

Stabilizing Weighted Graphs

by

Zhuan Khye Koh

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Combinatorics and Optimization

Waterloo, Ontario, Canada, 2017

© Zhuan Khye Koh 2017

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

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

Abstract

An edge-weighted graph $G = (V, E)$ is called *stable* if the value of a maximum-weight matching equals the value of a maximum-weight *fractional* matching. Stable graphs play an important role in some interesting game theory problems, such as *network bargaining* games and *cooperative matching* games, because they characterize instances which admit stable outcomes. Motivated by this, in the last few years many researchers have investigated the algorithmic problem of turning a given graph into a stable one, via edge- and vertex-removal operations. However, all the algorithmic results developed in the literature so far only hold for *unweighted* instances, i.e., assuming unit weights on the edges of G .

We give the first polynomial-time algorithm to find a minimum cardinality subset of vertices whose removal from G yields a stable graph, for any weighted graph G . The algorithm is combinatorial and exploits new structural properties of basic fractional matchings, which may be of independent interest. In contrast, we show that the problem of finding a minimum cardinality subset of edges whose removal from a weighted graph G yields a stable graph, does not admit any constant-factor approximation algorithm, unless $P = NP$. In this setting, we develop an $O(\Delta)$ -approximation algorithm for the problem, where Δ is the maximum degree of a node in G .

Acknowledgements

I would like to extend my deepest gratitude to my advisor, Laura Sanità, without whom this body of work would not have been possible. Her enthusiasm in research is infectious, and has inspired me to probe the boundaries of my knowledge. The many hours we spent brainstorming and discussing ideas have greatly influenced my approach to problem solving. I was encouraged and given the freedom to explore new methods to better understand the problem at hand. Moreover, she taught me how to properly communicate mathematics, both verbally and in writing. I am grateful for the opportunity to work and grow as a researcher under her guidance.

I would like to thank my committee members, Chaitanya Swamy and Joseph Cheriyan, for expressing interest in my work. Their invaluable feedback has improved the structure and presentation of this thesis. They also raised interesting questions, allowing me to view the problem tackled in this thesis from a new perspective.

I would also like to thank all the other faculty members at C&O for creating a conducive and supportive environment for learning mathematics. Special thanks to Melissa, Carol and the administrative team for taking care of me throughout my graduate studies.

During my stint at the department, I have had the pleasure of making many friends – to name a few: Sharat, Dhinakaran, Hemant, Anirudh, Karthik, Abhinav, Justin, Priya, Sanchit, Jimit, Mali, Edward, ... You are one of those people who make my life better just by being in it. Special thanks to Nishad for being a great mentor and a great friend. I would also like to give a shout out to my frat boys Jimmy, Dean and Alex. I will always cherish the memories and good times we share.

Lastly, I am thankful to my parents and brother for their unconditional love and support.

To my beloved parents, Elene and Stanley.

Your unwavering support of my endeavours
means the world to me.

Table of Contents

List of Figures	vii
1 Introduction	1
1.1 Our results and techniques	3
1.2 Related work	5
2 Preliminaries and notation	7
2.1 Minimum stabilizers do not preserve $\nu(G)$	10
3 Maximum fractional matching with minimum support	12
4 Computing vertex-stabilizers	20
5 Computing edge-stabilizers	29
6 Forcing an outcome	34
6.1 Finding augmenting walks	36
6.2 The algorithm	40
6.3 Proof of Theorem 2.4	43
References	50

List of Figures

2.1	Simple examples of a blossom, a flower and a bi-cycle.	8
2.2	An example showing that the removal of any minimum edge-stabilizer does not preserve $\nu(G)$. Bold edges indicate the maximum-weight matching. . .	10
2.3	An example showing that the removal of any minimum vertex-stabilizer does not preserve $\nu(G)$. Bold edges indicate the maximum-weight matching. . .	11
3.1	Two examples of connectors. Bold edges indicate M . Vertices of the same color belong to the same terminal set. White vertices are the ones in S_0 . . .	13
3.2	The auxiliary graph G' and the matching M' . Vertices in the ellipse are from the original graph G . Gray vertices represent pseudonodes.	17
4.1	An example of the graph induced by $\text{supp}(\hat{x} + \bar{x})$ in Subcase 2.1. Black bold edges are in $M(\bar{x})$ while gray bold edges are in $M(\hat{x})$	23
4.2	An example of the graph induced by $\text{supp}(x^* + \bar{x})$ in Subcase 2.2. Black bold edges are in $M(x^*)$ while gray bold edges are in $M(\bar{x})$	23
4.3	An example of the graph induced by $\text{supp}(x^* + \hat{x})$ in Subcase 2.3. Black bold edges are in $M(x^*)$ while gray bold edges are in $M(\hat{x})$	24
4.4	An example showing that the bound of $\frac{2}{3}$ can be asymptotically tight.	25
4.5	The gadget graph G^*	26
5.1	The gadget graph G^*	30
5.2	An example showing that $\gamma(G)$ is not a lower bound. The unique maximum-weight matching is shown in the left, while the unique maximum-weight fractional matching is shown in the right. Gray edges have value $\frac{1}{2}$	32

6.1	An example of $G(S)$ for a given graph and matching.	44
-----	--	----

Chapter 1

Introduction

Several interesting game theory problems are defined on networks, where the vertices represent players and the edges model the way players can interact with each other. In many such games, the structure of the underlying graph that describes the interactions among players is essential in determining the existence of stable outcomes for the corresponding games, i.e., outcomes where players have no incentive to deviate. Popular examples are *co-operative matching* games, introduced by Shapley and Shubik [21], and *network bargaining* games, defined by Kleinberg and Tardos [16], both extensively studied in the game theory community. Instances of such games are described by a graph $G = (V, E)$ with edge weights $w \in \mathbb{R}_{\geq 0}^E$, where V represents a set of players, and the value of a *maximum-weight matching*, denoted as $\nu(G)$, is the total value that the players could get by interacting with each other.

An important role in such games is played by so-called *stable* graphs. An edge-weighted graph $G = (V, E)$ is called *stable* if the value $\nu(G)$ of a maximum-weight matching equals the value of a maximum-weight *fractional* matching, denoted as $\nu_f(G)$. Formally, $\nu_f(G)$ is given by the optimal value of the standard linear programming relaxation of the matching problem, defined as

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

Here x is a vector in \mathbb{R}^E , $\delta(v)$ denotes the set of edges incident to the node v , and for a set $F \subseteq E$, $x(F) = \sum_{e \in F} x_e$. Feasible solutions of the above LP are called *fractional matchings*.

The relation that interplays between stable graphs and network games is as follows. In cooperative matching games [21], the goal is to find an allocation of the value $\nu(G)$ among

the vertices, given as a vector $y \in \mathbb{R}_{\geq 0}^V$, such that no subset $S \subseteq V$ has an incentive to form a *coalition* to deviate. This condition is formally defined by the constraints $\sum_{v \in S} y_v \geq \nu(G[S])$, $\forall S \subseteq V$, where $G[S]$ denotes the subgraph induced by S , and an allocation y that satisfies the above set of constraints is called *stable*. Deng et al. [11] proved that a stable allocation exists if and only if the graph describing the game is a stable graph. This is an easy consequence from LP duality. If y is a stable allocation, then y is a feasible dual solution of value $\nu(G)$, showing that $\nu_f(G) = \nu(G)$. Conversely, if $\nu_f(G) = \nu(G)$, then an optimal dual solution yields a stable allocation of $\nu(G)$.

In network bargaining games [16], each edge e represents a deal of value w_e . A player can enter in a deal with at most one neighbor, and when a deal is made, the players have to agree on how to split the value of the deal between them. An outcome of the game is given by a pair (M, y) , where M is a matching of G and stands for the set of deals made by the players, and $y \in \mathbb{R}_{\geq 0}^V$ is an allocation vector representing how the deal values have been split. Kleinberg and Tardos have defined a notion of *stable* outcome for such games, as well as a notion of *balanced* outcome, that are outcomes where players have no incentive to deviate, and in addition the deal values are “fairly” split among players. They proved that a balanced outcome exists if and only if a stable outcome exists, and this happens if and only if the graph G describing the game is stable.

Motivated by the above connection, in the last few years many researchers have investigated the algorithmic problem of turning a given graph into a stable one, by performing a minimum number of modifications on the input graph [8, 1, 13, 9, 17, 6, 7]. Two natural operations which have a nice network game interpretation, are vertex-deletion and edge-deletion. They correspond to *blocking players* and *blocking deals*, respectively, in order to achieve stability in the corresponding games. Formally, a subset of vertices $S \subseteq V$ is called a *vertex-stabilizer* if the graph $G \setminus S := G[V \setminus S]$ is stable. Similarly, a subset of edges $F \subseteq E$ is called an *edge-stabilizer* if the graph $G \setminus F := (V, E \setminus F)$ is stable. The corresponding optimization problems, which are the focus of this thesis, are:

Minimum Vertex-stabilizer: *Given an edge-weighted graph $G = (V, E)$, find a minimum-cardinality vertex-stabilizer.*

Minimum Edge-stabilizer: *Given an edge-weighted graph $G = (V, E)$, find a minimum-cardinality edge-stabilizer.*

The above problems have been studied quite intensively in the last few years on unweighted graphs. In particular, Bock et al. [8] have showed that finding a minimum-cardinality edge-stabilizer is hard to approximate within a factor of $(2 - \varepsilon)$, assuming Unique Game Conjecture (UGC) [14]. On the positive side, they have given an approximation algorithm for the edge-stabilizer problem, whose approximation factor depends on

the sparsity of the input graph G . In other work, Ahmadian et al. [1] and Ito et al. [13] have shown independently that finding a minimum-cardinality vertex-stabilizer is a polynomial-time solvable problem. These (exact and approximate) algorithmic results, developed for unweighted instances, do not easily generalize when dealing with arbitrary edge-weights, since they heavily rely on the structure of maximum matchings in unweighted graphs. In fact, unweighted instances of the above problems exhibit a very nice property, as shown in [8, 1]: the removal of any inclusion-wise minimal edge-stabilizer (resp. vertex-stabilizer) from a graph G *does not decrease* the cardinality of a maximum matching in the resulting graph. This property ensures that there is at least one maximum-cardinality matching that survives in the modified graph, and this insight can be successfully exploited when designing (exact and approximate) algorithms. Unfortunately, it is not difficult to realize that this crucial property does not hold anymore when dealing with edge-weighted graphs (see Section 2.1), and in fact, the development of algorithmic results for weighted graphs requires substantial new ideas.

1.1 Our results and techniques

Our main results are as follows:

Vertex-stabilizers. We give the first polynomial-time algorithm to find a minimum-cardinality vertex-stabilizer S , in any weighted graph G . Our algorithm also ensures that $\nu(G \setminus S) \geq \frac{2}{3}\nu(G)$, i.e., the value of a maximum-weight matching is preserved up to a factor of $\frac{2}{3}$, and we show that this factor is tight in general. Specifically, as previously mentioned, a minimum-cardinality vertex-stabilizer for a weighted graph might decrease the value of a maximum-weight matching in the resulting graph. From a network bargaining perspective, this means we are decreasing the total value that the players are able to get, which is of course undesirable. However, we can show this is inevitable, since deciding whether there exists *any* vertex-stabilizer S that preserves the value of a maximum-weight matching (i.e., such that $\nu(G \setminus S) = \nu(G)$) is an NP-hard problem. Furthermore, we give an example of a graph G where *any* vertex-stabilizer S decreases the value of a maximum-weight matching by a factor of essentially $\frac{2}{3}$, i.e., $\nu(G \setminus S) \leq (\frac{2}{3} + \varepsilon)\nu(G)$ (for an arbitrary small $\varepsilon > 0$). This shows that the bounds of our algorithm are essentially best possible: the algorithm finds a vertex-stabilizer S whose cardinality is the *smallest* possible, and preserves the value of a maximum-weight matching up to a factor of $\frac{2}{3}$, that is the tightest factor that holds for all instances.

The algorithm exploits new structural properties of basic fractional matchings which may be of independent interest. Its analysis relies on a suitable lower bound on the car-

dinality of a minimum vertex-stabilizer, which generalizes the lower bound used in the unweighted setting, and is based on the structure of optimal basic solutions of (P). In particular, it was shown in [1] that a lower bound on the cardinality of a vertex-stabilizer for unweighted graphs is given by the minimum number of *odd cycles* in the support of an optimal basic solution to (P). We show that this lower bound holds also for weighted graphs, though the proof is much more involved and requires different ideas. As a byproduct of our results, we develop a combinatorial algorithm for computing an optimal basic solution to (P) with the smallest number of odd cycles in its support, for weighted graphs.

Edge-stabilizers. When dealing with edge-removal operations, the problem becomes harder, already in the unweighted setting. It is shown in [8] that finding a minimum edge-stabilizer is as hard as vertex cover, and whether the problem admits a constant factor approximation algorithm is an interesting open question. We here show that the answer to this question is negative for weighted graphs, since we prove that the minimum edge-stabilizer problem for a weighted graph G does not admit any constant-factor approximation algorithm, unless $P = NP$. From an approximation point of view, we show that the algorithm we developed for the vertex-stabilizer problem translates into a $O(\Delta)$ -approximation algorithm for the edge-stabilizer problem, where Δ is the maximum degree of a node in G .

Once again, the analysis relies on proving a lower bound on the cardinality of a minimum edge-stabilizer. It was shown in [8] that a lower-bound on the cardinality on a minimum edge-stabilizer for unweighted graphs is again given by the minimum number of odd cycles in the support of an optimal basic solution to (P) (called $\gamma(G)$). Interestingly, we show that, differently from the vertex-stabilizer setting, here this lower bound does not generalize, and $\gamma(G)$ is *not* a lower bound on the cardinality of an edge-stabilizer for arbitrary weighted graphs. However, we are able to show that $\lceil \gamma(G)/2 \rceil$ is a lower bound on the cardinality of a minimum edge-stabilizer, and this is enough for our approximation purposes.

Additional results. Lastly, we also generalize a result given in [1] on finding a minimum vertex-stabilizer which avoids a fixed maximum matching M , on unweighted graphs. We prove that if M is a maximum-weight matching of a weighted graph G , then finding a minimum vertex-stabilizer that is element-disjoint from M is a polynomial-time solvable problem. Otherwise, if M is not a maximum-weight matching, the problem is at least as hard as vertex cover. We supplement this result with a 2-approximation algorithm for this case, that is best possible assuming UGC.

1.2 Related work

Biró et al. [6] were the first to consider the edge-stabilizer problem in weighted graphs, and they showed NP-hardness for this case through a reduction from the independent set problem. Stabilizing a graph via different operations on the input graph (other than removing edges/vertices) has also been studied. In particular, Ito et al. [13] have given polynomial-time algorithms to stabilize an unweighted graph by adding edges and by adding vertices. They also considered the scenario in which one can add different edges at different costs. Specifically, given a graph $G = (V, E)$ and a cost function $c : \binom{V}{2} \setminus E \rightarrow \mathbb{R}_{\geq 0}$, they were interested in finding a set of edges $F \subseteq \binom{V}{2} \setminus E$ such that the unweighted graph $(V, E \cup F)$ is stable and $c(F)$ is minimized. They proved that this problem is NP-hard, and left the complexity of the vertex addition variant as an open problem.

Chandrasekaran et al. [9] have recently studied the problem of stabilizing unweighted graphs by fractionally increasing edge weights. Given an unweighted graph $G = (V, E)$, a *fractional additive stabilizer* is a vector $w \in \mathbb{R}_{\geq 0}^E$ such that the graph G with edge weights $\mathbb{1} + w$ is stable. They proved that finding a minimum additive stabilizer, that is, a fractional additive stabilizer of the smallest weight $\mathbb{1}^\top w$, has no $o(\log |V|)$ -approximation unless $P = NP$. In addition, they showed that an $O(|V|^{1/24-\varepsilon})$ -approximation algorithm for this problem where $\varepsilon > 0$ would lead to an $O(|V|^{1/4-6\varepsilon})$ -approximation algorithm for the *densest k -subgraph* problem. This would be an improvement over the currently best known approximation factor of $\approx O(|V|^{1/4})$ for densest k -subgraph [5]. On the positive side, they gave a polynomial time algorithm to compute a minimum additive stabilizer in factor-critical graphs. They also presented a (non-polytime) algorithm to compute a minimum additive stabilizer for any graph G . Its runtime is exponential in $|A(G)|$, where $A(G)$ is the Tutte set of G .

