppose $\hat{\gamma}$ is not optimal for (2SAG-LP). Let $\widetilde{\gamma}$ be an optimal solution for (2SAG-LP). Then

$$\begin{aligned}
&\sum_{v\in V_0}\alpha_v\widetilde{y}_v + \sum_{S\in\mathcal{S}}\sum_{v\in V_S}\alpha_v^S\widetilde{y}_v^S + \sum_{S\in\mathcal{S}}\sum_{v\in V_0\cap V_S}\beta_v^S\widetilde{\delta}_v^S + b_v^S\widetilde{d}_v^S\\
&\qquad\qquad < \sum_{v\in V_0}\alpha_v\hat{y}_v + \sum_{S\in\mathcal{S}}\sum_{v\in V_S}\alpha_v^S\hat{y}_v^S + \sum_{S\in\mathcal{S}}\sum_{v\in V_0\cap v_S}\beta_v^S\hat{\delta}_v^S + b_v^S\hat{d}_v^S,
\end{aligned}$$

$1^\top\widetilde{y} = \nu(G_0) = 1^\top\hat{y}$, and $1^\top\widetilde{y}^S = \nu(G_S) = 1^\top\hat{y}^S$ for all $S \in \mathcal{S}$. It follows that

$$\begin{aligned}
&1^\top\widetilde{y} + \sum_{S\in\mathcal{S}}1^\top\widetilde{y}^S\\
&\qquad + \varepsilon\left(\sum_{v\in V_0}\alpha_v\widetilde{y}_v + \sum_{S\in\mathcal{S}}\sum_{v\in V_S}\alpha_v^S\widetilde{y}_v^S + \sum_{S\in\mathcal{S}}\sum_{v\in V_0\cap V_S}\beta_v^S\widetilde{\delta}_v^S + b_v^S\widetilde{d}_v^S\right)\\
&< 1^\top\hat{y} + \sum_{S\in\mathcal{S}}1^\top\hat{y}^S\\
&\qquad + \varepsilon\left(\sum_{v\in V_0}\alpha_v\hat{y}_v + \sum_{S\in\mathcal{S}}\sum_{v\in V_S}\alpha_v^S\hat{y}_v^S + \sum_{S\in\mathcal{S}}\sum_{v\in V_0\cap V_S}\beta_v^S\hat{\delta}_v^S + b_v^S\hat{d}_v^S\right),
\end{aligned}$$

because $\varepsilon > 0$. This contradicts the optimality of $\hat{\gamma}$ for (7.3), hence $\hat{\gamma}$ must be optimal for (2SAG-LP). □

Finally, Lemmas 7.1 to 7.3 prove Theorem 7.1.

7.2 Implicit Distribution

We here prove that, when the distribution is not known, the problem becomes hard but it can still be well approximated using the SAA method. For the SAA analysis, the integrality result proved in the previous section plays a central role. In terms of techniques, the results in this section follow closely the ones in Faenza et al. [20], we still include all the details for the sake of completeness.

7.2.1 Hardness

Theorem 7.2. *When the second-stage distribution is specified implicitly by a sampling oracle, there exists no algorithm that solves (2SAG) in time polynomial in the input size and the number of calls to the oracle, unless P = NP. This holds even if λ is nonzero for only one vertex $v \in V_0$, and if all second-stage scenarios are obtained by only removing vertices.*

Proof. As in Faenza et al. [20], we prove this hardness result by showing that if such an algorithm were to exist, then it could be used to count the number of vertex covers in a graph in polynomial time. However, counting the number of vertex covers in a graph is #P-hard (Provan and Ball [43]).

Let $G = (V, E)$ be any undirected graph. We create an instance of (2SAG) as follows.

- *First-stage instance:* The first-stage graph $G_0 = (V_0, E_0)$ is given by

  $$V_0 = \{v_1, \ldots, v_{d_v} : v \in V\} \cup E \cup \{\alpha, \beta_1, \beta_2\},$$

  and

  $$E_0 = \{ev_1, \ldots, ev_{d_v} : v \in e \in E\} \cup \{\alpha e : e \in E\} \cup \{\alpha\beta_1, \alpha\beta_2\}.$$

  This is a bipartite graph with bipartitions $\{v_1, \ldots, v_{d_v} : v \in V\} \cup \{\alpha\}$ and $E \cup \{\beta_1, \beta_2\}$.

- *Second-stage instance:* Sampling from the second-stage distribution $\mathcal{D}$ consists of the following: Add the players $\{v_1, \ldots, v_{d_v}\}$ with probability $\frac{1}{2}$, independently for all $v \in V$. Add the players $E \cup \alpha$. The second-stage graph is the subgraph of G_0 induced by these vertices. Observe that this graph is bipartite, and in particular has the same bipartition as G_0.

- *Dissatisfaction costs:* Set $\lambda = 0$ except for $\lambda_\alpha = 1$.

In the first stage, both β_1 and β_2 only have an edge to α, which means that at least one of them is exposed. We have $\nu(G_0 \setminus \beta_i) = \nu(G_0)$ for $i = 1, 2$. It follows that in any core element, they have value zero. To cover the edges between β_1, β_2 and α, it follows that in any core element, α must have value one. Also observe that in the second stage, α has core value at most one. The objective (2SAG) in this case becomes

$$\mathbb{E}_{S\sim\mathcal{D}}\left[\min_{y^S\in\text{core}(G_S)} 1 - y_\alpha^S\right].$$

A second-stage vertex set will look like

$$\{v_1, \dots, v_{d_v} : v \in S\} \cup E \cup \alpha,$$

for some $S \subseteq V$. Denote this set by $\Pi(S)$. For a given $S \subseteq V$, the probability that $\Pi(S)$ is the second-stage vertex set is $\frac{1}{2^{|V|}}$. With this information, we can write down the expectation explicitly:

$$\sum_{S\subseteq V} \frac{1}{2^{|V|}} \left[\min_{y^S\in\text{core}(G_0[\Pi(S)])} 1 - y_\alpha^S\right].$$

Suppose $S \subseteq V$ is a vertex cover of G. Since S is a vertex cover, for each edge $e = uv \in E$ at least one of u and v is in S. Without loss of generality, let us assume that $v \in S$. Then in $G_0[\Pi(S)]$, the vertex e can be matched to any copy of v; since we added d_v copies of v, there are definitely enough copies to cover all edge-vertices. This matching is perfect on one side of the bipartition, which means that it is maximum. Since α is not matched in this matching, we must have $y_\alpha^S = 0$.

Suppose $S \subseteq V$ is not a vertex cover of G. Since S is not a vertex cover, there exists an edge $e = uv \in E$ such that neither u nor v is in S. Consequently, in $G_0[\Pi(S)]$, the vertex e only has an edge to α. To cover this edge with the core element, we must have $y_e^S + y_\alpha^S = 1$. Since e is not incident with any other edges, it is feasible to set $y_e^S = 0$ and $y_\alpha^S = 1$. This is also the solution that minimizes $1 - y_\alpha^S$.

From these arguments it follows that

$$\begin{aligned}
&\sum_{S\subseteq V} \frac{1}{2^{|V|}} \left[\min_{y^S\in\text{core}(G_0[S])} 1 - y_\alpha^S\right] && \text{(7.4a)}\\
&\qquad = \frac{1}{2^{|V|}}(\#\text{vertex covers}(1-0) + \#\text{not vertex covers}(1-1)), && \text{(7.4b)}\\
&\qquad = \frac{1}{2^{|V|}}\#\text{vertex covers}. && \text{(7.4c)}
\end{aligned}$$

So, if we could solve (2SAG) in polynomial time, that is, determine its optimal objective value (= (7.4c)), then we could also determine the number of vertex covers in any graph in polynomial time. □

7.2.2 SAA Algorithm

The *sample average approximation (SAA)* method is a well-known method in stochastic programming. It has been exploited often to approximate two-stage stochastic combinatorial problems (see for example [14, 20, 44, 51, 52]). Let $S^1, \ldots, S^N$ be N independent and identically distributed samples drawn from the distribution $\mathcal{D}$. We replace the objective function in (2SAG) by the average taken over our samples $S^1, \ldots, S^N$:

$$\min_{y \in \mathrm{core}(G_0)} \frac{1}{N} \sum_{i=1}^{N} \left[\min_{y^i \in \mathrm{core}(G_i)} \sum_{v \in V_0 \cap V_i} \lambda_v \left[y_v - y_v^i \right]^+ \right], \tag{SAA}$$

where we use $G_i = (V_i, E_i)$ to denote $G_{S^i} = (V_{S^i}, E_{S^i})$ for $i = 1, \ldots, N$. Observe that (SAA) is an instance of (2SAG-expl), where $\mathcal{S} = \left\{ S^1, \ldots, S^N \right\}$ and $p_S = \frac{1}{N}$ for all $S \in \mathcal{S}$, which means we can solve this problem by solving (2SAG-expl), and in particular we can obtain an integral solution.