Ahmadian et al. [1] have also studied the vertex-stabilizer problem on unweighted graphs, but in the more-general setting where there are (non-uniform) costs for removing vertices. Given a graph $G = (V, E)$ with vertex costs $c_v \geq 0 \forall v \in V$, they considered the following two problems. The *min-cost* vertex-stabilizer problem asks to find a vertex-stabilizer S which minimizes $c(S)$, while the *max-cost* vertex-stabilizer problem asks to find a vertex-stabilizer S which maximizes $c(V \setminus S)$. They showed that both problems are NP-hard even if there are only two different costs. They gave a 2-approximation algorithm for the max-cost vertex-stabilizer problem, and an $O(|A(G)|)$ -approximation algorithm for the min-cost vertex-stabilizer problem, where $A(G)$ is the Tutte set of G . Independently, Ito et al. [13] also proved that the min-cost vertex-stabilizer problem is NP-hard.

Biró et al. [7] and Könemann et al. [17] studied a variant of the problem where the goal is

to compute a minimum-cardinality set of *blocking pairs*, that are, edges whose removal from the graph yields the existence of a fractional vertex cover of size at most $\nu(G)$. Note that the resulting graph might not be stable. Biró et al. showed that this problem is NP-hard, even on unweighted graphs, and this is achieved via a reduction from the independent set problem. Könemann et al. complemented this result by giving an approximation algorithm on sparse graphs. A graph $G = (V, E)$ is ω -sparse if $|E(S)| \leq \omega |S|$ for all $S \subseteq V$. Given an ω -sparse graph, their algorithm returns a set of blocking pairs of size at most $8\omega + 2$ times the optimum.

Mishra et al. [18] studied the problem of converting a graph into a *König-Egerváry graph*, via vertex-deletion and edge-deletion operations. A König-Egerváry graph is a graph where the size of a maximum matching equals the size of an (integral) minimum vertex cover. They gave an $O(\log n \log \log n)$ -approximation algorithm for the vertex-removal setting in unweighted graphs, and showed constant-factor hardness of approximation (assuming UGC) for both the minimum vertex-removal and edge-removal problem.

Thesis Organization. In Chapter 2 we give some preliminaries and discuss notation. In Chapter 3, we give a polynomial-time algorithm to compute an optimal basic solution to (P) with a minimum number of odd cycles in its support. This algorithm will be crucially used in Chapter 4, where we give our results on vertex-stabilizers. Chapter 5 reports our results on edge-stabilizers. Finally, Chapter 6 contains our additional results.

Chapter 2

Preliminaries and notation

A key concept that we will use is LP duality. The dual of (P) is given by

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

As feasible solutions to (P) are called fractional matchings, we call feasible solutions to (D) *fractional w -vertex covers*. In fact, (D) is the standard LP-relaxation of the problem of finding a minimum w -vertex cover, obtained by adding integrality constraints on (D). We also call basic feasible solutions to (P) as *basic fractional matchings*. An application of duality theory yields the following relationship $\nu(G) \leq \nu_f(G) = \tau_f(G)$. Recall that a graph G is *stable* if $\nu(G) = \nu_f(G) = \tau_f(G)$.

For a vector $x \in \mathbb{R}^E$ and any subset $F \subseteq E$, we denote $x_{-F} \in \mathbb{R}^{E-F}$ as the subvector obtained by dropping the entries corresponding to F . For any multisubset $F \subseteq E$, we define $x(F) := \sum_{e \in F} x_e$. Note that an element may be accounted for multiple times in the sum if it appears more than once in F . We denote $\text{supp}(x) := \{e \in E : x_e \neq 0\}$ as the support of x . For any positive integer k , $[k]$ represents the set $\{1, 2, \dots, k\}$.

Given an undirected graph G , we denote by n the number of vertices and by m the number of edges. For edge weights $w \in \mathbb{R}_+^m$ and a matching M in G , a path is called *M -alternating* if its edges alternately belong to M and $E \setminus M$. We say that an M -alternating path is *valid* if it starts with an M -exposed vertex or an edge in M , and ends with an M -exposed vertex or an edge in M . A valid M -alternating path P is called *M -augmenting* if $w(P \setminus M) > w(P \cap M)$. An *M -augmenting walk* and an *M -augmenting cycle* are defined similarly. Recall that a walk is a path that is non-simple. Note that an M -alternating cycle has even length.

Definition 2.1. An odd cycle $C = (e_1, e_2, \dots, e_{2k+1})$ is called an M -blossom if $e_i \in M$ for all even i and $e_i \notin M$ for all odd i . The vertex $v := e_1 \cap e_{2k+1}$ is called the *base* of the blossom. The blossom is *augmenting* if v is M -exposed and $w(C \setminus M) > w(C \cap M)$.

Definition 2.2. An M -flower $C \cup P$ consists of an M -blossom C with base v_1 and a valid M -alternating path $P = (v_1, v_2, \dots, v_k)$ where $v_1 v_2 \in M$. The vertex v_k is called the *root* of the flower. The flower is *augmenting* if

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

Given an M -augmenting flower $C \cup P$, if we replace the vector which places 1 on the edges of $M \cap (C \cup P)$, with the vector that places $\frac{1}{2}$ on the edges of C and 1 on the edges of $P \setminus M$, then the change in weight is exactly $\frac{1}{2}$ times LHS – RHS of the above inequality. So the inequality means that this operation increases the weight.

Definition 2.3. An M -bi-cycle $C \cup P \cup D$ consists of two M -blossoms C, D with bases v_1, v_k respectively and an odd M -alternating path $P = (v_1, v_2, \dots, v_k)$ where $v_1 v_2, v_{k-1} v_k \in M$. The bi-cycle is *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 the structures defined in Definition 2.2 and 2.3 might not be simple. For example, in a flower $C \cup P$, the path P might intersect the blossom C more than once. Figure 2.1 illustrates some simple examples of these structures. Notice that a blossom is always simple.

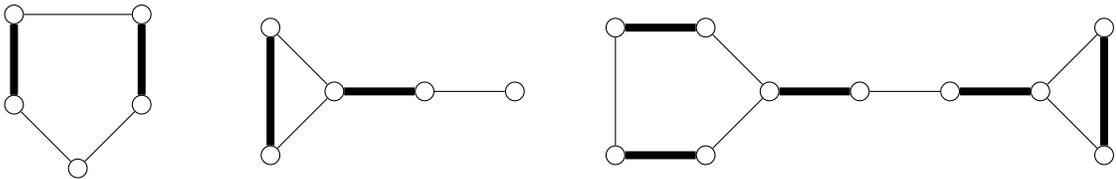


Figure 2.1: Simple examples of a blossom, a flower and a bi-cycle.

The significance of the structures defined above is given by the following theorem:

Theorem 2.4 ([16]). *If a graph is stable, then it does not have an M -augmenting flower or bi-cycle for every maximum-weight matching M . Otherwise, it has an M -augmenting flower or bi-cycle for every maximum-weight matching M .*

In the original paper, Theorem 2.4 was stated without proof. For the sake of completeness, we will provide a proof in Section 6.3. We will also need the following classical result on the structure of basic fractional matchings:

Theorem 2.5 ([4]). *A fractional matching x in $G = (V, E)$ is basic if and only if $x_e = \{0, \frac{1}{2}, 1\}$ for all $e \in E$ and the edges e having $x_e = \frac{1}{2}$ induce vertex-disjoint odd cycles in G .*

Let \hat{x} be a basic fractional matching in G . We partition the support of \hat{x} into two parts. Define

$$\mathcal{C}(\hat{x}) := \{C_1, \dots, C_q\}$$

as the set of odd cycles such that $\hat{x}_e = \frac{1}{2}$ for all $e \in E(C_i)$, and

$$M(\hat{x}) := \{e \in E : \hat{x}_e = 1\}$$

as the set of matched edges in \hat{x} . For ease of notation, we use $V(\mathcal{C}(\hat{x})) = \cup_{C \in \mathcal{C}(\hat{x})} V(C)$ and $E(\mathcal{C}(\hat{x})) = \cup_{C \in \mathcal{C}(\hat{x})} E(C)$ to denote the vertex set and edge set of $\mathcal{C}(\hat{x})$ respectively. We define two operations on the entries of \hat{x} associated with certain edge sets of G :

Definition 2.6. By *complementing* on $E' \subseteq E$, we mean replacing \hat{x}_e by $\bar{x}_e = 1 - \hat{x}_e$ for all $e \in E'$.

Definition 2.7. By *alternate rounding* on $C \in \mathcal{C}(\hat{x})$ at v where $C = \{e_1, \dots, e_{2k+1}\}$ and $v = e_1 \cap e_{2k+1}$, we mean replacing \hat{x}_e by $\bar{x}_e = 0$ for all $e \in \{e_1, e_3, \dots, e_{2k+1}\}$ and $\bar{x}_e = 1$ for all $e \in \{e_2, e_4, \dots, e_{2k}\}$. When v is clear from the context, we just say alternate rounding on C .

Let \mathcal{X} be the set of basic maximum-weight fractional matchings in G . Define

$$\gamma(G) := \min_{\hat{x} \in \mathcal{X}} |\mathcal{C}(\hat{x})|.$$

Note that G is stable if and only if $\gamma(G) = 0$.

We will use the following terminology given in [10] for the description of Edmonds' maximum matching algorithm. Given a graph G and a matching M , let T be an M -alternating tree rooted at a vertex r . We denote by $A(T)$ and $B(T)$ the sets of nodes in T at odd and even distance respectively from r . We call T *frustrated* if every edge of G having one end in $B(T)$ has the other end in $A(T)$.

Finally, the following theorem gives a sufficient condition for a graph to be Hamiltonian.

Theorem 2.8 (Ore's Theorem [20]). *Let G be a finite and simple graph with $n \geq 3$ vertices. If $\deg(u) + \deg(v) \geq n$ for every pair of distinct non-adjacent vertices u and v , then G is Hamiltonian.*

2.1 Minimum stabilizers do not preserve $\nu(G)$

In this section, we demonstrate that the removal of a minimum edge- or vertex-stabilizer does not preserve the value of a maximum-weight matching. We first look at edge-stabilizers. Let G denote the graph in Figure 2.2. It is not stable because $\nu(G) = 8 < 9 = \nu_f(G)$, where the maximum-weight fractional matching is given by

$$x_e = \begin{cases} \frac{1}{2}, & \text{if } e \in \{pq, qr, pr\} \\ 1, & \text{if } e = st \\ 0, & \text{otherwise.} \end{cases}$$

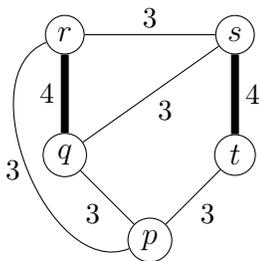


Figure 2.2: An example showing that the removal of any minimum edge-stabilizer does not preserve $\nu(G)$. Bold edges indicate the maximum-weight matching.

The minimum edge-stabilizer is $\{qr\}$. Observe that $M = \{pq, st\}$ is a maximum-weight matching in $G \setminus \{qr\}$ of weight 7. We can verify the stability of $G \setminus \{qr\}$ by constructing a fractional w -vertex cover y of the same weight:

$$y_v = \begin{cases} 3, & \text{if } v \in \{p, s\} \\ 1, & \text{if } v = t \\ 0, & \text{otherwise.} \end{cases}$$

It is left to show that if we delete any edge other than qr , the graph is still unstable. If edge rs , qs or pt is removed, the maximum-weight fractional matching remains the same. If edge pr is removed, we get $\nu_f(G \setminus \{pr\}) = 8.5$ by assigning $x_e = 1/2$ for all $e \in \{pq, qr, rs, st, pt\}$. If edge pq is removed, we get $\nu_f(G \setminus \{pq\}) = 8.5$ by assigning $x_e = 1/2$ for all $e \in \{pr, rq, qs, st, pt\}$. If edge st is removed, we get $\nu(G \setminus \{st\}) = 7$ and

$\nu_f(G \setminus \{st\}) = 8$ where the maximum-weight matching is $\{qr, pt\}$ and the maximum-weight fractional matching is

$$x_e = \begin{cases} \frac{1}{2}, & \text{if } e \in \{qr, rs, qs\} \\ 1, & \text{if } e = pt \\ 0, & \text{otherwise.} \end{cases}$$

The same negative result also holds for vertex-stabilizers. Consider the graph given in Figure 2.3. It is not stable because $\nu(G) = 5 < 6 = \nu_f(G)$, where the maximum-weight fractional matching is given by

$$y_v = \begin{cases} \frac{1}{2}, & \text{if } e \in \{pq, qr, pr\} \\ 0, & \text{otherwise.} \end{cases}$$

The minimum vertex-stabilizers of this graph are $\{p\}$, $\{q\}$ and $\{r\}$. However, $\nu(G \setminus p) = \nu(G \setminus q) = \nu(G \setminus r) = 4$.

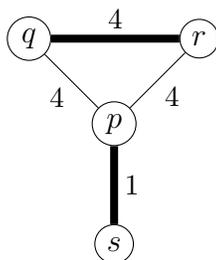


Figure 2.3: An example showing that the removal of any minimum vertex-stabilizer does not preserve $\nu(G)$. Bold edges indicate the maximum-weight matching.

Chapter 3

Maximum fractional matching with minimum support

In this section, we give a polynomial-time algorithm to compute a basic maximum-weight fractional matching \hat{x} for a weighted graph G with minimum number of odd cycles in its support, i.e., satisfying $|\mathcal{C}(\hat{x})| = \gamma(G)$. This algorithm will be used as a subroutine by our vertex-stabilizer algorithm, which we will develop in Section 4.

Our first step is to characterize basic maximum-weight fractional matchings which have more than $\gamma(G)$ odd cycles. Balas [3] considered this problem on unweighted graphs, and gave the following characterization:

Theorem 3.1 ([3]). *Let \hat{x} be a basic maximum fractional matching in an unweighted graph G . If $|\mathcal{C}(\hat{x})| > \gamma(G)$, then there exists an $M(\hat{x})$ -alternating path which connects two odd cycles $C_i, C_j \in \mathcal{C}(\hat{x})$. Furthermore, alternate rounding on the odd cycles and complementing on the path produces a basic maximum fractional matching \bar{x} such that $\mathcal{C}(\bar{x}) \subset \mathcal{C}(\hat{x})$.*

We generalize this to weighted graphs. Before stating the theorem, we need to introduce the concept of *connector* (see Figure 3.1 for some examples):

Definition 3.2. Let C be a cycle and S_0, S_1, \dots, S_k be a partition of $V(C)$ such that $k \geq 2$ and $|S_0| \geq 0$ is even. Note that we allow S_0 to be empty. Let M be a perfect matching on the vertex set S_0 . We call the graph $C \cup M$ a *connector*. Each S_i is called a *terminal set* for $i \geq 1$. An edge $e \in M$ is called a *chord* if $e \notin E(C)$.

Connectors are useful because of the following property:

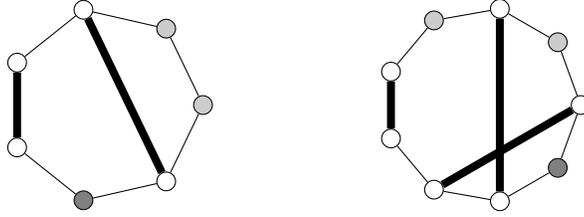


Figure 3.1: Two examples of connectors. Bold edges indicate M . Vertices of the same color belong to the same terminal set. White vertices are the ones in S_0 .