For any instance $\mathcal{I}$ of (2SAG), we denote by $y^{\mathcal{I}}$ the optimal solution for instance $\mathcal{I}$, and by $\mathrm{val}_{\mathcal{I}}(y)$ the objective value of y in instance $\mathcal{I}$.

Theorem 7.3. *Given an instance $\mathcal{I}$ of (2SAG) where a sampling oracle specifies the second-stage distribution implicitly, and two parameters $\epsilon > 0$, $\alpha \in (0, 1)$, one can compute a first-stage core element y such that*

$$\mathbb{P}(val_{\mathcal{I}}(y) \leq val_{\mathcal{I}}(y^{\mathcal{I}}) + \varepsilon) \geq 1 - \alpha,$$

in time polynomial in the size of $\mathcal{I}$, λ, $\ln(1/\alpha)$ and $1/\varepsilon$.

Let $\hat{y}$ be an extreme point optimal solution for (SAA), which is integral by Theorem 7.1. As in Faenza et al. [20], the main ingredient to prove Theorem 7.3 is the next lemma.

Lemma 7.4. *For any $\alpha \in (0, 1)$, the following holds with probability at least $1 - \alpha$*

$$val_{\mathcal{I}}(\hat{y}) \leq val_{\mathcal{I}}(y^{\mathcal{I}}) + \sqrt{2} \sum_{v \in V_0} \lambda_v \sqrt{\frac{\ln(2^{|V_0|}/\alpha)}{N}}.$$

The proof of this lemma follows the proof of the corresponding lemma in Faenza et al. [20] closely. The key difference is that where Faenza et al. [20]

use that the number of stable matchings is bounded, we use the integrality result of previous section to bound the amount of core solutions we have to consider. We denote by $\mathrm{val}_{\mathrm{SAA}}(y)$ the objective value of y in an instance of (SAA).

Proof of Lemma 7.4. For $i \in \{1, \ldots, N\}$ and $y \in \mathrm{core}(G_0)$, let

$$G_i(y) = \min_{y^i \in \mathrm{core}(G_i)} \sum_{v \in V_0 \cap V_i} \lambda_v \left[y_v - y_v^i \right]^+ .$$

Let

$$\varepsilon = \sqrt{2} \sum_{v \in V_0} \lambda_v \sqrt{\frac{\ln(2^{|V_0|}/\alpha)}{N}},$$

and let $\mathcal{F}^\varepsilon = \left\{ y \in \mathrm{core}(G_0) : \mathrm{val}_{\mathcal{I}}(y) \leq \mathrm{val}_{\mathcal{I}}(y^{\mathcal{I}}) + \varepsilon \right\}$. We obtain

$$\begin{aligned}
\mathbb{P}(\hat{y} \notin \mathcal{F}^\varepsilon) &\leq \sum_{y \notin \mathcal{F}^\varepsilon, y \in \{0,1\}^{|V_0|}} \mathbb{P}(y \text{ is an optimal solution for (SAA)}) \\
&\leq \sum_{y \notin \mathcal{F}^\varepsilon, y \in \{0,1\}^{|V_0|}} \mathbb{P}\left(\mathrm{val}_{\mathrm{SAA}}(y) \leq \mathrm{val}_{\mathrm{SAA}}(y^{\mathcal{I}}) \right) \\
&= \sum_{y \notin \mathcal{F}^\varepsilon, y \in \{0,1\}^{|V_0|}} \mathbb{P}\left(\frac{1}{N} \sum_{i=1}^N G_i(y) \leq \frac{1}{N} \sum_{i=1}^N G_i(y^{\mathcal{I}}) \right) \\
&= \sum_{y \notin \mathcal{F}^\varepsilon, y \in \{0,1\}^{|V_0|}} \mathbb{P}\left(\frac{1}{N} \sum_{i=1}^N \left(G_i(y) - G_i(y^{\mathcal{I}}) \right) \leq 0 \right).
\end{aligned}$$

Observe that we can restrict ourselves to sum over integral y not in $\mathcal{F}^\varepsilon$, as $\hat{y}$ is integral. Fix $y \notin \mathcal{F}^\varepsilon$, $y \in \{0,1\}^{|V_0|}$ and let $X_i = G_i(y) - G_i(y^{\mathcal{I}})$. We bound the probability $\mathbb{P}(\frac{1}{N} \sum_{i=1}^N X_i \leq 0)$ using the classical Hoeffding's inequality, stated below.

Lemma 7.5 (Hoeffding [29], Faenza et al. [20])**.** *Let $X_1, \ldots, X_n$ be independent random variables such that $a_i \leq X_i \leq b_i$ almost surely. Consider the sum of these random variables $S_n = X_1 + \cdots + X_n$. The Hoeffding's inequality states that for all $t > 0$,*

$$\mathbb{P}(\mathbb{E}(S_n) - S_n \geq t) \leq \exp\left(\frac{-2t^2}{\sum_{i=1}^n (b_i - a_i)^2} \right).$$

It follows from $0 \leq y \leq 1$ that $0 \leq [y_v - y^i]^+ \leq 1$, and consequently that

$$0 \leq G_i(y) \leq \sum_{v \in V_0 \cap V_i} \lambda_v \leq \sum_{v \in V_0} \lambda_v.$$

Let $a_i = -\sum_{v\in V_0}\lambda_v$ and $b_i = \sum_{v\in V_0}\lambda_v$. Then $a_i \leq X_i \leq b_i$. Now since $y \notin \mathcal{F}^\varepsilon$, it holds that $\mathbb{E}(X_i) \geq \varepsilon$. Let $S_N = \sum_{i=1}^N X_i$. We obtain

$$\begin{aligned}
\mathbb{P}\left(\frac{1}{N}\sum_{i=1}^N X_i \leq 0\right) &= \mathbb{P}\left(\frac{1}{N}S_N \leq 0\right) = \mathbb{P}\left(S_N \leq 0\right)\\
&= \mathbb{P}\left(\mathbb{E}(S_N) - S_N \geq \mathbb{E}(S_N)\right)\\
&\leq \mathbb{P}\left(\mathbb{E}(S_N) - S_N \geq \varepsilon N\right)\\
&\leq \exp\left(\frac{-2\varepsilon^2 N^2}{\sum_{i=1}^N (b_i - a_i)^2}\right)\\
&= \exp\left(\frac{-2^2\left(\sum_{v\in V_0}\lambda_v\right)^2 \ln(2^{|V_0|}/\alpha)N}{N(2\sum_{v\in V_0}\lambda_v)^2}\right)\\
&= \exp\left(-\ln(2^{|V_0|}/\alpha)\right) = \frac{\alpha}{2^{|V_0|}}.
\end{aligned}$$

So finally,

$$\mathbb{P}(\hat{y} \notin \mathcal{F}^\varepsilon) \leq \sum_{y\notin\mathcal{F}^\varepsilon, y\in\{0,1\}^{|V_0|}} \mathbb{P}\left(\frac{1}{N}\sum_{i=1}^N X_i \leq 0\right) \leq \sum_{y\notin\mathcal{F}^\varepsilon, y\in\{0,1\}^{|V_0|}} \frac{\alpha}{2^{|V_0|}} \leq \alpha,$$

where the last inequality follows from the fact that since $y \in \{0,1\}^{|V_0|}$, there are at most $2^{|V_0|}$ terms in the sum. $\square$

Finally, for any $\varepsilon > 0$ and $\alpha \in (0,1)$, setting $N = 2\left(\sum_{v\in V_0}\lambda_v\right)^2 \ln(2^{|V_0|}/\alpha)/\varepsilon^2$ proves Theorem 7.3.

7.3 Multistage Setting

In this section we consider a multistage setting where there are k stages, with predetermined graphs (no distribution). We denote by $G_i = (V_i, E_i)$ the graph of the i'th stage for $i = 1,\ldots,k$. This setting resembles the setting in Fluschnik et al. [23], who discuss multistage vertex cover. As before, and as in Fluschnik et al. [23], we consider the problem of minimizing the absolute difference. Again, one can still choose later to minimize the positive difference by choosing which variables to take into the objective. We can formulate this problem as follows.

$$\min_{y^i\in\text{core}(G_i), i=1,\ldots,k} \sum_{i=1}^{k-1}\sum_{v\in V_i\cap V_{i+1}} \lambda_v^i \left|y_v^i - y_v^{i+1}\right|. \tag{7.5}$$

Like before, we can formulate this as the following LP.