Lemma 3.3. *Let $C \cup M$ be a connector. For every terminal set S_i , there exists an M -augmenting path in the connector from a vertex $v \in S_i$ to a vertex $u \in S_j$, for some $j \neq i$.*

Proof. For every $e \in M \cap E(C)$, contract e and smooth away the vertex formed after the contraction (smoothing is the reverse operation of subdivision). The only edges in M that survive this process are the chords. Fix i and identify all the vertices in S_i into a single vertex v_i . Denote the resulting (multi)graph as $G = (V, E)$. Observe that there exists an M -augmenting path from S_i to S_j in $C \cup M$ if and only if there exists an M -augmenting path from v_i to S_j in G where $i \neq j$. Hence, we will work on the reduced graph G .

Apply Edmonds' maximum matching algorithm on G initialized with the matching $M \cap E$, and construct an M -alternating tree starting with the exposed vertex v_i . There are two possibilities: either we find an augmenting path from v_i to S_j for some $j \neq i$ or a frustrated tree rooted at v_i . For the purpose of contradiction, suppose we get a frustrated tree T rooted at v_i . Let $\tilde{T} = T \cup D$, where $D = \{uv \notin E(T) : u \in A(T), v \in B(T)\}$. Note that we do not have edges connecting two nodes in $B(T)$, otherwise T is not a frustrating tree.

We claim that each pseudonode in T is incident to at least two unmatched edges in \tilde{T} . Let v be a pseudonode in T , and $S(v)$ be the subset of vertices in G that are contained in v (after expanding pseudonodes). Note that $S(v) \subset V$ because there are at least two exposed vertices in G . Let $\delta_{G \setminus M}(\cdot)$ denote the cut function on $G \setminus M$. Since $G \setminus M$ is 2-edge-connected, we have $|\delta_{G \setminus M}(S(v))| \geq 2$. These edges are present in \tilde{T} because otherwise we can extend the alternating tree T . It follows that v is incident to at least two unmatched edges in \tilde{T} .

Let uv be a matched edge in T where $u \in A(T)$ and $v \in B(T)$. We claim that $\deg_{\tilde{T}}(u) \leq \deg_{\tilde{T}}(v)$. Note that $\deg_{\tilde{T}}(u)$ is either 2 or 3. This is because u is not a pseudonode, and $\deg_G(w) = 3$ for every M -covered vertex w in G . If v is not a pseudonode,

then $\deg_{\tilde{T}}(v) = 3$ as all edges in $\delta_G(v)$ are accounted for in \tilde{T} . Otherwise, if v is a pseudonode, then by the previous claim v is incident to at least two unmatched edges in \tilde{T} . So $\deg_{\tilde{T}}(v) \geq 3$.

Now, observe that \tilde{T} is a bipartite graph as the node set can be partitioned into $A(T)$ and $B(T)$ where $|B(T)| = |A(T)| + 1$. For every $v \in A(T)$, let $M(v)$ be its matched neighbour in $B(T)$. The extra node in $B(T)$ is the root of T , which has degree at least one in \tilde{T} . Summing up the node degrees in $A(T)$, we obtain

$$\sum_{v \in A(T)} \deg_{\tilde{T}}(v) \leq \sum_{v \in A(T)} \deg_{\tilde{T}}(M(v)) < \sum_{v \in A(T)} \deg_{\tilde{T}}(M(v)) + 1 \leq \sum_{v \in B(T)} \deg_{\tilde{T}}(v)$$

which is a contradiction. □

Let y be a minimum fractional w -vertex cover in G . We say that an edge uv is *tight* if $y_u + y_v = w_{uv}$. Similarly, we say that a path is tight if all of its edges are tight.

Theorem 3.4. *Let \hat{x} be a basic maximum-weight fractional matching and y be a minimum fractional w -vertex cover in G . If $|\mathcal{C}(\hat{x})| > \gamma(G)$, then there exists*

- (i) *a vertex $v \in V(C_i)$ for some odd cycle $C_i \in \mathcal{C}(\hat{x})$ such that $y_v = 0$; or*
- (ii) *a tight $M(\hat{x})$ -alternating path P which connects two odd cycles $C_i, C_j \in \mathcal{C}(\hat{x})$; or*
- (iii) *a tight and valid $M(\hat{x})$ -alternating path P which connects an odd cycle $C_i \in \mathcal{C}(\hat{x})$ and a vertex $v \notin V(\mathcal{C}(\hat{x}))$ such that $y_v = 0$.*

Furthermore, alternate rounding on the odd cycles and complementing on the path produces a basic maximum-weight fractional matching \bar{x} such that $\mathcal{C}(\bar{x}) \subset \mathcal{C}(\hat{x})$.

Proof. We will start by proving the second part of the theorem. For Case (i), let \bar{x} be the basic fractional matching obtained by alternate rounding on C_i at v . Since $y_v = 0$, both \bar{x} and y satisfy complementary slackness. Hence, \bar{x} is optimal to (P) and $\mathcal{C}(\bar{x}) = \mathcal{C}(\hat{x}) \setminus C_i$. For Case (ii), denote $u = V(P) \cap V(C_i)$ and $v = V(P) \cap V(C_j)$ as the endpoints of P . Let \bar{x} be the basic fractional matching obtained by alternate rounding on C_i, C_j at u, v respectively and complementing on P . Note that u and v are exposed after the alternate rounding, and covered after complementing. Since \bar{x} and y satisfy complementary slackness, \bar{x} is optimal to (P) and $\mathcal{C}(\bar{x}) = \mathcal{C}(\hat{x}) \setminus \{C_i, C_j\}$. For Case (iii), denote $u = V(P) \cap V(C_i)$ and $v \notin V(\mathcal{C}(\hat{x}))$ as the endpoints of P . Let \bar{x} be the basic fractional matching obtained by alternate rounding on C_i at u and complementing on P . Since $y_v = 0$, both \bar{x} and y satisfy complementary slackness. Thus, \bar{x} is optimal to (P) and $\mathcal{C}(\bar{x}) = \mathcal{C}(\hat{x}) \setminus C_i$.

Next, we prove the first part of the theorem. We may assume $y_v > 0$ for every vertex $v \in V(\mathcal{C}(\hat{x}))$. Let x^* be a basic maximum-weight fractional matching in G such that $|\mathcal{C}(x^*)| = \gamma(G)$. Define $N(\hat{x}) := M(\hat{x}) \setminus E(\mathcal{C}(x^*))$ and $N(x^*) := M(x^*) \setminus E(\mathcal{C}(\hat{x}))$. Consider the following subgraph

$$J = (V, N(\hat{x}) \Delta N(x^*)).$$

Since $N(\hat{x})$ and $N(x^*)$ are matchings in G , J is made up of vertex-disjoint paths and cycles of G . For each such path or cycle, its edges alternately belong to $N(\hat{x})$ or $N(x^*)$. Moreover, its intermediate vertices are disjoint from $\mathcal{C}(\hat{x})$ and $\mathcal{C}(x^*)$. Since \hat{x} and x^* are maximum-weight fractional matchings in G , every path in J is tight by complementary slackness. If there exists a path in J which connects two odd cycles from $\mathcal{C}(\hat{x})$, then we are done. If there exists a path in J which connects an odd cycle from $\mathcal{C}(\hat{x})$ and a vertex $v \notin V(\mathcal{C}(\hat{x}) \cup \mathcal{C}(x^*))$, then $y_v = 0$ because v is either exposed by $M(\hat{x})$ or $M(x^*)$. Hence, we are also done. So we may assume every path in J belongs to one of the following three categories:

- (a) Vertex disjoint from $\mathcal{C}(\hat{x})$ and $\mathcal{C}(x^*)$.
- (b) Starts and ends at the same cycle.
- (c) Connects an odd cycle from $\mathcal{C}(\hat{x})$ and an odd cycle from $\mathcal{C}(x^*)$.

Note that by the second part of the theorem, there is no path in J which connects two odd cycles from $\mathcal{C}(x^*)$ or an odd cycle from $\mathcal{C}(x^*)$ and a vertex $v \notin V(\mathcal{C}(\hat{x}) \cup \mathcal{C}(x^*))$. We say that two odd cycles C_i and C_j are *adjacent* if $V(C_i) \cap V(C_j) \neq \emptyset$ or if they are connected by a path in J .

Claim 3.5. *There exists an odd cycle in $\mathcal{C}(x^*)$ which is adjacent to at least two odd cycles in $\mathcal{C}(\hat{x})$.*

Proof. Let H be the subgraph of G induced by $\text{supp}(\hat{x} + x^*)$. Since $|\mathcal{C}(\hat{x})| > |\mathcal{C}(x^*)|$, there exists a component K in H which has more odd cycles from $\mathcal{C}(\hat{x})$ than $\mathcal{C}(x^*)$. We claim that K contains at least one odd cycle from $\mathcal{C}(x^*)$. Let $C \in \mathcal{C}(\hat{x})$ be an odd cycle in K . For every $v \in V(C)$, since we assumed $y_v > 0$, by complementary slackness it is either $M(x^*)$ -covered or belongs to $V(\mathcal{C}(x^*))$. If $v \in V(\mathcal{C}(x^*))$, then we are done. So we may assume that every vertex in C is $M(x^*)$ -covered. Let $uv \in M(x^*)$ where $u \in V(C)$ and $v \notin V(C)$. Observe that uv is the first edge of a path in J , so it either ends at an odd cycle in $\mathcal{C}(x^*)$ or C . Since C has an odd number of vertices, by the pigeonhole principle there exists a path in J which connects C and an odd cycle in $\mathcal{C}(x^*)$. Thus, we have shown that K contains at least one odd cycle from $\mathcal{C}(x^*)$. This implies that K contains at least two odd cycles from $\mathcal{C}(\hat{x})$. Let $C_1^*, \dots, C_k^* \in \mathcal{C}(x^*)$ be the odd cycles adjacent to C . Each C_i^*

is not adjacent to any other odd cycle in $\mathcal{C}(x^*)$. Note that by our assumption, C is not adjacent to any other odd cycle in $\mathcal{C}(\hat{x})$. As K contains at least two odd cycles from $\mathcal{C}(\hat{x})$, there exists a C_i^* which is adjacent to an odd cycle in $\mathcal{C}(\hat{x}) \setminus C$. \square

In light of the previous claim, let $C^* \in \mathcal{C}(x^*)$ be adjacent to $C_1, \dots, C_k \in \mathcal{C}(\hat{x})$ for some $k \geq 2$. For every $i \in [k]$, define

$$S_i = \{v \in V(C^*) : v \in V(C_i) \text{ or } \exists \text{ a path in } J \text{ from } v \text{ to } C_i\}$$

and $S_0 = V(C^*) \setminus \cup_{i=1}^k S_i$. Since $y_v > 0$ for every $v \in S_0$, by complementary slackness conditions, they are $M(\hat{x})$ -covered. Let $v \in S_0$. It is either matched to another vertex in S_0 or is an endpoint of a path in J whose other endpoint is also a vertex in S_0 . Hence, $|S_0|$ is even. Moreover, $S_i \neq \emptyset$ for all $i \geq 1$, and the sets S_0, \dots, S_k partition $V(C^*)$. Let \mathcal{P} be the set of paths in J that start and end at C^* , and consider the subgraph $C^* \cup \mathcal{P}$. We claim that there exists an $M(\hat{x})$ -alternating path from S_i to S_j in $C^* \cup \mathcal{P}$ for some $j \neq i$. Since every path in \mathcal{P} starts and ends with an edge in $M(\hat{x})$, we can perform the following reduction: contract every path in \mathcal{P} into a single edge in $M(\hat{x})$. It is easy to see that an $M(\hat{x})$ -alternating path from S_i to S_j in $C^* \cup \mathcal{P}$ corresponds to an $M(\hat{x})$ -alternating path from S_i to S_j in the reduced graph. Observe that the reduced graph along with the matching $M(\hat{x})$ forms a connector. By Lemma 3.3, there exists an $M(\hat{x})$ -alternating path P from S_i to S_j in $C^* \cup \mathcal{P}$ for some $j \neq i$.

Let $v_i \in S_i$ and $v_j \in S_j$ be the endpoints of P . Let P_i and P_j be the paths in J connecting v_i to C_i and v_j to C_j respectively. If $v_i \in V(C_i)$, set $P_i = \emptyset$. Similarly if $v_j \in V(C_j)$, set $P_j = \emptyset$. Then, $P_i \cup P \cup P_j$ forms a tight $M(\hat{x})$ -alternating path which connects C_i and C_j . \square

Corollary 3.6. *Let \hat{x} be a basic maximum-weight fractional matching in G . Then, there exists a basic maximum-weight fractional matching x^* in G such that $|\mathcal{C}(x^*)| = \gamma(G)$ and $\mathcal{C}(x^*) \subseteq \mathcal{C}(\hat{x})$.*

Given a basic maximum-weight fractional matching \hat{x} in G , we would like to reduce the number of odd cycles in $\mathcal{C}(\hat{x})$ to $\gamma(G)$. One way to accomplish this is to search for the structures described in Theorem 3.4. Fix a minimum fractional w -vertex cover y in G . Let G' be the unweighted graph obtained by applying the following operations to G (see Figure 3.2):

- (a) Delete all non-tight edges.
- (b) Add a vertex z .
- (c) For every vertex $v \in V$ where $\hat{x}(\delta(v)) = 1$ and $y_v = 0$, add the edge vz .

- (d) For every vertex $v \in V$ where $\hat{x}(\delta(v)) = 0$ and $y_v = 0$, add the vertex v' and the edges $vv', v'z$.
- (e) Shrink every odd cycle $C_i \in \mathcal{C}(\hat{x})$ into a pseudonode i .

Note that none of the edges in $M(\hat{x})$ and $\mathcal{C}(\hat{x})$ were deleted because they are tight. Consider the edge set $M' := M(\hat{x}) \cup \{vv' : v \in V\}$. It is easy to see that M' is a matching in G' . The significance of the auxiliary graph G' is given by the following lemma:

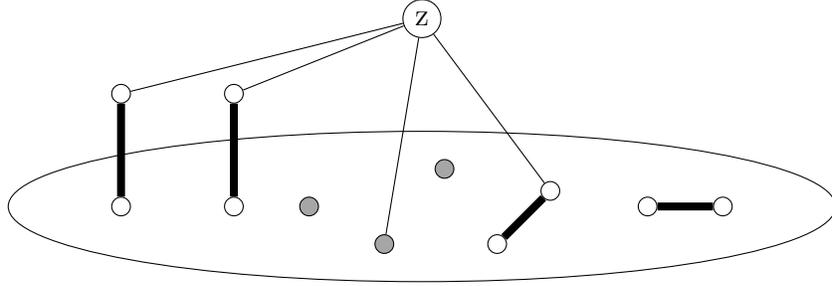


Figure 3.2: The auxiliary graph G' and the matching M' . Vertices in the ellipse are from the original graph G . Gray vertices represent pseudonodes.

Lemma 3.7. M' is a maximum matching in G' if and only if $|\mathcal{C}(\hat{x})| = \gamma(G)$.

Proof. (\Rightarrow) Let \hat{x} be a basic maximum-weight fractional matching where $|\mathcal{C}(\hat{x})| > \gamma(G)$ and y be a minimum fractional w -vertex cover in G . Applying Theorem 3.4 yields three cases. In Case (i), there exists a vertex $v \in C_i$ for some odd cycle $C_i \in \mathcal{C}(\hat{x})$ such that $y_v = 0$. Then, the edge iz is an M' -augmenting path in G' . In Case (ii), there exists a tight $M(\hat{x})$ -alternating path P in G connecting two odd cycles $C_i, C_j \in \mathcal{C}(\hat{x})$. In G' , P is an M' -augmenting path whose endpoints are pseudonodes i and j . In Case (iii), there exists a tight and valid $M(\hat{x})$ -alternating path P in G connecting an odd cycle $C_i \in \mathcal{C}(\hat{x})$ and a vertex $v \notin V(\mathcal{C}(\hat{x}))$ such that $y_v = 0$. If v is $M(\hat{x})$ -covered, then $P + vz$ is an M' -augmenting path in G' . Otherwise, $P + vv' + v'z$ is an M' -augmenting path in G' . Thus, M' is not a maximum matching in G' .

(\Leftarrow) Assume M' is not a maximum matching in G' . Then, there exists an M' -augmenting path P in G' . If both of its endpoints are pseudonodes i and j , then P is a tight $M(\hat{x})$ -alternating path in G which connects C_i and C_j . So we may assume the endpoints of P are a pseudonode i and z . If $iz \in E(P)$, then there exists a vertex $v \in V(C_i)$ such that $y_v = 0$. If $vz \in E(P)$ for some $v \in V$, then $y_v = 0$ and v is $M(\hat{x})$ -covered. Hence, $P - vz$ is a tight and valid $M(\hat{x})$ -alternating path in G connecting C_i and v . Otherwise, $v'z \in E(P)$

for some $v \in V$, which implies that $y_v = 0$ and v is $M(\hat{x})$ -exposed. Hence, $P - vv' - v'z$ is a tight and valid $M(\hat{x})$ -alternating path in G connecting C_i and v . By Theorem 3.4, $|\mathcal{C}(\hat{x})| > \gamma(G)$. \square

Thus, searching for the structures in Theorem 3.4 is equivalent to searching for an M' -augmenting path in G' . This immediately gives an algorithm to generate a basic maximum-weight fractional matching with $\gamma(G)$ odd cycles.

Algorithm 1: Minimize number of odd cycles

```

1 Compute a basic maximum-weight fractional matching  $\hat{x}$  in  $G$ 
2 Compute a minimum fractional  $w$ -vertex cover  $y$  in  $G$ 
3 Construct  $G'$  and  $M'$ 
4 while  $\exists$  an  $M'$ -exposed pseudonode  $r$  in  $G'$  do
5   | Grow an  $M'$ -alternating tree  $T$  rooted at  $r$  using Edmonds' algorithm
6   | if an  $M'$ -augmenting  $r$ - $s$  path  $P'$  is found in  $G'$  then
7     |   | Let  $P$  be the corresponding tight  $M(\hat{x})$ -alternating path in  $G$ 
8     |   | if  $s$  is a pseudonode then
9     |   |   | Alternate round on  $C_r, C_s$  and complement on  $P$ 
10    |   | else
11    |   |   | Alternate round on  $C_r$  and complement on  $P$ 
12    |   |   | Update  $G'$  and  $M'$ 
13    |   | else
14    |   |   |  $G' \leftarrow G' \setminus V(T)$ 
15 return  $\hat{x}$ 

```

After an M' -augmenting path P' is found, let \bar{x} denote the new basic maximum-weight fractional matching in G obtained by alternate rounding and complementing \hat{x} . We can update G' as follows. If s is a pseudonode, we unshrink C_r and C_s in G' because $\mathcal{C}(\bar{x}) = \mathcal{C}(\hat{x}) \setminus \{C_r, C_s\}$. Otherwise, $s = z$ and we only unshrink C_r . Then, there are two cases. In the first case, we have $vz \in E(P')$ for some $v \in V$. Observe that $\hat{x}(\delta(v)) = 1$ but $\bar{x}(\delta(v)) = 0$. Hence we replace the edge vz with edges $vv', v'z$. In the second case, we have $v'z \in E(P')$ for some $v \in V$. This implies $\hat{x}(\delta(v)) = 0$ but $\bar{x}(\delta(v)) = 1$. So we replace edges $vv', v'z$ with the edge vz .

Theorem 3.8. *Algorithm 1 computes a basic maximum-weight fractional matching with $\gamma(G)$ odd cycles in polynomial time.*

Proof. There are at most $O(n)$ vertex-disjoint odd cycles in $\mathcal{C}(\hat{x})$. At every iteration, we

eliminate at least one odd cycle from $\mathcal{C}(\hat{x})$ or a frustrated tree from G' . Hence, there are at most $O(n)$ iterations, and Algorithm 1 terminates in polynomial time. Next, we prove correctness. Suppose we obtain an M' -frustrated tree T . Every edge in T has one endpoint in $A(T)$ and another endpoint in $B(T)$. Every edge in $\delta_{G'}(T)$ has one endpoint in $A(T)$ and another endpoint outside T . Since the matching in T remains unchanged in every iteration, this property continues to hold throughout the execution of the algorithm. Thus, T is a frustrated tree in every subsequent iteration. This implies that the last matching generated by the algorithm is maximum. By Lemma 3.7, we have $|\mathcal{C}(\hat{x})| = \gamma(G)$. \square

We remark here that in Algorithm 1, we can avoid solving linear programs to obtain \hat{x} and y in Steps 1 and 2. They can be computed using a simple duplication technique by Nemhauser and Trotter [19]. Create an auxiliary bipartite graph, and run the Hungarian algorithm to get a maximum-weight matching and a minimum w -vertex cover. Then, they can be easily mapped back into a basic maximum-weight fractional matching and a minimum fractional w -vertex cover of our original graph G . We refer to [19] for details.

Chapter 4

Computing vertex-stabilizers

The goal of this chapter is to prove the following theorem:

Theorem 4.1. *There exists a polynomial-time algorithm that computes a minimum vertex-stabilizer S for a weighted graph G . Moreover, $\nu(G \setminus S) \geq \frac{2}{3}\nu(G)$.*

Let us start with discussing a lower bound on the size of a minimum vertex-stabilizer.

Lower bound. We will here prove that $\gamma(G)$ is a lower bound on the number of vertices to remove in order to stabilize a graph. We need a couple of lemmas.

Lemma 4.2. *Let \hat{x} be a basic maximum-weight fractional matching and y be a minimum fractional w -vertex cover in G . Let $s \in V(C)$ for some odd cycle $C \in \mathcal{C}(\hat{x})$. If \bar{x} is the fractional matching obtained by alternate rounding on C at s , then $\bar{x}_{-\delta(s)}$ is a basic maximum-weight fractional matching and y_{-s} is a minimum fractional w -vertex cover in $G \setminus s$.*

Proof. First, notice that $\bar{x}_{-\delta(s)}$ is a basic fractional matching and y_{-s} is a fractional w -vertex cover in $G \setminus s$. We will show that they satisfy complementary slackness. Let $uv \in E(C)$ be an edge where $\bar{x}_{uv} > 0$. Since $e \in E(C)$, we have $\hat{x}_{uv} > 0$ and so $y_u + y_v = w_{uv}$. Next, let $v \neq s$ be a vertex in C where $y_v > 0$. We only need to check the vertices in C because $\hat{x}_e = 0$ for every edge $e \in \delta(s) \setminus E(C)$. Since v is $M(\hat{x})$ -covered, we have $\bar{x}(\delta(v)) = 1$. Therefore, $\bar{x}_{-\delta(s)}$ and y_{-s} form a primal-dual optimal pair. \square

The following operation allows us to switch between fractional matchings on a set of edges:

Definition 4.3. Let x and x' be fractional matchings in G . By *switching* on $E' \subseteq E$ from x to x' , we mean replacing x_e by x'_e for all $e \in E'$.

Switching does not necessarily yield a feasible fractional matching. Hence, we will only use it on the components of a specific subgraph of G :

Claim 4.4. *Given two basic fractional matchings x and x' , let H be the subgraph of G induced by $\text{supp}(x + x')$. For any component K in H , switching on $E(K)$ from x to x' yields a basic fractional matching in G .*

Proof. Let \bar{x} denote the vector obtained by switching on $E(K)$ from x to x' . We first show that \bar{x} is a feasible fractional matching in G . For the purpose of contradiction, suppose there exists a vertex $v \in V(K)$ such that $\bar{x}(\delta(v)) > 1$. Since $\bar{x}_e = x'_e$ for all $e \in E(K)$ and $\bar{x}_e = x_e$ for all $e \notin E(K)$, we have $0 < \bar{x}(\delta(v) \cap E(K)) \leq 1$ and $0 < \bar{x}(\delta(v) \setminus E(K)) \leq 1$. So there exists an edge $f \in \delta(v) \setminus E(K)$ such that $x_f > 0$, which is a contradiction. It is easy to see that \bar{x} is basic. \square

Lemma 4.5. *For every vertex $v \in V$, $\gamma(G \setminus v) \geq \gamma(G) - 1$.*

Proof. Let x^* be a basic maximum-weight fractional matching in G such that $|\mathcal{C}(x^*)| = \gamma(G)$. Let y be a minimum fractional w -vertex cover in G . For the purpose of contradiction, suppose there exists a vertex $u \in V$ such that $\gamma(G \setminus u) < \gamma(G) - 1$. There are two cases:

Case 1: $u \in V(C)$ for some odd cycle $C \in \mathcal{C}(x^*)$. Let \bar{x} be the fractional matching obtained from x^* by alternate rounding on C at u . By Lemma 4.2, we know that $\bar{x}_{-\delta(u)}$ is a basic maximum-weight fractional matching and y_{-u} is a minimum fractional w -vertex cover in $G \setminus u$. We first give a proof sketch for this case. If $\bar{x}_{-\delta(u)}$ is not an optimal basic solution yielding $\gamma(G \setminus u)$ odd cycles, then one of the structures given by Theorem 3.4 must exist. This same structure would be a structure corresponding to the basic solution x^* , but this yields a contradiction since x^* is an optimal basic solution with $\gamma(G)$ odd cycles.

Notice that $\mathcal{C}(\bar{x}_{-\delta(u)}) = \mathcal{C}(\bar{x})$ and $M(\bar{x}_{-\delta(u)}) = M(\bar{x})$. Since $|\mathcal{C}(\bar{x})| = |\mathcal{C}(x^*)| - 1 = \gamma(G) - 1 > \gamma(G \setminus u)$, Theorem 3.4 tells us that $G \setminus u$ contains one of the following structures. The first structure is a vertex $v \in V(C_i)$ for some odd cycle $C_i \in \mathcal{C}(\bar{x})$ such that $y_v = 0$. However, since $C_i \in \mathcal{C}(x^*)$, by Theorem 3.4 we arrive at the contradiction $|\mathcal{C}(x^*)| > \gamma(G)$. The second structure is a tight and valid $M(\bar{x})$ -alternating path P which connects two odd cycles $C_i, C_j \in \mathcal{C}(\bar{x})$, or an odd cycle $C_i \in \mathcal{C}(\bar{x})$ and a vertex $v \notin V(\mathcal{C}(\bar{x}))$ such that $y_v = 0$. Note that $C_i, C_j \in \mathcal{C}(x^*)$. If $V(P) \cap V(C) = \emptyset$, then P is also a tight and valid $M(x^*)$ -alternating path in G which connects C_i and C_j , or C_i and v . Otherwise, let

$s = V(C_i) \cap V(P)$ and let t denote the first vertex of C encountered while traversing along P from s . Then, the s - t subpath of P is a tight $M(x^*)$ -alternating path which connects $C_i, C \in \mathcal{C}(x^*)$. We again obtain the contradiction $|\mathcal{C}(x^*)| > \gamma(G)$ by Theorem 3.4.

Case 2: $u \notin V(\mathcal{C}(x^))$.* If u is $M(x^*)$ -exposed, then $\nu_f(G \setminus u) = \nu_f(G)$ and $\gamma(G \setminus u) = \gamma(G)$. So we may assume u is $M(x^*)$ -covered. Let \hat{x} be a basic maximum-weight fractional matching in $G \setminus u$ such that $|\mathcal{C}(\hat{x})| < \gamma(G) - 1$. Define $N(\hat{x}) := M(\hat{x}) \setminus E(\mathcal{C}(x^*))$ and $N(x^*) := M(x^*) \setminus E(\mathcal{C}(\hat{x}))$. Consider the subgraph $J = (V, N(x^*) \Delta N(\hat{x}))$. Note that u is covered by $N(x^*)$ and exposed by $N(\hat{x})$. Let P be the component in J which contains u . We know that P is a path with u as an endpoint. Let v be the other endpoint of P . There are 3 subcases. We first give an overview of how we arrive at a contradiction in each subcase. We show that one can move from x^* to a new solution \tilde{x} such that:

- (i) \tilde{x} is a basic maximum-weight fractional matching for a subgraph G' obtained by deleting at most 1 vertex from a cycle of $\mathcal{C}(x^*)$; and
- (ii) $|\mathcal{C}(\tilde{x})| < \gamma(G')$.

Clearly, both of the above properties cannot hold, so this yields a contradiction.

Subcase 2.1: $v \in C$ for some odd cycle $C \in \mathcal{C}(x^)$.* In this subcase, the path P has even length. Let \bar{x} be the fractional matching obtained from x^* by alternate rounding on C at v . By Lemma 4.2, $\bar{x}_{-\delta(v)}$ is a basic maximum-weight fractional matching in $G \setminus v$. Let H be the subgraph of G induced by $\text{supp}(\hat{x} + \bar{x})$ (see Figure 4.1 for an example). Note that $\hat{x}_e + \bar{x}_e = 0$ for every edge $e \notin E(P)$ which is incident to a vertex in P . Thus, P is a component in H . Since $|\mathcal{C}(\bar{x})| = \gamma(G) - 1 > |\mathcal{C}(\hat{x})|$, there exists a component K in H which has more odd cycles from $\mathcal{C}(\bar{x})$ than $\mathcal{C}(\hat{x})$. Switching on K from $\bar{x}_{-\delta(v)}$ to \hat{x} yields a basic fractional matching in $G \setminus v$ with less than $\gamma(G) - 1$ odd cycles. To yield a contradiction to Case 1, it is left to show that it is maximum-weight. This is because we are deleting a vertex v from an odd cycle of $\mathcal{C}(x^*)$, but $\gamma(G \setminus v)$ decreases by more than 1. Now, since \hat{x} and $\bar{x}_{-\delta(v)}$ are maximum-weight fractional matchings in $G \setminus u$ and $G \setminus v$ respectively, we have $\sum_{e \in E(K)} w_e \hat{x}_e = \sum_{e \in E(K)} w_e \bar{x}_e$ because $u, v \notin V(K)$. Thus, the resulting matching is indeed maximum-weight in $G \setminus v$.

Subcase 2.2: $v \in C$ for some odd cycle $C \in \mathcal{C}(\hat{x})$. In this subcase, the path P has odd length. Let \bar{x} be the fractional matching obtained from \hat{x} by alternate rounding on C at v . By Lemma 4.2, $\bar{x}_{-\delta(v)}$ is a basic maximum-weight fractional matching in $G \setminus \{u, v\}$. Let H be the subgraph of G induced by $\text{supp}(x^* + \bar{x})$ (see Figure 4.2 for an example). Note that $x^*_e + \bar{x}_e = 0$ for every edge $e \notin E(P)$ incident to a vertex in P . Thus, P is a component in H . Since $|\mathcal{C}(\bar{x})| = |\mathcal{C}(\hat{x})| - 1 < \gamma(G) - 2 < |\mathcal{C}(x^*)|$, there exists a component K in H which has more odd cycles from $\mathcal{C}(x^*)$ than $\mathcal{C}(\bar{x})$. Switching on K from x^* to \bar{x} yields a basic

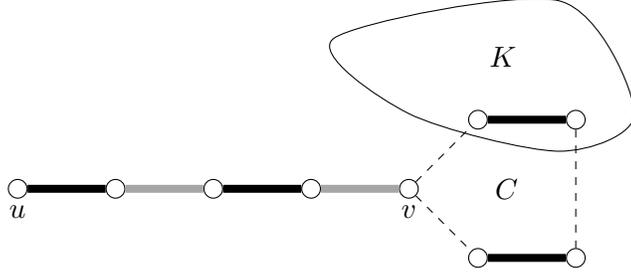


Figure 4.1: An example of the graph induced by $\text{supp}(\hat{x} + \bar{x})$ in Subcase 2.1. Black bold edges are in $M(\bar{x})$ while gray bold edges are in $M(\hat{x})$.

fractional matching in G with less than $\gamma(G)$ odd cycles. To yield a contradiction, it is left to show that it is maximum-weight. Since x^* and $\bar{x}_{-\delta(v)}$ are maximum-weight fractional matchings in G and $G \setminus \{u, v\}$ respectively, we have $\sum_{e \in E(K)} w_e x_e^* = \sum_{e \in E(K)} w_e \bar{x}_e$ because $u, v \notin V(K)$. Thus, the resulting basic fractional matching is maximum-weight in G .