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{k-1} \sum_{v \in V_i \cap V_{i+1}} \lambda_v^i \left(\delta_v^i + d_v^i\right) \\
\text{s.t.} \quad & y_u^i + y_v^i \geq 1 \quad \forall uv \in E_i, \forall i = 1, \dots, k \\
& 1^\top y^i = \nu(G_i) \quad \forall i = 1, \dots, k \\
& y^i \in \mathbb{R}_{\geq 0}^{V_i} \quad \forall i = 1, \dots, k \\
& y_v^i - y_v^{i+1} \leq \delta_v^i \quad \forall v \in V_i \cap V_{i+1}, \forall i = 1, \dots, k-1 \\
& y_v^{i+1} - y_v^i \leq d_v^i \quad \forall v \in V_i \cap V_{i+1}, \forall i = 1, \dots, k-1 \\
& \delta^i \in \mathbb{R}_{\geq 0}^{V_i \cap V_{i+1}} \quad \forall i = 1, \dots, k-1 \\
& d^i \in \mathbb{R}_{\geq 0}^{V_i \cap V_{i+1}} \quad \forall i = 1, \dots, k-1
\end{aligned}
\tag{7.6}
$$

Let $V = \cup_{i=1}^k V_i$. This LP has $O(|V|k)$ variables and $O(|V|^2 k)$ constraints. So it has size polynomial in the input size, which means we can solve (7.5) in polynomial time, by solving (7.6). Like in Section 7.1, we can show that the feasible region of (7.6) is an integral polyhedron.

Theorem 7.4. *The feasible region of* (7.6) *is an integral polyhedron.*

Before we go into the proof, we discuss a consequence of this theorem. In particular, we can use this theorem to prove a result about the multistage vertex cover problem, which we now define formally. A vertex cover C in a graph $G = (V, E)$ is a subset of vertices $C \subseteq V$ such for each edge $uv \in E$, we have $\{u, v\} \cap C \neq \emptyset$. In the multistage vertex cover problem there are k stages with predetermined graphs $G_1, \dots, G_k$. The goal is to find a vertex cover for each stage $C_1, \dots, C_k$ such that the total absolute difference between stages is minimized, that is,

$$\sum_{i=1}^{k-1} |C_i \setminus C_{i+1}| + |C_{i+1} \setminus C_i|.$$

Fluschnik et al. [23] have shown that the multistage vertex cover problem is NP-hard, already when $k = 2$, the first-stage graph is a path and the second-stage graph is a tree. Both graphs are indeed very simple bipartite graphs. Our result shows that the difficulty lies in the fact that the bipartitions for the two stages are different. In fact, with the additional requirement that the bipartitions are the same, we prove that the problem becomes polynomial-time solvable for any number of stages.

Theorem 7.5. *The multistage vertex cover problem is solvable in polynomial time when all G_i, $i = 1, \dots, k$, are bipartite with the same bipartition.*

Proof. We can solve the multistage vertex cover problem on G_i, $i = 1, \dots, k$ by modeling it as a multistage assignment game with the LP in (7.6). There, we set $\lambda_v^i = 1$ for all $i = 1, \dots, k-1$ and all $v \in V_i \cap V_{i+1}$. By Theorem 7.4 we can obtain an integral optimal solution, which means that the vectors y^i in this solution indicate vertex covers C_i. □

Now we go into the proof of Theorem 7.4, following the same line of argument as in the proof of Theorem 7.1.

We consider (7.6) with an arbitrary objective, and show that for each objective we can find an integral optimal solution. Since for each extreme point of a polyhedron there is an objective such that the extreme point is the unique optimal solution, this proves that the polyhedron is integral. We consider the following objective, where α can take any value, $\beta \geq 0$ and $b \geq 0$. Note that if $\beta_v^i < 0$ or $b_v^i < 0$ for any $v \in V_i \cap V_{i+1}$, $i = 1, \dots, k-1$, then the LP becomes unbounded, and so there are no extreme points in those directions.

$$\min \quad \sum_{i=1}^{k} \sum_{v \in V_i} \alpha_v^i y_v + \sum_{i=1}^{k-1} \sum_{v \in V_i \cap V_{i+1}} \beta_v^i \delta_v^i + b_v^i d_v^i$$

We will formulate a maximum flow problem on an auxiliary graph, and compute its dual LP. It is known that the constraint matrix of a maximum flow LP is totally unimodular (TU). The dual LP has the transpose as constraint matrix, which is then also TU. Since in addition the right hand side of the dual constraints are integral, this means the feasible region of the dual is an integral polyhedron. (See for example Schrijver [47] for properties of TU matrices.) Finally, we show that we can map an optimal dual solution to an optimal solution for (7.6).

Let $V_s, V_t \subseteq \cup_{i=1}^k V_i$ be the bipartition of $\cup_{i=1}^k G_i$. Let

$$\varepsilon = \frac{1}{1 + \sum_{v \in V_i} |\alpha_v^i| + \sum_{i=1}^{k-1} \sum_{v \in V_i \cap V_{i+1}} \beta_v^i + b_v^i}.$$

Note that $\varepsilon > 0$. We create the auxiliary graph $G' = (V', A)$ as follows. Let $V' = \cup_{i=1}^k V_i \cup \{s, t\}$. To make clear about which version of a vertex $v \in V_i \cap V_{i+1}$ we are talking, we (sometimes) denote them by v^i and v^{i+1}. The arc set A contains the following arcs:

- sv for all $v \in V_i \cap V_s$ and $i = 1, \dots, k$, with flow-capacity $1 + \varepsilon\alpha_v^i$;
- vt for all $v \in V_i \cap V_t$ and $i = 1, \dots, k$, with flow-capacity $1 + \varepsilon\alpha_v^i$;
- $v^{i+1}v^i$ for all $v \in (V_i \cap V_{i+1}) \cap V_s$, $i = 1, \dots, k-1$, with flow-capacity $\varepsilon\beta_v^i$;

- $v^i v^{i+1}$ for all $v \in (V_i \cap V_{i+1}) \cap V_s$, $i = 1, \dots, k-1$, with flow-capacity εb_v^i;
- $v^i v^{i+1}$ for all $v \in (V_i \cap V_{i+1}) \cap V_t$, $i = 1, \dots, k-1$, with flow-capacity $\varepsilon \beta_v^i$;
- $v^{i+1} v^i$ for all $v \in (V_i \cap V_{i+1}) \cap V_t$, $i = 1, \dots, k-1$, with flow-capacity εb_v^i;
- uv for all $uv \in \bigcup_{i=1}^k E_i$, such that $u \in V_s$, $v \in V_t$, without upperbound on the flow-capacity.

We formulate the maximum flow problem in G' as an LP as follows.

$$\begin{aligned}
\max \quad & \sum_{v \in V_s} f_{sv} \\
\text{s.t.} \quad & f_{sv} + \mathbb{1}[v \in V_{i-1}](f_{v^{i-1}v} - f_{vv^{i-1}}) + \mathbb{1}[v \in V_{i+1}](f_{v^{i+1}v} - f_{vv^{i+1}}) \\
& \qquad\qquad - \sum_{u: uv \in E_i} f_{vu} = 0 \quad \forall v \in V_i \cap V_s, \forall i = 1, \dots, k \\
& \sum_{u: uv \in E_i} f_{uv} + \mathbb{1}[v \in V_{i-1}](f_{v^{i-1}v} - f_{vv^{i-1}}) + \mathbb{1}[v \in V_{i+1}](f_{v^{i+1}v} \\
& \qquad\qquad - f_{vv^{i+1}}) - f_{vt} = 0 \quad \forall v \in V_i \cap V_t, \forall i = 1, \dots, k \\
& f_{sv} \leq 1 + \varepsilon \alpha_v^i \quad \forall v \in V_i \cap V_s, \forall i = 1, \dots, k \\
& f_{vt} \leq 1 + \varepsilon \alpha_v^i \quad \forall v \in V_i \cap V_t, \forall i = 1, \dots, k \\
& f_{v^{i+1}v^i} \leq \varepsilon \beta_v^i \quad \forall v \in (V_i \cap V_{i+1}) \cap V_s, \forall i = 1, \dots, k-1 \\
& f_{v^i v^{i+1}} \leq \varepsilon b_v^i \quad \forall v \in (V_i \cap V_{i+1}) \cap V_s, \forall i = 1, \dots, k-1 \\
& f_{v^i v^{i+1}} \leq \varepsilon \beta_v^i \quad \forall v \in (V_i \cap V_{i+1}) \cap V_t, \forall i = 1, \dots, k-1 \\
& f_{v^{i+1}v^i} \leq \varepsilon b_v^i \quad \forall v \in (V_i \cap V_{i+1}) \cap V_t, \forall i = 1, \dots, k-1 \\
& f \in \mathbb{R}_{\geq 0}^A
\end{aligned} \tag{7.7}$$

For this LP to have a feasible flow, the flow-capacities need to be nonnegative: $1 + \varepsilon \alpha_v^i \geq 0$ for all $v \in V_i$, $i = 1, \dots, k$, $\varepsilon \beta_v^i \geq 0$ for all $v \in V_i \cap V_{i+1}$, $i = 1, \dots, k-1$, and $\varepsilon b_v^i \geq 0$ for all $v \in V_i \cap V_{i+1}$, $i = 1, \dots, k-1$. The later two are satisfied as $\varepsilon > 0$ and we set $\beta, b \geq 0$. For any $v \in V_i$, $i = 1, \dots, k$, we have

$$\alpha_v^i \geq -|\alpha_v^i| \geq -\left(1 + \sum_{v \in V_i} |\alpha_v^i| + \sum_{i=1}^{k-1} \sum_{v \in V_i \cap V_{i+1}} \beta_v^i + b_v^i\right) = -\frac{1}{\varepsilon},$$

and so, since $\varepsilon > 0$, we have $1 + \varepsilon \alpha_v^i \geq 1 + \varepsilon \frac{-1}{\varepsilon} = 0$.

The dual of the flow LP in Equation (7.7) is as follows.

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{k} \sum_{v \in V_i} y_v^i + \varepsilon \left(\sum_{i=1}^{k} \sum_{v \in V_i} \alpha_v^i y_v^i + \sum_{i=1}^{k-1} \sum_{v \in V_i \cap V_{i+1}} \beta_v^i \delta_v^i + b_v^i d_v^i \right) \\
\text{s.t.} \quad & \gamma_v^i + y_v^i \geq 1 \quad \forall v \in V_i \cap V_s, \forall i = 1, \dots, k \\
& \gamma_v^i - \gamma_v^{i+1} + \delta_v^i \geq 0 \quad \forall v \in (V_i \cap V_{i+1}) \cap V_s, \forall i = 1, \dots, k-1 \\
& \gamma_v^{i+1} - \gamma_v^i + d_v^i \geq 0 \quad \forall v \in (V_i \cap V_{i+1}) \cap V_s, \forall i = 1, \dots, k-1 \\
& \gamma_v^i - \gamma_u^i \geq 0 \quad \forall uv \in E_i \text{ such that } u \in V_s, v \in V_t, \forall i = 1, \dots, k \\
& -\gamma_v^i + y_v^i \geq 0 \quad \forall v \in V_i \cap V_t, \forall i = 1, \dots, k \\
& \gamma_v^{i+1} - \gamma_v^i + \delta_v^i \geq 0 \quad \forall v \in (V_i \cap V_{i+1}) \cap V_t, \forall i = 1, \dots, k-1 \\
& \gamma_v^i - \gamma_v^{i+1} + d_v^i \geq 0 \quad \forall v \in (V_i \cap V_{i+1}) \cap V_t, \forall i = 1, \dots, k-1 \\
& y^i \in \mathbb{R}_{\geq 0}^{V_i} \quad \forall i = 1, \dots, k \\
& \delta^i \in \mathbb{R}_{\geq 0}^{V_i \cap V_{i+1}} \quad \forall i = 1, \dots, k-1 \\
& d^i \in \mathbb{R}_{\geq 0}^{V_i \cap V_{i+1}} \quad \forall i = 1, \dots, k-1 \\
& \gamma^i \in \mathbb{R}^{V_i} \quad \forall i = 1, \dots, k
\end{aligned}
\tag{7.8}
$$

Lemma 7.6. *The feasible region of* (7.8) *is an integral polyhedron.*

Proof. The constraint matrix of a maximum flow LP is TU; in particular the constraint matrix of (7.7) is TU.

The transpose of a TU matrix is TU. The constraint matrix of a dual LP is the transpose of the constraint matrix of the primal. Together they imply that the constraint matrix of (7.8) is TU.

Finally, a polyhedron with a TU constraint matrix and integral right hand side is an integral polyhedron; in particular, the feasible region of (7.8) is an integral polyhedron. □

We map an integral optimal solution for (7.8) to an integral optimal solution for (7.6) in two steps. First we map it to the LP in Equation (7.9).

Lemma 7.7. *We can obtain an optimal solution for* (7.9) *from an optimal solution for* (7.8).

Proof.

Claim 7.3. *Any feasible solution for* (7.9) *can be mapped to a feasible solution for* (7.8) *with the same objective value.*

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{k} 1^\top y^i + \varepsilon \left(\sum_{i=1}^{k} \sum_{v \in V_i} \alpha_v^i y_v^i + \sum_{i=1}^{k-1} \sum_{v \in V_i \cap V_{i+1}} \beta_v^i \delta_v^i + b_v^i d_v^i \right) \\
\text{s.t.} \quad & y_u^i + y_v^i \geq 1 \quad \forall uv \in E_i, \forall i = 1, \dots, k \\
& y^i \in \mathbb{R}_{\geq 0}^{V_i} \quad \forall i = 1, \dots, k \\
& y_v^i - y_v^{i+1} \leq \delta_v^i \quad \forall v \in V_i \cap V_{i+1}, \forall i = 1, \dots, k-1 \\
& y_v^{i+1} - y_v^i \leq d_v^i \quad \forall v \in V_i \cap V_{i+1}, \forall i = 1, \dots, k-1 \\
& \delta^i \in \mathbb{R}_{\geq 0}^{V_i \cap V_{i+1}} \quad \forall i = 1, \dots, k-1 \\
& d^i \in \mathbb{R}_{\geq 0}^{V_i \cap V_{i+1}} \quad \forall i = 1, \dots, k-1
\end{aligned}
\tag{7.9}
$$

Proof. Let $\hat{\mu} = (\hat{y}^i$ for $i = 1, \dots, k, \hat{\delta}^i$ for $i = 1, \dots, k-1, \hat{d}^i$ for $i = 1, \dots, k-1)$ be a feasible solution for (7.9). We extend it to a solution $\hat{\mu}_{\text{ext}}$ for (7.8) by setting $\hat{\gamma}_v^i = 1 - \hat{y}_v^i$ for all $v \in V_i \cap V_s$, $i = 1, \dots, k$, and $\hat{\gamma}_v^i = \hat{y}_v^i$ for all $v \in V_i \cap V_t$, $i = 1, \dots, k$. It is clear that $\hat{\mu}_{\text{ext}}$ has the same objective value as $\hat{\mu}$, as the objective functions of the two linear programs are the same and do not involve the γ variables. We next show that $\hat{\mu}_{\text{ext}}$ is feasible.

The constraints $\gamma_v^i + y_v^i \geq 1$ and $-\gamma_v^i + y_v^i \geq 0$ of (7.8) are satisfied by $\hat{\mu}_{\text{ext}}$ by definition. Nonnegativity of y^i for $i = 1, \dots, k$, and of δ^i and d^i for $i = 1, \dots, k-1$ are satisfied by $\hat{\mu}_{\text{ext}}$, because $\hat{\mu}$ is feasible for (7.9).

Let $i = 1, \dots, k-1$ and $v \in (V_i \cap V_{i+1}) \cap V_s$. We have

$$\hat{\gamma}_v^i - \hat{\gamma}_v^{i+1} + \hat{\delta}_v^i = 1 - \hat{y}_v^i - (1 - \hat{y}_v^{i+1}) + \hat{\delta}_v^i = \hat{y}_v^{i+1} - \hat{y}_v^i + \hat{\delta}_v^i \geq 0,$$

and

$$\hat{\gamma}_v^{i+1} - \hat{\gamma}_v^i + \hat{d}_v^i = 1 - \hat{y}_v^{i+1} - (1 - \hat{y}_v^i) + \hat{d}_v^i = \hat{y}_v^i - \hat{y}_v^{i+1} + \hat{d}_v^i \geq 0,$$

where both times the last inequality follows from $\hat{\mu}$'s feasibility for (7.9). Similarly, we have for $i = 1, \dots, k-1$ and $v \in (V_i \cap V_{i+1}) \cap V_t$

$$\hat{\gamma}_v^{i+1} - \hat{\gamma}_v^i + \hat{\delta}_v^i = \hat{y}_v^{i+1} - \hat{y}_v^i + \hat{\delta}_v^i \geq 0,$$

and

$$\hat{\gamma}_v^i - \hat{\gamma}_v^{i+1} + \hat{d}_v^i = \hat{y}_v^i - \hat{y}_v^{i+1} + \hat{d}_v^i \geq 0.$$

Finally, let $i = 1, \dots, k$ and $uv \in E_i$ such that $u \in V_s$ and $v \in V_t$. We have

$$\hat{\gamma}_v^i - \hat{\gamma}_u^i = \hat{y}_v^i - (1 - \hat{y}_u^i) = \hat{y}_v^i + \hat{y}_u^i - 1 \geq 0.$$

□

Claim 7.4. *Any feasible solution to* (7.8) *can be mapped to a feasible solution to* (7.9) *with the same objective value.*

Proof. Let $\hat{\mu} = (\hat{y}^i$ for $i = 1, \ldots, k, \hat{\delta}^i$ for $i = 1, \ldots, k-1, \hat{d}^i$ for $i = 1, \ldots, k-1, \hat{\gamma}^i$ for $i = 1, \ldots, k)$ be a feasible solution for (7.8). We restrict it to a solution $\hat{\mu}_{\text{res}}$ for (7.9) by disregarding the γ variables. It is clear that $\hat{\mu}_{\text{res}}$ has the same objective value as $\hat{\mu}$, as the objective functions of the two linear programs are the same and do not involve the γ variables. We next show that $\hat{\mu}_{\text{res}}$ is feasible.