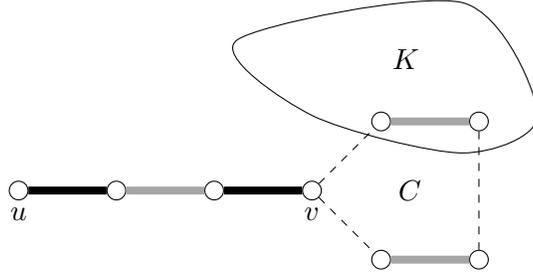


Figure 4.2: An example of the graph induced by $\text{supp}(x^* + \bar{x})$ in Subcase 2.2. Black bold edges are in $M(x^*)$ while gray bold edges are in $M(\bar{x})$.

Subcase 2.3: $v \notin V(\mathcal{C}(x^*) \cup \mathcal{C}(\hat{x}))$. Let H be the subgraph of G induced by $\text{supp}(x^* + \hat{x})$ (see Figure 4.3 for an example). Note that $x_e^* + \hat{x}_e = 0$ for every edge $e \notin E(P)$ which is incident to a vertex in P . Thus, the path P is a component in H . Since $|\mathcal{C}(x^*)| > \gamma(G) - 1 > |\mathcal{C}(\hat{x})|$, there exists a component K in H which has more odd cycles from $\mathcal{C}(x^*)$ than $\mathcal{C}(\hat{x})$. Switching on K from x^* to \hat{x} yields a basic fractional matching in G with less than $\gamma(G)$ odd cycles. To yield a contradiction, it is left to show that it is maximum-weight. Since x^* and \hat{x} are maximum-weight fractional matchings in G and $G \setminus u$ respectively, we have $\sum_{e \in E(K)} w_e x_e^* = \sum_{e \in E(K)} w_e \hat{x}_e$ because $u \notin V(K)$. This implies that the resulting basic fractional matching is indeed maximum-weight in G . \square

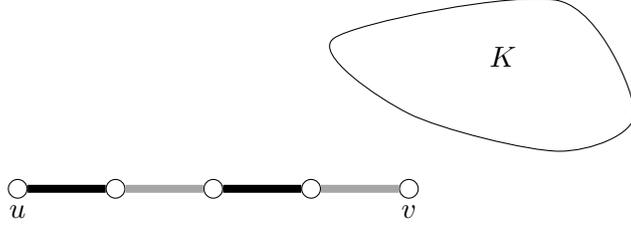


Figure 4.3: An example of the graph induced by $\text{supp}(x^* + \hat{x})$ in Subcase 2.3. Black bold edges are in $M(x^*)$ while gray bold edges are in $M(\hat{x})$.

As a corollary to the above lemma, we obtain the claimed lower bound.

Lemma 4.6. *For every vertex-stabilizer S of G , $|S| \geq \gamma(G)$.*

The algorithm. The algorithm we use to stabilize a graph is very simple: it computes a basic maximum-weight fractional matching \hat{x} in G with $\gamma(G)$ odd cycles (this can be done using Algorithm 1) and a minimum fractional w -vertex cover y in G , and then removes *one* vertex from every cycle in $\mathcal{C}(\hat{x})$, namely, the vertex with the least y -value in the cycle. Algorithm 2 formalizes this.

Algorithm 2: Minimum vertex-stabilizer

- 1 Initialize $S \leftarrow \emptyset$
 - 2 Compute a minimum fractional w -vertex cover y in G
 - 3 Compute a basic maximum-weight fractional matching \hat{x} in G with $\gamma(G)$ odd cycles
 - 4 Let $\mathcal{C}(\hat{x}) = \{C_1, C_2, \dots, C_{\gamma(G)}\}$
 - 5 **for** $i = 1$ **to** $\gamma(G)$ **do**
 - 6 Let $v_i = \arg \min_{v \in V(C_i)} y_v$
 - 7 $S \leftarrow S + v_i$
 - 8 **return** S
-

We are now ready to prove the main theorem stated at the beginning of the section, Theorem 4.1.

Proof of Theorem 4.1. Let $S = \{v_1, v_2, \dots, v_{\gamma(G)}\}$ be the set of vertices returned by the algorithm. Let \bar{x} be the vector obtained from \hat{x} by alternate rounding on C_i at v_i for all i respectively. By Lemma 4.2, $\bar{x}_{-\cup_{i=1}^{\gamma(G)} \delta(v_i)}$ is a basic maximum-weight fractional matching in $G \setminus S$. Note that it is also a maximum-weight integral matching in $G \setminus S$. Thus,

$\nu(G \setminus S) = \nu_f(G \setminus S)$ and $G \setminus S$ is stable. Moreover, S is minimum by Lemma 4.6. It is left to show that $\nu(G \setminus S) \geq \frac{2}{3}\nu(G)$. For every odd cycle $C_i \in \mathcal{C}(\hat{x})$, we have

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

because v_i has the smallest fractional w -vertex cover in C_i . From Lemma 4.2, we also know that y_{-S} is a minimum fractional w -vertex cover in $G \setminus S$. Then,

$$\nu(G \setminus S) = \tau_f(G \setminus S) = \mathbb{1}^\top y - \sum_{i=1}^{\gamma(G)} y_{v_i} \geq \mathbb{1}^\top y - \frac{1}{3} \sum_{i=1}^{\gamma(G)} y(C_i) \geq \mathbb{1}^\top y - \frac{1}{3} \mathbb{1}^\top y = \frac{2}{3} \tau_f(G) \geq \frac{2}{3} \nu(G)$$

□

Note that removing any single vertex from each cycle of $\mathcal{C}(\hat{x})$ yields a minimum-cardinality vertex stabilizer. The reason we chose the vertex with the smallest y_v is to preserve the value of the original maximum-weight matching by a factor of $\frac{2}{3}$.

Tightness of the matching bound. A natural question is whether it is possible to design an algorithm that always returns a vertex-stabilizer S satisfying $\nu(G \setminus S) \geq \alpha \nu(G)$, for some $\alpha > \frac{2}{3}$. We report here an example showing that, in general, this is not possible since the bound of $\frac{2}{3}$ can be asymptotically tight. Consider the graph G in Figure 4.4 for some sufficiently small $\varepsilon > 0$. It is unstable because it is an augmenting flower. The maximum-weight matching is given by the bold edges. For any vertex stabilizer S ,

$$\nu(G \setminus S) \leq 2 = \frac{2}{3 - \varepsilon} (3 - \varepsilon) = \frac{2}{3 - \varepsilon} \nu(G)$$

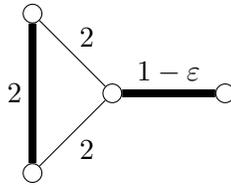


Figure 4.4: An example showing that the bound of $\frac{2}{3}$ can be asymptotically tight.

Another natural question is whether one can at least distinguish if, for a specific instance, there exists a vertex-stabilizer S such that $\nu(G \setminus S) = \nu(G)$. Once again, we show

that the answer is negative. Specifically, let us call a vertex-stabilizer S *weight-preserving* if $\nu(G \setminus S) = \nu(G)$. We show that finding such a vertex-stabilizer is hard in general. The proof is based on a reduction from the independent set problem, similar to the one given by Biró et al. [6].

Theorem 4.7. *Deciding whether a graph has a weight-preserving vertex-stabilizer is NP-complete.*

Proof. The problem is clearly in NP because any yes-instance can be verified using a weight-preserving vertex-stabilizer in polynomial time. To prove NP-hardness, we give a reduction from the independent set problem. Let $G = (V, E)$ and k be an independent set instance, where $V = \{v_1, v_2, \dots, v_n\}$. The independent set problem asks to determine whether G has an independent set of size at least k . We may assume $2 \leq k \leq n$. We construct the gadget graph G^* as follows. First, set the weight on every edge in E to 1. For each $v_i \in V$, add a vertex v'_i and the edge $v_i v'_i$ with weight 1. Denote this set of new vertices as $V' = \{v'_1, v'_2, \dots, v'_n\}$. Next, create k pairwise-disjoint copies of the three cycle $C_i = (V_i, E_i)$ where $V_i = \{a_i, b_i, c_i\}$, $E_i = \{a_i b_i, b_i c_i, a_i c_i\}$ and the weight on every edge in E_i is 4. Finally, add the edge $b_i v_j$ for every $i \in [k]$ and $j \in [n]$ with weight 2. (See Figure 4.5)

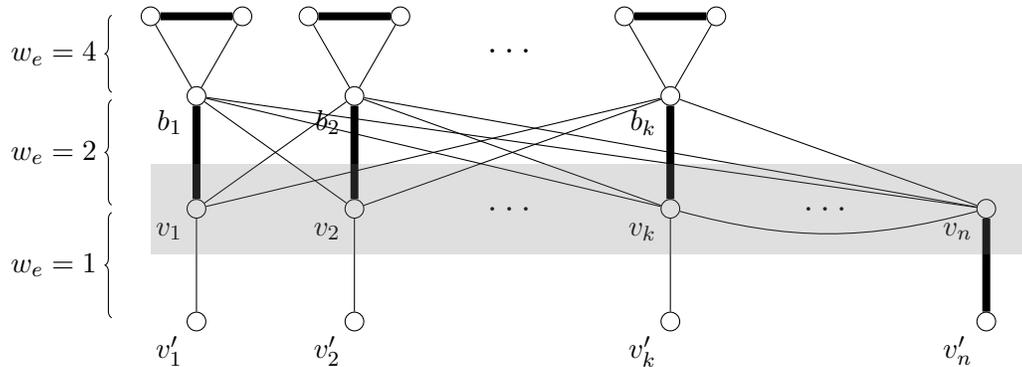


Figure 4.5: The gadget graph G^* .

Our goal is to show that G has an independent set of size at least k if and only if G^* has a weight-preserving vertex-stabilizer. Before proving the main result, we first derive some properties of maximum-weight matchings in G^* .

Claim 4.8. *If M is a maximum-weight matching in G^* , then $M \cap E = \emptyset$.*

Proof. For the purpose of contradiction, suppose there exists an edge $v_i v_j \in M \cap E$. Then, (v'_i, v_i, v_j, v'_j) forms an augmenting path, which is a contradiction. \square

Claim 4.9. *If M is a maximum-weight matching in G^* , every b_i is matched to some v_j in M .*

Proof. For the purpose of contradiction, suppose there exists an $i \in [k]$ such that $b_i v_j \notin M$ for all $j \in [n]$. Then, b_i is either M -exposed or matched to a_i or c_i . Since $k \leq n$, by the pigeonhole principle there exists an $\ell \in [n]$ such that $b_j v_\ell \notin M$ for all $j \in [k]$. By Claim 4.8, $v_\ell v'_\ell \in M$. If b_i is M -exposed, then (b_i, v_ℓ, v'_ℓ) forms an augmenting path. Otherwise, we may without loss of generality assume $a_i b_i \in M$. Then, $(c_i, a_i, b_i, v_\ell, v'_\ell)$ forms an augmenting path. We have reached a contradiction. \square

Claim 4.10. $\nu(G^*) = n + 5k$

Proof. Let M be a maximum-weight matching in G^* . By Claim 4.9, there are k edges of the form $b_i v_j$ in M . Hence, there are also k edges of the form $a_i c_i$ in M . Moreover, we have $n - k$ edges of the form $v_i v'_i$ in M . This gives a total weight of $2k + 4k + n - k = n + 5k$. \square

Claim 4.11. *The set of inessential vertices in G^* is V' .*

Proof. It is easy to see that the vertices in G and $\cup_{i=1}^k C_i$ are essential because they are covered by every maximum-weight matching in G^* . Let $v'_i \in V'$ and M be a maximum-weight matching in G^* such that $v_i v'_i \in M$. Since $k \geq 1$, there exist $j \in [k]$ and $\ell \in [n]$ such that $b_j v_\ell \in M$. Define a new matching $M' := M + b_j v_i - b_j v_\ell + v_\ell v'_\ell - v_i v'_i$. Note that M' is a maximum-weight matching in G^* and v'_i is M' -exposed. Thus, v'_i is inessential. \square

For the forward direction, let S be an independent set of G where $|S| = k$. Without loss of generality, assume $S = \{v_1, v_2, \dots, v_k\}$. Let M be the matching defined by

$$M := \{a_i c_i, b_i v_i : 1 \leq i \leq k\} \cup \{v_i v'_i : k < i \leq n\}.$$

Since $w(M) = n + 5k$, by Claim 4.10 it is a maximum-weight matching in G^* . We claim that $S' := \{v'_1, v'_2, \dots, v'_k\}$ is a weight-preserving vertex-stabilizer of G^* . First, note that the matching M survives after removing S' from G^* . Hence, M is a maximum-weight matching in $G^* \setminus S'$. It is left to show that $G^* \setminus S'$ is stable. We define a fractional w -vertex cover y on $G^* \setminus S'$ as follows:

$$y_v = \begin{cases} 2, & \text{if } v \in \cup_{i=1}^k \{a_i, b_i, c_i\} \\ 1, & \text{if } v \in \cup_{i=k+1}^n v_i \\ 0, & \text{otherwise.} \end{cases}$$

It is easy to check that for every $uv \notin E$, the condition $y_u + y_v \geq w_{uv}$ holds. Let $v_i v_j \in E$. Since S is an independent set, at most one of v_i and v_j is in S . This implies that $y_{v_i} + y_{v_j} \geq 1$. So y is indeed a fractional w -vertex cover. Since $\tau_f(G^* \setminus S') = n + 5k = \nu(G^* \setminus S')$, $G^* \setminus S'$ is stable.

For the converse, let S' be a weight-preserving vertex-stabilizer of G . We know that $S' \subseteq V'$ by Claim 4.11 because S' does not contain essential vertices. Let M be a maximum-weight matching in $G^* \setminus S'$. Since $\nu(G^* \setminus S') = \nu(G^*)$, M is also a maximum-weight matching in G^* . We claim that $|S'| = k$. If $|S'| < k$, then there exists an $i \in [n]$ such that $v_i v'_i \in G^* \setminus S'$ and $b_j v_i \in M$ for some $j \in [k]$. Then, $a_j c_j \in M$, so $C_j \cup (b_j, v_i, v'_i)$ forms an augmenting flower. This is a contradiction because $G^* \setminus S'$ is stable. On the other hand, if $|S'| > k$, then $w(M) = 6k + n - |S'| < n + 5k$. This is also a contradiction because $\nu(G^* \setminus S') = \nu(G^*)$. It follows that $|S'| = k$.

Without loss of generality, assume $S' = \{v'_1, v'_2, \dots, v'_k\}$. Let $S := \{v_1, v_2, \dots, v_k\}$. We claim that every $v_i \in S$ is matched to some b_j in M . For the purpose of contradiction, suppose there exists $v_i \in S$ such that v_i is M -exposed. By the pigeonhole principle, there exists $j \in [n]$ such that $v_j v'_j \in G^* \setminus S'$ and $b_\ell v_j \in M$ for some $\ell \in [k]$. Then, (v_i, b_ℓ, v_j, v'_j) forms an augmenting path, which is a contradiction. It is left to show that S is an independent set. For the purpose of contradiction, suppose there exist $v_i, v_j \in S$ such that $v_i v_j \in E$. Let $b_p v_i, b_q v_j \in M$ for some $p, q \in [k]$. Then, $C_p \cup (b_p, v_i, v_j, b_q) \cup C_q$ forms an augmenting bi-cycle. This implies that $G^* \setminus S'$ is unstable, which is a contradiction. Thus, S is an independent set and $|S| \geq k$. \square

Chapter 5

Computing edge-stabilizers

In contrast to the vertex-stabilizer problem, finding a minimum edge-stabilizer is computationally difficult. Biró et al. [6] proved that the problem is NP-hard on weighted graphs. We strengthen this result by showing the following hardness of approximation:

Theorem 5.1. *There is no constant factor approximation for the minimum edge-stabilizer problem unless $P = NP$.*

Proof. We give a gap-producing reduction from the independent set problem. Let $G = (V, E)$ and k be an independent set instance where $V = \{v_1, v_2, \dots, v_n\}$. The independent set problem asks to determine whether G has an independent set of size at least k . We may assume $2 \leq k \leq n$. Let $\rho \geq 1$ be an integer. We construct the gadget graph $G^* = (V^*, E^*)$ as follows. For every edge $v_i v_j \in E$, replace it with ρk paths of length 3, i.e. $(v_i, u_{ij}^\ell, u_{ji}^\ell, v_j)$ for $\ell \in [\rho k]$. Assign weight 1 to every edge in the paths. For each $v_i \in V$, add a vertex v'_i and the edge $v_i v'_i$ with weight 1. Next, create k pairwise-disjoint copies of the complete graph on $2\rho k + 1$ vertices. Denote each of them as H_i and assign weight 4 to every edge in H_i . In addition, for every H_i , label one of the vertices as b_i . Finally, add the edge $b_i v_j$ for every $i \in [k]$ and $j \in [n]$ with weight 2. (See Figure 5.1)

Claim 5.2. *If G has an independent set of size at least k , then G^* has an edge-stabilizer of size at most k .*

Proof. Let S be an independent set in G where $|S| = k$. Without loss of generality, we may assume $S = \{v_1, v_2, \dots, v_k\}$. Let $F = \cup_{i=1}^k v_i v'_i$. We claim that F is an edge-stabilizer of G^* .