First observe that $\hat{\mu}_{\text{res}} \geq 0$.

Let $i = 1, \ldots, k$ and $uv \in E_i$ such that $u \in V_s$ and $v \in V_t$. We have

$$\hat{y}_u^i + \hat{y}_v^i = (\hat{\gamma}_u^i + \hat{y}_u^i) + (\hat{\gamma}_v^i - \hat{\gamma}_u^i) + (-\hat{\gamma}_v^i + \hat{y}_v^i) \geq 1 + 0 + 0 = 1.$$

Let $i = 1, \ldots, k$, $v \in (V_i \cap V_{i+1}) \cap V_s$. We will show that without loss of generality we can assume that $\hat{\gamma}_u^j + \hat{y}_u^j = 1$ for $u \in V_j \cap V_S$, $j = 1, \ldots, k$, so in particular for $u = v$ and $j = i$, $j = i+1$. Consequently, we have

$$\begin{aligned}
\hat{\delta}_v^i + \hat{y}_v^{i+1} - \hat{y}_v^i &= \hat{\delta}_v^i + \hat{y}_v^{i+1} - \hat{y}_v^i + \hat{\gamma}_v^{i+1} - \hat{\gamma}_v^{i+1} + \hat{\gamma}_v^i - \hat{\gamma}_v^i \\
&= (\hat{\delta}_v^i + \hat{\gamma}_v^i - \hat{\gamma}_v^{i+1}) + (\hat{\gamma}_v^{i+1} + \hat{y}_v^{i+1}) - (\hat{\gamma}_v^i + \hat{y}_v^i) \\
&\geq 0 + 1 - 1 = 0,
\end{aligned}$$

and

$$\begin{aligned}
\hat{d}_v^i + \hat{y}_v^i - \hat{y}_v^{i+1} &= \hat{d}_v^i + \hat{y}_v^i - \hat{y}_v^{i+1} + \hat{\gamma}_v^i - \hat{\gamma}_v^i + \hat{\gamma}_v^{i+1} - \hat{\gamma}_v^{i+1} \\
&= (\hat{d}_v^i + \hat{\gamma}_v^{i+1} - \hat{\gamma}_v^i) + (\hat{\gamma}_v^i + \hat{y}_v^i) - (\hat{\gamma}_v^{i+1} + \hat{y}_v^{i+1}) \\
&\geq 0 + 1 - 1 = 0.
\end{aligned}$$

Now, let $i = 1, \ldots, k$, $v \in (V_i \cap V_{i+1}) \cap V_t$. We will also show that without loss of generality we can assume that $-\hat{\gamma}_u^j + \hat{y}_u^j = 0$ for $u \in V_j \cap V_t$, $j = 1, \ldots, k$. Consequently, we have

$$\begin{aligned}
\hat{\delta}_v^i + \hat{y}_v^{i+1} - \hat{y}_v^i &= \hat{\delta}_v^i + \hat{y}_v^{i+1} - \hat{y}_v^i + \hat{\gamma}_v^{i+1} - \hat{\gamma}_v^{i+1} + \hat{\gamma}_v^i - \hat{\gamma}_v^i \\
&= (\hat{\delta}_v^i + \hat{\gamma}_v^{i+1} - \hat{\gamma}_v^i) + (-\hat{\gamma}_v^{i+1} + \hat{y}_v^{i+1}) - (-\hat{\gamma}_v^i + \hat{y}_v^i) \\
&\geq 0 + 0 - 0 = 0,
\end{aligned}$$

and

$$\begin{aligned}
\hat{d}_v^i + \hat{y}_v^i - \hat{y}_v^{i+1} &= \hat{d}_v^i + \hat{y}_v^i - \hat{y}_v^{i+1} + \hat{\gamma}_v^i - \hat{\gamma}_v^i + \hat{\gamma}_v^{i+1} - \hat{\gamma}_v^{i+1} \\
&= (\hat{d}_v^i + \hat{\gamma}_v^i - \hat{\gamma}_v^{i+1}) + (-\hat{\gamma}_v^i + \hat{y}_v^i) - (-\hat{\gamma}_v^{i+1} + \hat{y}_v^{i+1}) \\
&\geq 0 + 0 - 0 = 0,
\end{aligned}$$

This finishes the feasibility proof of $\hat{\mu}_{\text{res}}$.

To show that we can indeed assume without loss of generality that $\hat{\gamma}_v^i + \hat{y}_v^i = 1$ for $i = 1, \ldots, k$, $v \in V_i \cap V_s$, suppose it does not hold, so: $\hat{\gamma}_v^i + \hat{y}_v^i > 1$ for some $i \in \{1, \ldots, k\}$ and $v \in V_i \cap V_s$. In particular, let $\hat{\gamma}_v^i + \hat{y}_v^i > 1$ for all i in a range, that is, for all $i \in \{j, \ldots, j+l\}$ for some $j \in \{1, \ldots, k\}$ and integral $l \geq 0$, such that either $v \notin V_{j-1}$ or $\hat{\gamma}_v^{j-1} + \hat{y}_v^{j-1} = 1$, likewise for $j+l+1$.

If $\hat{y}_v^i > 0$ for some $i \in \{j, \ldots, j+l\}$, then lower $\hat{y}_v^i$ by $\min\{\hat{y}_v^i, 1 - \hat{\gamma}_v^i\}$ (this does not affect feasibility as $\hat{y}_v^i$ is contained in only this one constraint, and it improves the objective). If now $\hat{\gamma}_v^i + \hat{y}_v^i = 1$, then we continue with a smaller range, otherwise we continue with the same range. So, we can assume that for all $i \in \{j, \ldots, j+l\}$, we have $\hat{y}_v^i = 0$ and $\hat{\gamma}_v^i > 1$. Let $\eta > 0$ be small enough such that if we set $(\hat{\gamma}_v^i)' = \hat{\gamma}_v^i - \eta$ (this does not change the objective) for all $i \in \{j, \ldots, j+l\}$, we still have $(\hat{\gamma}_v^i)' \geq 1$.

For $i \in \{j, \ldots, j+l\}$ and $u \in V_t$ such that $uv \in E_i$, we have

$$\hat{\gamma}_u^i - (\hat{\gamma}_v^i)' = \hat{\gamma}_u^i - (\hat{\gamma}_v^i - \eta) = \hat{\gamma}_u^i - \hat{\gamma}_v^i + \eta \geq \eta > 0.$$

For $i \in \{j, \ldots, j+l-1\}$, we have

$$(\hat{\gamma}_v^i)' - (\hat{\gamma}_v^{i+1})' + \hat{\delta}_v^i = (\hat{\gamma}_v^i - \eta) - (\hat{\gamma}_v^{i+1} - \eta) + \hat{\delta}_v^i = \hat{\gamma}_v^i - \hat{\gamma}_v^{i+1} + \hat{\delta}_v^i \geq 0,$$

and

$$(\hat{\gamma}_v^{i+1})' - (\hat{\gamma}_v^i)' + \hat{d}_v^i = (\hat{\gamma}_v^{i+1} - \eta) - (\hat{\gamma}_v^i - \eta) + \hat{d}_v^i = \hat{\gamma}_v^{i+1} - \hat{\gamma}_v^i + \hat{d}_v^i \geq 0.$$

If $v \in V_{i-1}$, then $\hat{\gamma}_v^{i-1} + \hat{y}_v^{i-1} = 1$, and so

$$\hat{\gamma}_v^{i-1} - (\hat{\gamma}_v^i)' + \hat{\delta}_v^{i-1} = \hat{\gamma}_v^{i-1} - (\hat{\gamma}_v^i - \eta) + \hat{\delta}_v^{i-1} = \hat{\gamma}_v^{i-1} - \hat{\gamma}_v^i + \hat{\delta}_v^{i-1} + \eta \geq \eta > 0,$$

and

$$\begin{aligned}(\hat{\gamma}_v^i)' - \hat{\gamma}_v^{i-1} + \hat{\delta}_v^{i-1} &= (\hat{\gamma}_v^i - \eta) - (1 - \hat{y}_v^{i-1}) + \hat{\delta}_v^{i-1} \\ &= (\hat{\gamma}_v^i - \eta - 1) + \hat{y}_v^{i-1} + \hat{\delta}_v^{i-1} \geq 0 + 0 + 0 = 0.\end{aligned}$$

If $v \in V_{j+l+1}$, then $\hat{\gamma}_v^{j+l+1} + \hat{y}_v^{j+l+1} = 1$, and so

$$\begin{aligned}(\hat{\gamma}_v^{j+l})' - \hat{\gamma}_v^{j+l+1} + \hat{\delta}_v^{j+l} &= (\hat{\gamma}_v^{j+l} - \eta) - (1 - \hat{y}_v^{j+l+1}) + \hat{\delta}_v^{j+l} \\ &= (\hat{\gamma}_v^{j+l} - \eta - 1) + \hat{y}_v^{j+l+1} + \hat{\delta}_v^{j+l} \geq 0 + 0 + 0 = 0,\end{aligned}$$

and

$$\begin{aligned}\hat{\gamma}_v^{j+l+1} - (\hat{\gamma}_v^{j+l})' + \hat{\delta}_v^{j+l} &= \hat{\gamma}_v^{j+l+1} - (\hat{\gamma}_v^{j+l} - \eta) + \hat{\delta}_v^{j+l} \\ &= \hat{\gamma}_v^{j+l+1} - \hat{\gamma}_v^{j+l} + \hat{\delta}_v^{j+l} + \eta \geq \eta > 0.\end{aligned}$$

So we find that the solution with $\hat{\gamma}_v^i$ replaced by $(\hat{\gamma}_v^i)'$ for all $i \in \{j, \ldots, j+l\}$ is feasible. By possibly repeating this argument, we can decrease all $\hat{\gamma}_v^i$ to 1, such that $\hat{\gamma}_v^i + \hat{y}_v^i = 1$ holds for all $i \in \{1, \ldots, k\}$ and $v \in V_i \cap V_s$.

By similar arguments we can show that we can also assume without loss of generality that $-\hat{\gamma}_v^i + \hat{y}_v^i = 0$ for $i = 1, \dots, k$, $v \in V_i \cap V_t$. $\square$

Now we can map an optimal solution for (7.8) to a solution for (7.9) with the same objective value, as described above. This solution is optimal for (7.9), as otherwise we could find a better solution, map the better solution back to a solution for (7.8) with the same objective value, contradicting the optimality of the starting solution. $\square$

Lemma 7.8. *An integral optimal solution for* (7.9) *is also an integral optimal solution for* (7.6).

Proof. Let $\hat{\gamma} = (\hat{y}^i$ for $i = 1, \dots, k, \hat{\delta}^i$ for $i = 1, \dots, k-1, \hat{d}^i$ for $i = 1, \dots, k-1)$ be an integral optimal solution for (7.9).

Suppose that $1^\top \hat{y}^i > \nu(G_i)$ for some $i = 1, \dots, k$, then because $\hat{y}^i$ is integral, $1^\top \hat{y}^i \geq \nu(G) + 1$. Now replace $\hat{y}^i$ by a minimum vertex cover $\widetilde{y}^i$, that is, $1^\top \widetilde{y}^i = \nu(G_i)$, and set $\widetilde{\delta}^i$ and $\widetilde{d}^i$ accordingly. We have $\widetilde{y}^i \leq 1$ and $\hat{y}^{i+1} \leq 1$, and hence also $\widetilde{\delta}^i \leq 1$ and $\widetilde{d}^i \leq 1$ (this also holds for superscripts other than i). Therefore,

$$
\begin{aligned}
\varepsilon &\left(\sum_{i=1}^{k} \sum_{v \in V_i} \alpha_v^i \hat{y}_v^i + \sum_{i=1}^{k-1} \sum_{v \in V_i \cap V_{i+1}} \beta_v^i \widetilde{\delta}_v^i + b_v^i \widetilde{d}_v^i \right) \\
&\qquad\qquad \leq \varepsilon \left(\sum_{i=1}^{k} \sum_{v \in V_i} \alpha_v^i + \sum_{i=1}^{k-1} \sum_{v \in V_i \cap V_{i+1}} \beta_v^i + b_v^i \right) \\
&\qquad\qquad \leq \varepsilon \left(\sum_{i=1}^{k} \sum_{v \in V_i} |\alpha_v^i| + \sum_{i=1}^{k-1} \sum_{v \in V_i \cap V_{i+1}} \beta_v^i + b_v^i \right) \\
&\qquad\qquad < \varepsilon \left(1 + \sum_{i=1}^{k} \sum_{v \in V_i} |\alpha_v^i| + \sum_{i=1}^{k-1} \sum_{v \in V_i \cap V_{i+1}} \beta_v^i + b_v^i \right) \\
&\qquad\qquad = 1,
\end{aligned}
$$

which means we obtain a strictly better solution, contradicting that $\hat{\gamma}$ is optimal. So, $1^\top \hat{y}^i = \nu(G_i)$. Hence, $\hat{\gamma}$ is feasible for (7.6).

Suppose $\hat{\gamma}$ is not optimal for (7.6). Let $\widetilde{\gamma}$ be an optimal solution for (7.6).

Then

$$\sum_{k=1}^{k} \sum_{v \in V_i} \alpha_v^i \widetilde{y}_v^i + \sum_{i=1}^{k-1} \sum_{v \in V_i \cap V_{i+1}} \beta_v^i \widetilde{\delta}_v^i + b_v^i \widetilde{d}_v^i$$
$$< \sum_{k=1}^{k} \sum_{v \in V_i} \alpha_v^i \hat{y}_v^i + \sum_{i=1}^{k-1} \sum_{v \in V_i \cap V_{i+1}} \beta_v^i \hat{\delta}_v^i + b_v^i \hat{d}_v^i$$

and $1^\top \widetilde{y}^i = \nu(G_i) = 1^\top \hat{y}^i$ for all $i = 1, \dots, k$. It follows that

$$\sum_{i=1}^{k} 1^\top \widetilde{y}^i + \varepsilon \left(\sum_{k=1}^{k} \sum_{v \in V_i} \alpha_v^i \widetilde{y}_v^i + \sum_{i=1}^{k-1} \sum_{v \in V_i \cap V_{i+1}} \beta_v^i \widetilde{\delta}_v^i + b_v^i \widetilde{d}_v^i \right)$$
$$< \sum_{i=1}^{k} 1^\top \hat{y}^i + \varepsilon \left(\sum_{k=1}^{k} \sum_{v \in V_i} \alpha_v^i \hat{y}_v^i + \sum_{i=1}^{k-1} \sum_{v \in V_i \cap V_{i+1}} \beta_v^i \hat{\delta}_v^i + b_v^i \hat{d}_v^i \right),$$

because $\varepsilon > 0$. This contradicts the optimality of $\hat{\gamma}$ for (7.9), hence $\hat{\gamma}$ must be optimal for (7.6). □

Finally, Lemmas 7.6 to 7.8 prove Theorem 7.4.

Bibliography

[1] Sara Ahmadian, Hamideh Hosseinzadeh and Laura Sanità. 2018. *Stabilizing network bargaining games by blocking players*. In: Mathematical Programming. Volume: 172. Number: 1. Pages: 249–275. DOI: 10.1007/s10107-017-1177-9.

[2] Paola Alimonti and Viggo Kann. 2000. *Some APX-completeness results for cubic graphs*. In: Theoretical Computer Science. Volume: 237. Number: 1. Pages: 123–134. DOI: 10.1016/S0304-3975(98)00158-3.

[3] Gautam Appa and Balázs Kotnyek. 2006. *A bidirected generalization of network matrices*. In: Networks. Volume: 47. Number: 4. Pages: 185–198. DOI: 10.1002/net.20108.

[4] Evripidis Bampis, Bruno Escoffier, Kevin Schewior and Alexandre Teiller. 2021. *Online multistage subset maximization problems*. In: Algorithmica. Volume: 83. Number: 8. Pages: 2374–2399. DOI: 10.1007/s00453-021-00834-7.

[5] Evripidis Bampis, Bruno Escoffier and Paul Youssef. 2023. *Online 2-stage stable matching*. In: Discrete Applied Mathematics. Volume: 341. Pages: 394–405. DOI: 10.1016/j.dam.2023.09.009.

[6] Francisco Barahona. 1993. *Reducing matching to polynomial size linear programming*. In: SIAM Journal on Optimization. Volume: 3. Number: 4. Pages: 688–695. DOI: 10.1137/0803035.