Proof. We prove the contrapositive. Assume G^* has an edge-stabilizer F such that $|F| < (\rho + 1)k$. Let M be a maximum-weight matching in $G^* \setminus F$. Let c denote the number of edges removed from the complete graphs, i.e. $c := |F \cap \cup_{i=1}^k E(H_i)|$. We first show that $c < 2\rho k - 1$. According to Ore's Theorem, we need to remove at least $2\rho k - 1$ edges from H_i in order to make it non-Hamiltonian. Let $\mathcal{H} := \{i : H_i \setminus F \text{ is Hamiltonian}\}$. Then, $|\mathcal{H}| \geq k - \frac{c}{2\rho k - 1}$. For every $i \in \mathcal{H}$, $b_i v_j \in M$ for some $j \in [n]$, otherwise $H_i \setminus F$ contains an augmenting blossom because it has an odd number of vertices. Thus, $v_j v'_j \subseteq F$, otherwise $H_i \setminus F \cup (b_i, v_j, v'_j)$ contains an augmenting flower. We have

$$\begin{aligned}
c + |\mathcal{H}| &< (\rho + 1)k \\
c + \left(k - \frac{c}{2\rho k - 1}\right) &< (\rho + 1)k \\
c \left(1 - \frac{1}{2\rho k - 1}\right) &< \rho k \\
c \left(\frac{2\rho k - 2}{2\rho k - 1}\right) &< \rho k \\
c &< (2\rho k - 1) \left(\frac{\rho k}{2\rho k - 2}\right) \\
c &< 2\rho k - 1
\end{aligned}$$

Since $c < 2\rho k - 1$, $|\mathcal{H}| = k$. Without loss of generality, we may assume $b_i v_i \in M$ for every $i \in [k]$. Then, $\cup_{i=1}^k v_i v'_i \subseteq F$. We claim that $S = \{v_1, v_2, \dots, v_k\}$ is an independent set in G . For the purpose of contradiction, suppose there exists an edge $v_i v_j \in E$ for some $i, j \in [k]$. Let $\mathcal{P}_{ij} = \cup_{\ell=1}^{\rho k} (v_i, u_{ij}^\ell, u_{ji}^\ell, v_j)$ denote the set of paths between v_i and v_j . Since $|F \cap \mathcal{P}_{ij}| < (\rho + 1)k - k = \rho k$, at least one path $(v_i, u_{ij}^t, u_{ji}^t, v_j) \in \mathcal{P}_{ij}$ is present in $G^* \setminus F$. Observe that $u_{ij}^t, u_{ji}^t \in M$, and $(H_i \setminus F) \cup (b_i, v_i, u_{ij}^t, u_{ji}^t, v_j, b_j) \cup (H_j \setminus F)$ contains an augmenting bi-cycle. Thus, $G^* \setminus F$ is unstable, which is a contradiction. \square

Now, suppose we have an α -approximation to the minimum edge-stabilizer problem for some constant $\alpha \geq 1$. Set $\rho = \lceil \alpha \rceil$ and construct the gadget graph G^* as shown above. Run this algorithm on G^* and let F be the returned edge-stabilizer. Let OPT be size of a minimum edge-stabilizer in G^* . If G has an independent set of size at least k , then by Claim 5.2 we have $\text{OPT} \leq k$ and $|F| \leq \alpha \cdot \text{OPT} \leq \rho k$. On the other hand, if G does not have an independent set of size at least k , then by Claim 5.3 we have $\text{OPT} \geq (\rho + 1)k > \rho k$. This implies that $|F| > \rho k$. Therefore, we can use this algorithm to decide the independent set problem in polynomial time. \square

In this section, we prove that Algorithm 2 is a $O(\Delta)$ -approximation algorithm for the minimum edge-stabilizer problem. We first need to establish a lower bound on the optimal solution. Next example shows that, differently from the unweighted case, $\gamma(G)$ is not a lower bound on the size of a minimum edge-stabilizer for weighted graphs. Let G be the graph depicted in Figure 5.2, whose maximum-weight matching is given by the bold edges. It is unstable because it is an augmenting bi-cycle. Even though $\gamma(G) = 2$, the edge with weight 0.5 is a minimum edge-stabilizer.

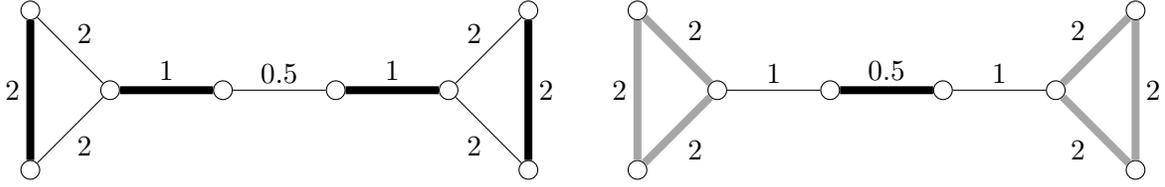


Figure 5.2: An example showing that $\gamma(G)$ is not a lower bound. The unique maximum-weight matching is shown in the left, while the unique maximum-weight fractional matching is shown in the right. Gray edges have value $\frac{1}{2}$.

However, we can prove the following.

Lemma 5.4. *For every edge $e \in E$, $\gamma(G \setminus e) \geq \gamma(G) - 2$.*

Proof. Let x^* be a basic maximum-weight fractional matching in G such that $|\mathcal{C}(x^*)| = \gamma(G)$. Let y be a minimum fractional w -vertex cover in G . Pick an edge $ab \in E$. If $x_{ab}^* = 0$, then $\nu_f(G \setminus ab) = \nu_f(G)$ and $\gamma(G \setminus ab) = \gamma(G)$. So we may assume $x_{ab}^* \in \{\frac{1}{2}, 1\}$. Let G' be the graph obtained by replacing the edge ab with edges $ab', b'a', a'b$ where $w_{ab'} = w_{b'a'} = w_{a'b} = w_{ab}$ and $a', b' \notin V$. Define the vectors \hat{x} and \hat{y} as

$$\hat{x}_e = \begin{cases} 1 - x_{ab}^*, & \text{if } e = b'a', \\ x_{ab}^*, & \text{if } e \in \{ab', a'b\}, \\ x_e^*, & \text{otherwise.} \end{cases} \quad \hat{y}_v = \begin{cases} y_a, & \text{if } v = a', \\ y_b, & \text{if } v = b', \\ y_v, & \text{otherwise.} \end{cases}$$

Note that \hat{x} is a basic fractional matching while \hat{y} is a fractional w -vertex cover in G' . Furthermore, they satisfy complementary slackness conditions as $\hat{y}_u + \hat{y}_v = w_{uv}$ for all $uv \in \{ab', b'a', a'b\}$ and $\hat{x}(\delta(v)) = 1$ for all $v \in \{a, a', b, b'\}$. Hence, they form a primal-dual optimal pair. Since $|\mathcal{C}(\hat{x})| = \gamma(G)$, we have $\gamma(G') \leq \gamma(G)$. We claim that $\gamma(G') = \gamma(G)$. For the purpose of contradiction, suppose $\gamma(G') < \gamma(G)$. By Theorem 3.4, G' contains one of the following:

Structure (i): a vertex $v \in V(C_i)$ for some odd cycle $C_i \in \mathcal{C}(\hat{x})$ such that $\hat{y}_v = 0$. If $v \in V$, then $v \in V(\mathcal{C}(x^*))$ and $y_v = 0$. Otherwise, $v \in \{a', b'\}$ which implies $a, b \in V(\mathcal{C}(x^*))$ and $y_a = 0$ or $y_b = 0$. By Theorem 3.4, we arrive at the contradiction $\gamma(G) < |\mathcal{C}(x^*)|$.

Structure (ii): a tight $M(\hat{x})$ -alternating path P connecting two odd cycles $C_i, C_j \in \mathcal{C}(\hat{x})$. If P does not have any intermediate vertices from $\{a, a', b, b'\}$, then it is also a tight $M(x^*)$ -alternating path in G connecting two odd cycles from $\mathcal{C}(x^*)$. Otherwise, $ab', b'a', a'b \in E(P)$ and $C_i, C_j \in \mathcal{C}(x^*)$. Then, $(P \cup ab) \setminus \{ab', b'a', a'b\}$ is a tight $M(x^*)$ -alternating path in G connecting C_i and C_j . By Theorem 3.4, we arrive at the contradiction $\gamma(G) < |\mathcal{C}(x^*)|$.

Structure (iii): a tight and valid $M(\hat{x})$ -alternating path P connecting an odd cycle $C_i \in \mathcal{C}(\hat{x})$ and a vertex $v \notin V(\mathcal{C}(\hat{x}))$ such that $\hat{y}_v = 0$. If P does not have any intermediate vertices from $\{a, a', b, b'\}$, then $v \notin \{a, a', b, b'\}$. Hence, P is also a tight and valid $M(x^*)$ -alternating path in G connecting an odd cycle from $\mathcal{C}(x^*)$ and v . If $ab', b'a', a'b \in E(P)$, then $v \notin \{a', b'\}$ and $C_i \in \mathcal{C}(x^*)$. Thus, $(P \cup ab) \setminus \{ab', b'a', a'b\}$ is a tight and valid $M(x^*)$ -alternating path in G connecting C_i and v . If $a'b \in E(P)$ but $ab', b'a' \notin E(P)$, then $v = a'$ and $y_a = 0$. So $P + ab - a'b$ is a tight and valid $M(x^*)$ -alternating path in G connecting C_i and a . If $ab' \in E(P)$ but $b'a', a'b \notin E(P)$, then $v = b'$ and $y_b = 0$. So $P + ab - ab'$ is a tight and valid $M(x^*)$ -alternating path in G connecting C_i and b . By Theorem 3.4, we arrive at the contradiction $\gamma(G) < |\mathcal{C}(x^*)|$.

Thus, we have shown that $\gamma(G') = \gamma(G)$. Since $G' \setminus \{a', b'\} = G \setminus ab$, applying Lemma 4.5 yields

$$\gamma(G \setminus ab) = \gamma(G' \setminus \{a', b'\}) \geq \gamma(G') - 2 = \gamma(G) - 2.$$

□

As a corollary to the above lemma, we obtain the following lower bound.

Lemma 5.5. *For every edge-stabilizer F of G , $|F| \geq \left\lceil \frac{\gamma(G)}{2} \right\rceil$.*

Since Algorithm 2 deletes $\gamma(G)$ vertices, at most $\gamma(G)\Delta$ edges are deleted, proving the following.

Theorem 5.6. *There exists an efficient $O(\Delta)$ -approximation algorithm for the minimum edge-stabilizer problem.*

Chapter 6

Forcing an outcome

Given a set of deals M , we here look at the problem of removing as few players as possible in order to make M realizable as a stable outcome. This corresponds to finding a minimum vertex-stabilizer S with the additional constraint that M is a maximum-weight matching in $G \setminus S$. Note that this implicitly implies $S \cap V(M) = \emptyset$. A solution to this problem is called an M -vertex-stabilizer. We would like to point out that the following variants of the problem, which are along the lines of Chandrasekaran et al. [9] are polytime solvable: find a weight vector w' such that M is a stable outcome for (G, w') so as to minimize $\|w - w'\|_1$ or $\|w - w'\|_\infty$ (or even the ℓ_p norm $\|w - w'\|_p$). This is an inverse-optimization problem that can be cast as an LP (or convex program for the ℓ_p norm) by exploiting complementary slackness.

Ahmadian et al. [1] previously showed that the M -vertex-stabilizer problem is polynomial-time solvable on unweighted graphs when M is a maximum matching in G . We prove that when M is any arbitrary matching in G , the problem becomes hard:

Theorem 6.1. *The M -vertex-stabilizer problem is NP-hard on unweighted graphs. Furthermore, no efficient $(2 - \varepsilon)$ -approximation algorithm exists for any $\varepsilon > 0$ assuming UGC.*

Proof. We give an approximation-preserving reduction from the vertex cover problem. Let $G = (V, E)$ be a vertex cover instance. For every edge $uv \in E$, replace it with an augmenting path of length three, i.e. (u, u', v', v) where $u'v' \in M$. Denote the resulting (unweighted) graph as $G' = (V', E')$. We will show that every vertex cover in G corresponds to an M -vertex-stabilizer in G' and vice versa. This implies that the reduction is approximation-preserving and the inapproximability results for the vertex cover problem [12, 15] carry over to the problem of finding a minimum M -vertex-stabilizer.

Observe that G' does not contain any alternating cycle or blossom. This implies that there is no augmenting cycle, flower or bi-cycle in G' . Let S be a vertex cover of G . Then, $G' \setminus S$ has no augmenting path because every augmenting path in G' corresponds to an edge in G . Thus, M is a maximum matching in $G' \setminus S$ and hence S is an M -vertex stabilizer. For the converse, suppose S is an M -vertex stabilizer of G' . Note that $S \subseteq V$ as every vertex in $V' \setminus V$ is M -covered. Then, $G \setminus S$ has no edges because every edge in G corresponds to an augmenting path in G' . It follows that S is a vertex cover for G . This concludes the proof of inapproximability. \square

On unweighted graphs, every instance of this problem admits a solution. However, this is not the case for weighted graphs. Consider an M -augmenting bi-cycle. It is unstable by Theorem 2.4, but does not have an M -vertex-stabilizer because every vertex is M -covered. In general, if the graph contains an M -augmenting path whose endpoints are M -covered, or an M -augmenting cycle, or an M -augmenting flower whose root is M -covered, or an M -augmenting bi-cycle, then it does not have an M -vertex-stabilizer. We would like to point out that recognizing an infeasible instance of the M -vertex-stabilizer problem can be done in polynomial time. In particular, we prove that:

Theorem 6.2. *The M -vertex-stabilizer problem admits an efficient 2-approximation algorithm. Furthermore, if M is a maximum-weight matching, then it can be solved in polynomial time.*

We first sketch the main ideas. Given a weighted graph G and a matching M , the algorithm searches for the structures which prevent G from being stable or M from being a maximum-weight matching. Among all such structures, the ones that can be tampered with are augmenting paths with at least one M -exposed endpoint and augmenting flowers with an M -exposed root. The algorithm then proceeds to delete these M -exposed vertices. If there exist augmenting paths whose endpoints are both M -exposed, the problem becomes hard because we do not know which endpoint is optimal to remove. In this case, the algorithm removes both endpoints, thus yielding a 2-approximation. Note that this last case cannot happen if M is maximum-weight, and this explains why the problem becomes polynomial-time solvable. Kleinberg and Tardos [16] were the first to give a method of locating these structures. It involves solving a certain linear program using the dynamic programming algorithm of Aspvall and Shiloach [2]. We use a slightly different algorithm for finding these structures, which in fact, will allow us to prove a strengthened version of Theorem 6.2 (Theorem 6.7). Our algorithm relies on searching for augmenting walks of a certain length, via a slight modification of the dynamic programming algorithm given by Aspvall and Shiloach [2]. Using this as a subroutine, we design an algorithm for the M -vertex-stabilizer problem.

6.1 Finding augmenting walks

The first ingredient is an algorithm to find augmenting walks of a certain length. We say that a valid M -alternating uv -walk P of length at most k is *optimal* if for any other valid M -alternating uv -walk P' of length at most k , we have $w(P \setminus M) - w(P \cap M) \geq w(P' \setminus M) - w(P' \cap M)$. Given a source vertex s and an integer $k \in \mathbb{Z}_+$, the algorithm searches for optimal valid alternating sv -walks of length at most k for all $v \in V$. The significance of optimality is as follows. Let P be an optimal valid alternating sv -walk returned by the algorithm. If P is augmenting, then we have found an augmenting sv -walk of length at most k . Otherwise, we can conclude that there are no augmenting sv -walks of length at most k by the optimality of P .

The inner workings of our algorithm is similar to the Grapevine algorithm given by Aspvall and Shiloach [2]. In their paper, the Grapevine algorithm is used as a subroutine to solve linear systems of the form $Ax \leq b$, where each constraint contains at most 2 variables. They first constructed an auxiliary graph to model the relationship between variables and constraints. The Grapevine algorithm is then run on this auxiliary graph to compute lower and upper bounds of each variable.

We now give an overview of our algorithm. For every vertex $v \in V$, we define two variables $y_1(v)$ and $y_2(v)$. In iteration i , if v is M -exposed, we would like $y_1(v)$ to represent the quantity $w(P \setminus M) - w(P \cap M)$, where P is an optimal valid alternating sv -walk of length at most i . On the other hand, if v is M -covered, we would like $y_2(v)$ to represent the quantity $w(P \setminus M) - w(P \cap M)$, where P is an optimal valid alternating sv -walk of length at most i . We first initialize $y_1(s) = 0$ and $y_2(s) = -\infty$ if s is M -covered, and $y_1(s) = y_2(s) = 0$ if s is M -exposed. For every other vertex $v \neq s$, we set $y_1(v) = y_2(v) = -\infty$. At every iteration, each vertex v determines whether it could increase its $y_1(v)$ value by replacing it with $y_2(u) + w_{uv}$ for some $uv \in E \setminus M$, and similarly, whether it could increase its $y_2(v)$ value by replacing it with $y_1(u) - w_{uv}$ for some $uv \in M$. In a way, this is analogous to the Bellman-Ford algorithm for computing shortest paths. The main difference is that we are maintaining two variables for each vertex, instead of one. This ensures the walk we obtain is M -alternating and valid. Also, notice that we are adding the weights of unmatched edges and subtracting the weights of matched edges. This will

give us our desired quantity $w(P \setminus M) - w(P \cap M)$.

Algorithm 3: Optimal valid M -alternating sv -walks of length at most k

```

1 Initialize vectors  $y_1, y_2, z_1, z_2 \in \mathbb{R}^n$ 
2 if  $s$  is  $M$ -covered then
3    $y_1(s) \leftarrow 0$ 
4    $y_2(s) \leftarrow -\infty$ 
5 else
6    $y_1(s) \leftarrow 0$ 
7    $y_2(s) \leftarrow 0$ 
8 foreach vertex  $v \neq s$  do
9    $y_1(v) \leftarrow -\infty$ 
10   $y_2(v) \leftarrow -\infty$ 
11 for  $i = 1$  to  $k$  do
12   foreach vertex  $v$  do
13      $z_1(v) \leftarrow \max_{u:uv \in E \setminus M} \{y_2(u) + w_{uv}\}$ 
14      $z_2(v) \leftarrow \max_{u:uv \in M} \{y_1(u) - w_{uv}\}$ 
15   foreach vertex  $v$  do
16      $y_1(v) \leftarrow \max(y_1(v), z_1(v))$ 
17      $y_2(v) \leftarrow \max(y_2(v), z_2(v))$ 
18 return  $y_1, y_2$ 

```

In the algorithm, we take the maximum of the empty set to be $-\infty$. For the analysis, we will use $y_1^i(v)$ and $y_2^i(v)$ to denote the value of $y_1(v)$ and $y_2(v)$ respectively at iteration i for all $i < k$. Note that $y_1^0(v)$ and $y_2^0(v)$ refer to the initial value received by $y_1(v)$ and $y_2(v)$ before the main “for” loop. The following lemmas verify our intuition.

Lemma 6.3. *Let v be an M -covered vertex. If there is no valid M -alternating sv -walk of length at most k , then $y_2(v) = -\infty$. Otherwise, there exists an optimal valid M -alternating sv -walk P of length at most k , and $y_2(v) = w(P \setminus M) - w(P \cap M)$.*

Proof. We start by proving the contrapositive of the first statement. Let v be an M -covered vertex where $y_2(v)$ is finite. We proceed by induction on k . We look at two base cases. When $k = 1$, $y_2(v) = y_1^0(s) - w_{sv}$ where $sv \in M$. So (s, v) is our desired walk. When $k = 2$, if $y_2(v)$ was updated in iteration 1, then this reduces to the previous case. Otherwise, $y_2(v) = y_2^0(s) + w_{su} - w_{uv}$ for some $uv \in M$. Since $y_2^0(s)$ is finite, s is M -exposed

and (s, u, v) is our desired walk. For the inductive hypothesis, assume the statement holds for some $k \geq 2$. Consider the case $k + 1$. Let j be the last iteration in which $y_2(v)$ was updated, i.e. $y_2(v) = y_1^{j-1}(u) - w_{uv}$ for some $uv \in M$. We may assume $j > 2$, otherwise we are back at the base cases. Since the update of $y_2(v)$ was triggered by the update of $y_1(u)$, we know that $y_1(u)$ was updated at iteration $j - 1$, i.e. $y_1^{j-1}(u) = y_2^{j-2}(t) + w_{tu}$ for some $tu \in E \setminus M$. Similarly, since the update of $y_1(u)$ was triggered by the update of $y_2(t)$, we know that $y_2(t)$ was updated at iteration $j - 2 > 0$. This implies that t is M -covered. As $y_2^{j-2}(t)$ is finite and $j - 2 < k$, by the inductive hypothesis there exists a valid M -alternating st -walk of length at most $j - 2$. Appending (t, u, v) to this walk yields a valid M -alternating sv -walk of length at most $j \leq k + 1$.

Next, we prove the second statement. Let v be an M -covered vertex where a valid M -alternating sv -walk of length at most k exists. Since the number of such walks is finite, there exists an optimal one. Among all such optimal walks, let P be the shortest one in terms of number of edges. We proceed by induction on k . We look at two base cases. When $k = 1$, $P = (s, v)$ and $y_2(v) = -w_{sv} = w(P \setminus M) - w(P \cap M)$. When $k = 2$, if $|E(P)| = 1$ then this reduces to the previous case. Otherwise, $P = (s, u, v)$ for some $uv \in M$ and $y_2(v) = w_{su} - w_{uv} = w(P \setminus M) - w(P \cap M)$. For the inductive hypothesis, assume the statement holds for some $k \geq 2$. Consider the case $k + 1$. We may assume P has length exactly $k + 1$, otherwise by the inductive hypothesis we are done. Denote $P = (v_0, v_1, \dots, v_{k+1})$ where $v_0 = s$ and $v_{k+1} = v$. Then, $P' = (v_0, v_1, \dots, v_{k-1})$ is an optimal valid M -alternating sv_{k-1} -walk of length at most $k - 1$. Since v_{k-1} is M -covered, by the inductive hypothesis we have $y_2^{k-1}(v_{k-1}) = w(P' \setminus M) - w(P' \cap M)$. We also know that $y_2^{k-1}(v_{k-1}) + w_{v_{k-1}v_k} \geq y_2^{k-1}(u) + w_{uv_k}$ for all $uv_k \in E \setminus M$ because P is optimal. We claim that $y_2^{k-1}(v_{k-1}) + w_{v_{k-1}v_k} > y_1^{k-1}(v_k)$. For the purpose of contradiction, suppose otherwise. Then,

$$\begin{aligned} y_2^k(v) &\geq y_1^{k-1}(v_k) - w_{v_kv} \\ &\geq y_2^{k-1}(v_{k-1}) + w_{v_{k-1}v_k} - w_{v_kv} \\ &= w(P' \setminus M) - w(P' \cap M) + w_{v_{k-1}v_k} - w_{v_kv} \\ &= w(P \setminus M) - w(P \cap M). \end{aligned}$$

Since $y_2^k(v)$ is finite, by the first part of the lemma there exists a valid M -alternating sv -walk of length at most k . So by the inductive hypothesis, $y_2^k(v) = w(Q \setminus M) - w(Q \cap M)$ where Q is an optimal valid M -alternating sv -walk of length at most k . Note that $w(Q \setminus M) - w(Q \cap M) = w(P \setminus M) - w(P \cap M)$ because P is optimal. However, Q is shorter than P , which is a contradiction. Thus, we obtain $y_1^k(v_k) = y_2^{k-1}(v_{k-1}) + w_{v_{k-1}v_k}$ and

$$y_2^k(v) = y_1^k(v_k) - w_{v_kv} = y_2^{k-1}(v_{k-1}) + w_{v_{k-1}v_k} - w_{v_kv} = w(P \setminus M) - w(P \cap M).$$

□

Lemma 6.4. *Let v be an M -exposed vertex. If there is no valid M -alternating sv -walk of length at most k , then $y_1(v) = -\infty$. Otherwise, there exists an optimal valid M -alternating sv -walk P of length at most k , and $y_1(v) = w(P \setminus M) - w(P \cap M)$.*

Proof. We start by proving the contrapositive of the first statement. Let v be an M -exposed vertex where $y_1(v)$ is finite. We proceed by induction on k . We look at two base cases. When $k = 0$, we have $v = s$ and the empty path (s) is our desired walk. When $k = 1$, if $y_1(v)$ was never updated, then $v = s$ and so this reduces to the previous case. Otherwise, $y_1(v) = y_2^0(s) + w_{sv}$. Since $y_2^0(s)$ is finite, s is M -exposed and (s, v) is our desired walk. For the inductive hypothesis, assume the statement holds for some $k \geq 1$. Consider the case $k + 1$. Let j be the last iteration in which $y_1(v)$ was updated, i.e. $y_1(v) = y_2^{j-1}(u) + w_{uv}$ for some $uv \in E \setminus M$. We may assume $j > 1$, otherwise we are back at the base cases. Since the update of $y_1(v)$ was triggered by the update of $y_2(u)$, we know that $y_2(u)$ was updated at iteration $j - 1 > 0$. This implies that u is M -covered. As $y_2^{j-1}(u)$ is finite and $j - 1 \leq k$, by Lemma 6.3 there exists a valid M -alternating su -walk of length at most $j - 1$. Appending (u, v) to this walk yields a valid M -alternating sv -walk of length at most $j \leq k + 1$.

Next, we prove the second statement. Let v be an M -exposed vertex where a valid M -alternating sv -walk of length at most k exists. Since the number of such walks is finite, there exists an optimal one. Among all such optimal walks, let P be the shortest one in terms of number of edges. We proceed by induction on k . We look at two base cases. When $k = 0$, $P = (s)$ and $y_1(v) = 0 = w(P \setminus M) - w(P \cap M)$. When $k = 1$, if $|E(P)| = 0$ then this reduces to the previous case. Otherwise, $P = (s, v)$ and $y_1(v) = w_{sv} = w(P \setminus M) - w(P \cap M)$. For the inductive hypothesis, assume the statement holds for some $k \geq 1$. Consider the case $k + 1$. We may assume P has length exactly $k + 1$, otherwise by the inductive hypothesis we are done. Denote $P = (v_0, v_1, \dots, v_{k+1})$ where $v_0 = s$ and $v_{k+1} = v$. Then, $P' = (v_0, v_1, \dots, v_k)$ is an optimal valid M -alternating sv_k -walk of length at most k . Since v_k is M -covered, by Lemma 6.3 we have $y_2^k(v_k) = w(P' \setminus M) + w(P' \cap M)$. We also know $y_2^k(v_k) + w_{v_kv} \geq y_2^k(u) + w_{uv}$ for all $uv \in E \setminus M$ because P is optimal. We claim that $y_2^k(v_k) + w_{v_kv} > y_1^k(v)$. For the purpose of contradiction, suppose otherwise. Then

$$y_1^k(v) \geq y_2^k(v_k) + w_{v_kv} = w(P' \setminus M) - w(P' \cap M) + w_{v_kv} = w(P \setminus M) - w(P \cap M).$$

Since $y_1^k(v)$ is finite, by the first part of the lemma there exists a valid M -alternating sv -walk of length at most k . So by the inductive hypothesis, $y_1^k(v) = w(Q \setminus M) - w(Q \cap M)$ where Q is an optimal valid M -alternating sv -walk of length at most k . Note that $w(Q \setminus M) - w(Q \cap M) = w(P \setminus M) - w(P \cap M)$ because P is optimal. However, Q is shorter than P , which is a contradiction. Thus, we obtain $y_1(v) = y_2^k(v_k) + w_{v_kv} = w(P \setminus M) - w(P \cap M)$. □

6.2 The algorithm

The reason we look for augmenting walks is because of the following:

Lemma 6.5. *An augmenting uv -walk contains an augmenting uv -path, an augmenting cycle, an augmenting flower rooted at u or v , or an augmenting bi-cycle.*

Proof. We first prove the following claim:

Claim 6.6. *If P is an alternating walk, then it can be decomposed into $P = P_1P_2 \dots P_\ell$ such that:*

- (i) *Every P_i is an alternating path, an alternating cycle or a blossom.*
- (ii) *There is no i such that P_i and P_{i+1} are both alternating paths or blossoms.*

Proof. Let $P = (v_1, v_2, \dots, v_t)$ be an M -alternating walk. We proceed by induction on t . For the base case $t = 2$, P is an alternating path of length 1 as there are no loops in G . Suppose the lemma is true for $t \leq k$ for some $k \geq 2$. Consider the case $t = k + 1$. We may assume P is not simple. Let j be the smallest index such that $v_j = v_i$ for some $i < j$. Decompose P into $P_1 = (v_1, v_2, \dots, v_i)$, $P_2 = (v_i, v_{i+1}, \dots, v_j)$ and $P_3 = (v_j, v_{j+1}, \dots, v_t)$. P_1 is a (possibly empty) alternating path while P_2 is an alternating cycle or a blossom. Since P_3 is an M -alternating walk with fewer edges, by the inductive hypothesis it can be decomposed into $P_3 = P'_1P'_2 \dots P'_\ell$ where every P'_i is an alternating path, an alternating cycle or a blossom. Moreover, there are no consecutive paths or blossoms in this decomposition. Note that P'_1 is not a blossom because P_3 starts with an edge in M . Thus, $P = P_1P_2P'_1P'_2 \dots P'_\ell$ is our desired decomposition. \square

Let P be an M -augmenting uv -walk. Using Claim 6.6, decompose P into $P = P_1P_2 \dots P_k$. If P_i is an augmenting cycle for some $i \in [k]$, then we are done. So we may assume that every alternating cycle in the decomposition is not augmenting. Note that P_k is not an alternating cycle, otherwise P is not valid because it ends with an unmatched edge whose endpoints are M -covered. Let P' be the alternating uv -walk obtained by dropping all the alternating cycles in the decomposition. It is easy to see that P' is still augmenting. Repeat this process until we are left with an augmenting uv -walk $P^* = P'_1P'_2 \dots P'_\ell$ such that every P'_i is an alternating path or a blossom.