[7] MohammadHossein Bateni, MohammadTaghi Hajiaghayi, Nicole Immorlica and Hamid Mahini. 2010. *The cooperative game theory foundations of network bargaining bames*. In: Automata, Languages and Programming. Pages: 67–78. DOI: 10.1007/978-3-642-14165-2_7.

[8] Péter Biró, Matthijs Bomhoff, Petr A. Golovach, Walter Kern and Daniël Paulusma. 2014. *Solutions for the stable roommates problem with payments*. In: Theoretical Computer Science. Volume: 540-541. Pages: 53–61. DOI: 10.1016/j.tcs.2013.03.027. Combinatorial Optimization: Theory of algorithms and Complexity.

[9] Péter Biró, Walter Kern and Daniël Paulusma. 2010. *On Solution Concepts for Matching Games*. In: Theory and Applications of Models of Computation. Pages: 117–127. DOI: 10.1007/978-3-642-13562-0_12.

[10] Péter Biró, Walter Kern, Daniël Paulusma and Péter Wojuteczky. 2018. *The stable fixtures problem with payments*. In: Games and Economic Behavior. Volume: 108. Pages: 245–268. DOI: 10.1016/j.geb.2017.02.002. Special Issue in Honor of Lloyd Shapley: Seven Topics in Game Theory.

[11] Adrian Bock, Karthekeyan Chandrasekaran, Jochen Könemann, Britta Peis and Laura Sanità. 2015. *Finding small stabilizers for unstable graphs*. In: Mathematical Programming. Volume: 154. Number: 1. Pages: 173–196. DOI: 10.1007/s10107-014-0854-1.

[12] Karthekeyan Chandrasekaran. 2017. *Graph Stabilization: A Survey*. In: Combinatorial Optimization and Graph Algorithms: Communications of NII Shonan Meetings. Pages: 21–41. DOI: 10.1007/978-981-10-6147-9_2.

[13] Karthekeyan Chandrasekaran, Corinna Gottschalk, Jochen Könemann, Britta Peis, Daniel Schmand and Andreas Wierz. 2019. *Additive stabilizers for unstable graphs*. In: Discrete Optimization. Volume: 31. Pages: 56–78. DOI: 10.1016/j.disopt.2018.08.003.

[14] Moses Charikar, Chandra Chekuri and Martin Pál. 2005. *Sampling bounds for stochastic optimization*. In: Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques. Pages: 257–269. DOI: 10.1007/11538462_22.

[15] George B. Dantzig, W.O. Blattner and M.R. Rao. 1966. *Finding a cycle in a graph with minimum cost to time ratio with application to a ship routing problem*. Technical report number 66-1. Stanford University, Department of Operations Research. URL: https://apps.dtic.mil/sti/pdfs/AD0646553.pdf.

[16] Jesús A. De Loera, Sean Kafer and Laura Sanità. 2022. *Pivot rules for circuit-augmentation algorithms in linear optimization*. In: SIAM Journal on Optimization. Volume: 32. Number: 3. Pages: 2156–2179. DOI: 10.1137/21M1419994.

[17] Xiaotie Deng, Toshihide Ibaraki and Hiroshi Nagamochi. 1999. *Algorithmic aspects of the core of combinatorial optimization games*. In: Mathematics of Operations Research. Volume: 24. Number: 3. Pages: 751–766. DOI: 10.1287/moor.24.3.751.

[18] Szymon Dudycz and Katarzyna Paluch. 2018. *Optimal general matchings*. In: Graph-Theoretic Concepts in Computer Science. Pages: 176–189. DOI: 10.1007/978-3-030-00256-5_15.

[19] H.A. Eiselt and Carl-Louis Sandblom. 2000. *Shortest path problems*. In: Integer Programming and Network Models. Pages: 283–313. DOI: 10.1007/978-3-662-04197-0_13.

[20] Yuri Faenza, Ayoub Foussoul and Chengyue He. 2024. *Two-stage stochastic stable matching*. In: Integer Programming and Combinatorial Optimization. Pages: 154–167. DOI: 10.1007/978-3-031-59835-7_12.

[21] Linda Farczadi, Konstantinos Georgiou and Jochen Könemann. 2013. *Network bargaining with general capacities*. In: Algorithms – ESA 2013. Pages: 433–444. DOI: 10.1007/978-3-642-40450-4_37.

[22] Elisabeth Finhold. 2014. *Circuit diameters and their application to transportation problems*. PhD thesis. Technische Universität München. URL: https://mediatum.ub.tum.de/doc/1233751.

[23] Till Fluschnik, Rolf Niedermeier, Valentin Rohm and Philipp Zschoche. 2022. *Multistage vertex cover*. In: Theory of Computing Systems. Volume: 66. Number: 2. Pages: 454–483. DOI: 10.1007/s00224-022-10069-w.

[24] Lester R. Ford and Delbert R. Fulkerson. 1962. *Flows in networks*.

[25] Martin Grötschel, László Lovász and Alexander Schrijver. 1993. *Geometric algorithms and combinatorial optimization*. DOI: 10.1007/978-3-642-78240-4.

[26] Martin Grötschel, László Lovász and Alexander Schrijver. 1981. *The ellipsoid method and its consequences in combinatorial optimization*. In: Combinatorica. Volume: 1. Number: 2. Pages: 169–197. DOI: 10.1007/BF02579273.

[27] Anupam Gupta, Kunal Talwar and Udi Wieder. 2014. *Changing bases: multistage optimization for matroids and matchings*. In: Automata, Languages, and Programming. Pages: 563–575. DOI: 10.1007/978-3-662-43948-7_47.

[28] Magnús M. Halldórsson. 1993. *Approximating the minimum maximal independence number*. In: Information Processing Letters. Volume: 46. Number: 4. Pages: 169–172. DOI: 10.1016/0020-0190(93)90022-2.

[29] Wassily Hoeffding. 1994. *Probability inequalities for sums of bounded random variables*. In: The Collected Works of Wassily Hoeffding. Pages: 409–426. DOI: 10.1007/978-1-4612-0865-5_26.

[30] Takehiro Ito, Naonori Kakimura, Naoyuki Kamiyama, Yusuke Kobayashi and Yoshio Okamoto. 2017. *Efficient stabilization of cooperative matching games*. In: Theoretical Computer Science. Volume: 677. Pages: 69–82. DOI: 10.1016/j.tcs.2017.03.020.

[31] Sean Kafer. 2022. *Polyhedral diameters and applications to optimization.* PhD thesis. University of Waterloo. URL: http://hdl.handle.net/10012/18705.

[32] L.G. Khachiyan. 1980. *Polynomial algorithms in linear programming.* In: USSR Computational Mathematics and Mathematical Physics. Volume: 20. Number: 1. Pages: 53–72. DOI: 10.1016/0041-5553(80)90061-0.

[33] Subhash Khot. 2002. *On the power of unique 2-prover 1-round games.* In: Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing. Pages: 767–775. DOI: 10.1145/509907.510017.

[34] Jon Kleinberg and Éva Tardos. 2008. *Balanced outcomes in social exchange networks.* In: Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing. Pages: 295–304. DOI: 10.1145/1374376.1376994.

[35] Anton J. Kleywegt, Alexander Shapiro and Tito Homem-de-Mello. 2002. *The sample average approximation method for stochastic discrete optimization.* In: SIAM Journal on Optimization. Volume: 12. Number: 2. Pages: 479–502. DOI: 10.1137/S1052623499363220.

[36] Zhuan Khye Koh and Laura Sanità. 2020. *Stabilizing weighted graphs.* In: Mathematics of Operations Research. Volume: 45. Number: 4. Pages: 1318–1341. DOI: 10.1287/moor.2019.1034.

[37] Jochen Könemann, Kate Larson and David Steiner. 2015. *Network Bargaining: Using Approximate Blocking Sets to Stabilize Unstable Instances.* In: Theory of Computing Systems. Volume: 57. Number: 3. Pages: 655–672. DOI: 10.1007/s00224-015-9650-4.

[38] Eugene Lawler. 1976. *Combinatorial optimization: networks and matroids.*

[39] Euiwoong Lee and Sahil Singla. 2020. *Maximum matching in the online batch-arrival model.* In: ACM Transactions on Algorithms. Volume: 16. Number: 4. DOI: 10.1145/3399676.

[40] Adam N. Letchford, Gerhard Reinelt and Dirk Oliver Theis. 2008. *Odd minimum cut sets and b-matchings revisited.* In: SIAM Journal on Discrete Mathematics. Volume: 22. Number: 4. Pages: 1480–1487. DOI: 10.1137/060664793.

[41] John F. Nash. 1950. *The bargaining problem.* In: Econometrica. Volume: 18. Number: 2. Pages: 155–162. DOI: 10.2307/1907266.

[42] Daniël Paulusma. 2001. *Complexity aspects of cooperative games.* PhD thesis. University of Twente. URL: https://research.utwente.nl/en/publications/complexity-aspects-of-cooperative-games.

[43] J. Scott Provan and Michael O. Ball. 1983. *The complexity of counting cuts and of computing the probability that a graph is connected*. In: SIAM Journal on Computing. Volume: 12. Number: 4. Pages: 777–788. DOI: 10.1137/0212053.

[44] Ramamoorthi Ravi and Amitabh Sinha. 2006. *Hedging uncertainty: approximation algorithms for stochastic optimization problems*. In: Mathematical Programming. Volume: 108. Number: 1. Pages: 97–114. DOI: 10.1007/s10107-005-0673-5.

[45] Thomas Rothvoss. 2017. *The matching polytope has exponential extension complexity*. In: Journal of the ACM. Volume: 64. Number: 6. DOI: 10.1145/3127497.

[46] Laura Sanità. 2018. *The diameter of the fractional matching polytope and its hardness implications*. In: IEEE 59th Annual Symposium on Foundations of Computer Science. Pages: 910–921. DOI: 10.1109/FOCS.2018.00090.

[47] Alexander Schrijver. 2003. *Combinatorial optimization: polyhedra and efficiency*.

[48] András Sebő. 1997. *Potentials in undirected graphs and planar multiflows*. In: SIAM Journal on Computing. Volume: 26. Number: 2. Pages: 582–603. DOI: 10.1137/S0097539790186704.

[49] Paul D. Seymour. 1979. *Sums of circuits*. In: Graph Theory and Related Topics. Pages: 341–355.

[50] Lloyd S. Shapley and Martin Shubik. 1971. *The assignment game I: The core*. In: International Journal of Game Theory. Volume: 1. Number: 1. Pages: 111–130. DOI: 10.1007/BF01753437.

[51] Chaitanya Swamy and David B. Shmoys. 2012. *Sampling-based approximation algorithms for multistage stochastic optimization*. In: SIAM Journal on Computing. Volume: 41. Number: 4. Pages: 975–1004. DOI: 10.1137/100789269.

[52] Chaitanya Swamy and David B. Shmoys. 2008. *The sample average approximation method for 2-stage stochastic optimization*. Unpublished manuscript. URL: https://www.math.uwaterloo.ca/~cswamy/papers/SAAproof.pdf.

List of Publications

As convention in the field, the authors are listed in alphabetical order.

Journal Articles

[V1] Matthew Gerstbrein, Laura Sanità and Lucy Verberk. 2025. *Stabilization of capacitated matching games*. In: Mathematical Programming. Volume: 210. Number: 1. Pages: 313–334. DOI: 10.1007/s10107-024-02169-x.

[V2] Laura Sanità and Lucy Verberk. 2025. *A note on the core of 2-matching games*. In: Operations Research Letters. Volume: 61. Pages: 107299. DOI: 10.1016/j.orl.2025.107299.

[V3] Laura Sanità and Lucy Verberk. 2025. *Capacitated Network Bargaining Games: Stability and Structure*. In: Mathematics of Operations Research. DOI: 10.1287/moor.2024.0668. (Available online ahead of print.)

Conference Proceedings

[V4] Matthew Gerstbrein, Laura Sanità and Lucy Verberk. 2023. *Stabilization of capacitated matching games*. In: Integer Programming and Combinatorial Optimization. Pages: 157–171. DOI: 10.1007/978-3-031-32726-1_12.

Preprints

[V5] Laura Sanità and Lucy Verberk. 2025. *Two-stage stochastic assignment games*. arXiv: 2506.01509.

Summary

In this dissertation we study network bargaining games and cooperative matching games. These games are defined on graphs, and involve the structure of matchings. The vertices of the graph represent the players of the game, and the edges represent how players can interact with each other. Most of this dissertation considers c-matchings (capacity-matchings), where each vertex has a capacity that indicates how often the vertex can be used in a matching.

There is a notion of stable solutions for network bargaining games, which nicely relates to some structural property of the graph, called stability. Not all graphs are stable, which naturally yields the problem of stabilizing graphs. We study this in the first part of this dissertation: We consider stabilizing a graph by removing a minimum number of vertices, reducing a minimum amount of capacity and removing a minimum number of edges. We show that stabilizing by removing vertices is APX-hard when there are vertex capacities. On the other hand, stabilizing by reducing the capacity of vertices is solvable in polynomial time and our algorithm generalizes the unit-capacity algorithm for removing vertices. Stabilizing by removing edges was already known to be NP-hard in unit-capacity graphs. We generalize the unit-capacity approximation algorithm. For the latter two results we use new polyhedral techniques.

There is a variation of the stabilization problem where in addition a c-matching is given that needs to be avoided by the stabilizer, and needs to have maximum weight in the resulting graph. In unit-capacity graphs it is known that stabilizing by removing vertices with this additional constraint is NP-hard but attains a 2-approximation algorithm, and can be solved exactly if the given matching has maximum weight. We show that both of these algorithmic results generalize to capacitated graphs.

In the second part of this dissertation we focus on cooperative matching games. First we study the core of 2-matching games (cooperative matching games where all vertex capacities are at most two): we discuss how to separate over the core of 2-matching games in polynomial time, and prove the existence of a compact extended formulation for it. Next we study two-stage stochastic cooperative matching games in bipartite graphs, where in the second stage the players and their possible interactions are sampled from a distribution. The goal here is to compute a first-stage core element that minimizes the expected total decrease in the second stage. We show that if the second-stage distribution is given explicitly, this problem can be solved in polynomial time. On the other hand, if the second-stage distribution is given implicitly by a sampling oracle, the problem is computationally hard, but it can be approximated with the SAA method.

Acknowledgements

I would like to thank everyone that contributed to the writing of this dissertation, in any way.

First and foremost, I want to thank Laura, my supervisor, for introducing me to the research world, and teaching me how to be a good researcher. Even though you moved to Milan at the end of my first year, we kept meeting once or twice every week. I appreciate the amount of time you had for these meetings, where we could think about the problems together. Your guidance throughout my PhD was invaluable. It was a pleasure to work together with you.

Next, I want to thank Frits. It has been great to be part of your research group, the CO group. We first got to work together, when you involved me in the supervision of the final project of two master students. Then in the last year of my PhD, you became my second promotor. I thank you for taking the time to carefully read this dissertation and providing detailed and valuable feedback.

I would like to thank my committee members for taking the time to read this dissertation and providing valuable feedback.

I would like to thank all my colleagues for providing a great and fun work environment these last four years, in particular, my colleagues in the CO group. The lunches and snack competitions were a relaxing distraction during work. I also enjoyed the SPOR drinks and pizza nights, together with everyone from the 4th floor. I especially want to thank Nils and Emanuel, whom I shared the office with the last year, and who occasionally listened to Taylor Swift with me during work. Thinking back to how this journey started, I want to give special thanks to Céline and Sjanne, who answered all of my questions about what it is like to do a PhD.

Lastly, I want to thank my friends and family for their unconditional love and support. In particular:

Zoë, bedankt voor bijna iedere week een ontspannen avondje bijkletsen en series kijken. Laura en Yori, mijn lieve zusjes, bedankt dat jullie tijdens de verdediging naast me staan. Mama en John, papa en Yvonne, Pim en Miranda, Doriene en Martin, bedankt dat jullie er altijd voor mij en Gijs zijn, en dat we altijd bij jullie terecht kunnen. In het bijzonder, mama en papa, bedankt voor jullie onvoorwaardelijke liefde, voor de vrijheid om mijn eigen keuzes te maken, voor het vertrouwen en de steun die jullie hebben in die keuzes, en voor het vertrouwen in mij. Bedankt voor alles.

Tot slot, Gijs. Ten eerste wil ik je bedanken voor je directe hulp bij mijn PhD, zoals als eerste na mijn presentaties luisteren, en de proloog van dit proefschrift doorlezen. Ik waardeer je feedback enorm. Maar ik wil je vooral bedanken voor al 8 jaar samen. Van samen in mijn kleine studio zonder uitzicht, tot samen in ons appartement op de 11e verdieping. Van onze vakanties en dagjes uit, tot onze gaming dagjes in. Van samen studeren, tot tegelijkertijd een PhD doen, en nu op naar het volgende avontuur.

Lucy Verberk
Eindhoven, September 2025

About the Author

Lucy Verberk was born on July 15, 1998 in Sint-Oedenrode, the Netherlands. She completed her secondary school education in 2016 at Zwijsen College Veghel. She proceeded to study Applied Mathematics at Eindhoven University of Technology, where she obtained her bachelor's degree in 2019 and her master's degree in 2021, specializing in combinatorial optimization. She stayed at Eindhoven University of Technology to start her PhD in October 2021 under the supervision of dr. Laura Sanità. The results of the research done during her PhD are presented in this dissertation.