If P'_1 is a blossom, then u is M -exposed because the first edge of P^* is not in M . Similarly, if P'_ℓ is a blossom, then v is M -exposed because the last edge of P^* is not in M . In both cases, since P^* does not have any M -exposed intermediate vertices, we get $u = v$ and $\ell = 1$. This implies that P^* is an augmenting blossom with base u , which is trivially

an augmenting flower with root u . Thus, we may assume P_1 and P_ℓ are alternating paths. If $\ell = 1$, then P^* is an augmenting uv -path. Otherwise, from Claim 6.6 we know that P_i is an alternating path for all odd i while P_i is a blossom for all even i . Observe that $P_1^* \cup P_2^*$ and $P_{\ell-1}^* \cup P_\ell^*$ form flowers rooted at u and v respectively, where the former is simple while the latter might not be simple. Moreover, $P_{2i}^* \cup P_{2i+1}^* \cup P_{2i+2}^*$ form bi-cycles for all $i \in [\frac{\ell-3}{2}]$. Since $2w(P^* \setminus M) > 2w(P^* \cap M)$ and

$$2w(P^*) = 2w(P_1^*) + w(P_2^*) + \sum_{i=1}^{(\ell-3)/2} (w(P_{2i}^*) + 2w(P_{2i+1}^*) + w(P_{2i+2}^*)) + w(P_{\ell-1}^*) + 2w(P_\ell^*),$$

at least one of them is augmenting. □

We are now ready to present the algorithm for the M -vertex-stabilizer problem:

Algorithm 4: M -vertex-stabilizer

```

1 Initialize  $S \leftarrow \emptyset$ 
2 foreach  $M$ -exposed vertex  $u$  do
3   Search for  $M$ -augmenting  $uv$ -walks of length at most  $3n$  using Algorithm 3
4   if  $\exists$  an  $M$ -augmenting  $uv$ -walk or  $uv$ -walk for some  $M$ -covered vertex  $v$  then
5      $S \leftarrow S \cup \{u\}$ 
6      $G \leftarrow G \setminus u$ 
7 foreach  $M$ -exposed vertex  $u$  do
8   Search for  $M$ -augmenting  $uv$ -walks of length at most  $n$  using Algorithm 3
9   if  $\exists$  an  $M$ -augmenting  $uv$ -walk for some  $M$ -exposed vertex  $v$  then
10     $S \leftarrow S \cup \{u, v\}$ 
11     $G \leftarrow G \setminus \{u, v\}$ 
12 if  $w(M) < \nu_f(G)$  then
13   return "INFEASIBLE"
14 else
15   return  $S$ 

```

Let S_1 denote the set of M -exposed vertices in G which are roots of augmenting flowers or endpoints of augmenting paths whose other endpoint is M -covered. Note that given a feasible instance, every M -vertex-stabilizer contains S_1 . We prove a stronger statement than Theorem 6.2:

Theorem 6.7. *The M -vertex-stabilizer problem admits an efficient 2-approximation algorithm. Furthermore, if M is a maximum-weight matching in $G \setminus S_1$, then it is polynomial-time solvable.*

Proof. Let G be the input graph and M be a matching in G . Let R be the set of M -exposed vertices in G , and let R' be any subset of R . If G contains an augmenting path whose endpoints are M -covered or an augmenting cycle, then it is also present in $G \setminus R'$. Since M is not a maximum-weight matching in $G \setminus R'$, R' is not an M -vertex-stabilizer. Similarly, if G contains an augmenting flower whose root is M -covered or an augmenting bi-cycle, then it is also present in $G \setminus R'$. By Theorem 2.4, $G \setminus R'$ is not stable. In these two cases, there is no M -vertex-stabilizer. Since $S \subseteq R$ and $w(M) < v_f(G \setminus S)$, the algorithm will return “INFEASIBLE”.

Thus, we may assume G does not contain any of the aforementioned structures. The only structure which can make G unstable is an augmenting flower whose root is M -exposed. In addition, the only structure which can prevent M from being a maximum-weight matching in G is an augmenting path with at least one M -exposed endpoint.

Claim 6.8. *Let u be an M -exposed vertex. Then, u is the root of an augmenting flower if and only if there exists an augmenting uu -walk of length at most $3n$.*

Proof. Let $C \cup P$ be a augmenting flower rooted at u where $C = (v_1, v_2, \dots, v_j, v_1)$ is the blossom and $P = (u_1, u_2, \dots, u_k)$ is the valid alternating path. Assume $u_1 = v_1$ and $u_k = u$. Let $P^{-1} = (u_k, u_{k-1}, \dots, u_1)$ denote the reverse of path P . Then, $Q = P^{-1}CP$ is a valid alternating uu -walk, and its length is at most $3n$. Moreover, since

$$w(Q \setminus M) = w(C \setminus M) + 2w(P \setminus M) > w(C \cap M) + 2w(P \cap M) = w(Q \cap M),$$

it is augmenting. For the converse, let P be an augmenting uu -walk of length at most $3n$. By Lemma 6.5, P contains an augmenting flower rooted at u . \square

Claim 6.9. *Let u be an M -exposed vertex and v be an M -covered vertex. If there is no augmenting flower rooted at u , then there exists an augmenting uv -path if and only if there exists an augmenting uv -walk of length at most $3n$.*

Proof. A uv -path is trivially a uv -walk. For the converse, let P be an augmenting uv -walk of length at most $3n$. By Lemma 6.5, P contains an augmenting uv -path. \square

By the two claims above, the set of vertices collected in the first “for” loop of the algorithm is exactly S_1 . Let S^* be a minimum M -vertex-stabilizer. Then, $S_1 \subseteq S^*$. Now, the only structure which can prevent M from being a maximum-weight matching in $G \setminus S_1$ is an augmenting path whose endpoints are both M -exposed.

Claim 6.10. *Let u and v be M -exposed vertices. There exists an augmenting uv -path in $G \setminus S_1$ if and only if there exists an augmenting uv -walk of length at most n in $G \setminus S_1$.*

Proof. A uv -path is trivially a uv -walk. For the converse, let P be an augmenting uv -walk of length at most n . By Lemma 6.5, P contains an augmenting uv -path. \square

Let S_2 be the set of vertices collected in the second “for” loop of the algorithm. At every iteration, a pair of vertices were added to S_2 because they are the endpoints of an augmenting path. Note that at least one of them is in S^* , otherwise this augmenting path is present in $G \setminus S^*$. Thus, we have $|S^*| \geq |S_1| + \frac{1}{2}|S_2| \geq \frac{1}{2}|S|$. The matching M is maximum-weight in $G \setminus (S_1 \cup S_2)$ because there are no augmenting paths or cycles. Moreover, $G \setminus (S_1 \cup S_2)$ is stable because it does not contain any augmenting flowers or bi-cycles. Thus, $S = S_1 \cup S_2$ is an M -vertex-stabilizer. Finally, if M is a maximum-weight matching in $G \setminus S_1$, then $S_2 = \emptyset$. We get $|S| = |S_1| \leq |S^*|$ implying that S is optimal. \square

6.3 Proof of Theorem 2.4

The goal of this section is to provide a proof for Theorem 2.4. Given a matching M , the stability requirement can be encoded as the following linear system:

$$\begin{aligned}
y_u &\geq 0 && \forall u \in V(M) \\
y_u &= 0 && \forall u \notin V(M) \\
y_u + y_v &= w_{uv} && \forall uv \in M \\
y_u + y_v &\geq w_{uv} && \forall uv \in E \setminus M
\end{aligned} \tag{6.1}$$

Converting equalities to inequalities, we obtain a system of the form $Ax \leq c$ where each inequality has at most two variables:

$$\begin{aligned}
-y_u &\leq 0 && \forall u \in V(M) \\
-y_u &\leq 0 && \forall u \notin V(M) \\
y_u &\leq 0 && \forall u \notin V(M) \\
-y_u - y_v &\leq -w_{uv} && \forall uv \in M \\
y_u + y_v &\leq w_{uv} && \forall uv \in M \\
-y_u - y_v &\leq -w_{uv} && \forall uv \in E \setminus M
\end{aligned} \tag{6.2}$$

Let S denote System 6.2. Then, determining whether a graph is stable is equivalent to deciding whether S is feasible for a maximum-weight matching M . We use a graph construction method introduced by Shostak [22]. Let y_z be an auxiliary variable that always occurs with zero coefficient. Then, without loss of generality we can assume every

inequality in S contains exactly two variables. Construct an undirected multigraph $G(S)$ as follows (see Figure 6.1):

- (a) For each variable y_v in S , add a vertex v to $G(S)$.
- (b) For each inequality $ay_u + by_v \leq c$ in S , add an undirected edge between u and v in $G(S)$, and label the edge with this inequality.

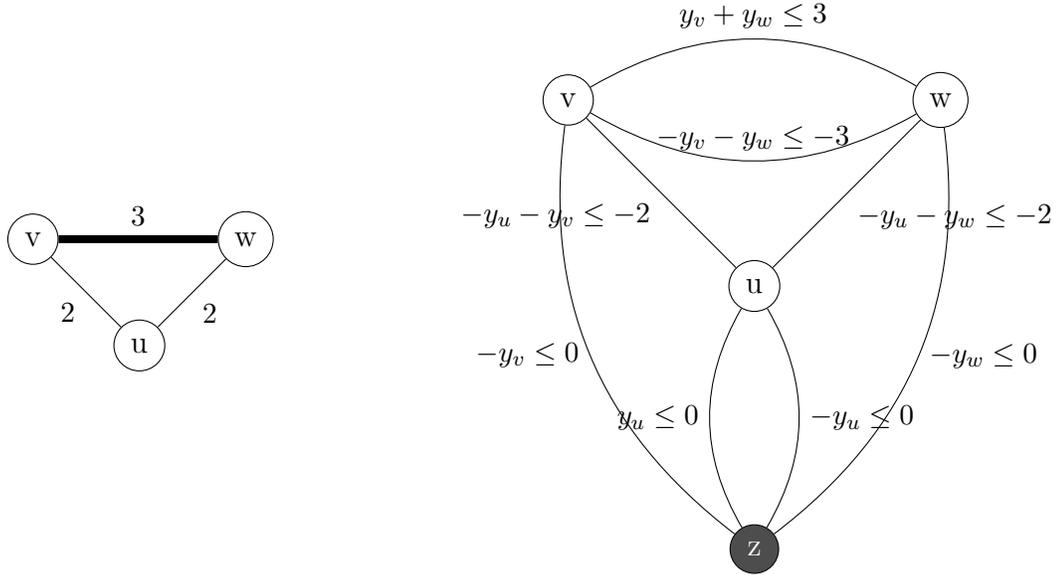


Figure 6.1: An example of $G(S)$ for a given graph and matching.

Let P be a walk in $G(S)$ determined by vertices $v_1, v_2, \dots, v_{\ell+1}$ and edges $e_1, e_2, \dots, e_{\ell}$. The *triple sequence* for P is given by

$$\langle a_1, b_1, c_1 \rangle, \langle a_2, b_2, c_2 \rangle, \dots, \langle a_{\ell}, b_{\ell}, c_{\ell} \rangle$$

where $a_i y_{v_i} + b_i y_{v_{i+1}} \leq c_i$ is the inequality associated with e_i for $1 \leq i \leq \ell$. P is called *admissible* if a_{i+1} and b_i have opposite signs for $1 \leq i < \ell$. Intuitively, an admissible walk is a chain of inequalities that exhibit the transitivity property. Define the *residue* of P as

$$\langle a_P, b_P, c_P \rangle = \langle a_1, b_1, c_1 \rangle \odot \langle a_2, b_2, c_2 \rangle \odot \dots \odot \langle a_{\ell}, b_{\ell}, c_{\ell} \rangle$$

where \odot is the associative binary operator defined on triples by

$$\langle a, b, c \rangle \odot \langle a', b', c' \rangle = \langle kaa', -kbb', k(ca' - c'b) \rangle \text{ and } k = a' / |a'|$$

Intuitively, the operator \odot takes two inequalities and derive a new inequality by eliminating their common variable. For example, $ay_u + by_v \leq c$ and $a'y_v + b'y_w \leq c'$ imply $-aa'y_u + bb'y_w \leq -(ca' - c'b)$ if $a' < 0$ (and $b > 0$). The resulting inequality $a_P v_1 + b_P v_\ell \leq c_P$ is known as the *hidden inequality* of P . In our setting, since coefficients can only be 0, 1 or -1, $ka' = (a')^2 = 1$ and $kb = -ka' = -1$. Thus, the operator \odot can be simplified to

$$\langle a, b, c \rangle \odot \langle a', b', c' \rangle = \langle a, b', c + c' \rangle$$

and the residue of P becomes

$$\langle a_P, b_P, c_P \rangle = \left\langle a_1, b_\ell, \sum_{i=1}^{\ell} c_i \right\rangle$$

Lemma 6.11 ([22]). *If P is an admissible walk with initial vertex u , final vertex v and residue $\langle a_P, b_P, c_P \rangle$, then any vector y which satisfies the inequalities that label the edges of P satisfies $a_P y_u + b_P y_v \leq c_P$.*

A walk is called a *loop* if its initial and final vertices are the same. A walk is *simple* if its intermediate vertices are distinct. An admissible loop P is said to be *infeasible* if $a_P + b_P = 0$ and $c_P < 0$.

Definition 6.12. Let S be a system of linear inequalities. For every vertex v in $G(S)$, and for every admissible simple loop P in $G(S)$ with $a_P + b_P \neq 0$ and initial vertex v , add the new inequality $(a_P + b_P)y_v \leq c_P$ to S . We call the new system S' the *Shostak extension* of S .

Note that there could be exponentially many admissible simple loops in $G(S)$. Notice also that $G(S)$ is a spanning subgraph of $G(S')$. The significance of the Shostak extension is given by the following theorem:

Theorem 6.13 ([22]). *Let S' be the Shostak extension of S . The system of inequalities S is feasible if and only if $G(S')$ has no infeasible simple loop.*

This theorem will be used to prove Theorem 2.4. We first establish the relationship between admissible simple loops in $G(S)$ and M -alternating structures in G .

Claim 6.14. *Let P be a simple loop in $G(S)$ which does not contain z and has length at least 3. Then, P is admissible if and only if P is an alternating cycle or a blossom in G .*

Proof. (\Rightarrow) Let $P = (v_1, v_2, \dots, v_k, v_1)$ be an admissible simple loop in $G(S)$ where $z \notin V(P)$ and $k \geq 3$. For the purpose of contradiction, assume $e_i = v_i v_{i+1}$, $e_{i+1} = v_{i+1} v_{i+2} \notin M$ for some $i \in [k-1]$. Then there is only one edge connecting v_i to v_{i+1} and v_{i+1} to v_{i+2} in $G(S)$. Moreover, their labels are $-y_{v_i} - y_{v_{i+1}} \leq -w_{e_i}$ and $-y_{v_{i+1}} - y_{v_{i+2}} \leq -w_{e_{i+1}}$ respectively. Since the signs of b_i and a_{i+1} are the same, P is not admissible. This is a contradiction. Thus, P is either an M -alternating cycle or an M -blossom with base v_1 .

(\Leftarrow) Let $P = (v_1, v_2, \dots, v_k, v_1)$ be an M -alternating cycle or an M -blossom with base v_1 in G . Then, $z \notin V(P)$ and $k \geq 3$. For every edge $e_i = v_i v_{i+1}$ in P , if it is matched, take the edge in $G(S)$ with label $y_{v_i} + y_{v_{i+1}} \leq w_{e_i}$. Otherwise, take the edge with label $-y_{v_i} - y_{v_{i+1}} \leq -w_{e_i}$. Since a_{i+1} and b_i have opposite signs for $1 \leq i < k$, P is an admissible simple loop in $G(S)$. \square

Next, we need to understand the inequalities that were added during the construction of the Shostak extension.

Lemma 6.15. *Let S' be the Shostak extension of S . The inequalities in $S' \setminus S$ are*

$$y_v \geq \frac{w(P \setminus M) - w(P \cap M)}{2}$$

for every vertex $v \in V$ and for every blossom P with base v .

Proof. Let $P = (v_1, v_2, \dots, v_k, v_1)$ be an admissible simple loop in $G(S)$. If $v_1 = z$, then $a_P = a_1 = 0$ and $b_P = b_k = 0$. Since $a_P + b_P = 0$, no inequality is added for P . So we may assume that $v_1 \neq z$. Note that $z \notin V(P)$ as otherwise P would not be admissible. Moreover, $k \geq 3$ as otherwise $a_P + b_P = a_1 - a_1 = 0$. By Claim 6.14, P is an alternating cycle or a blossom in G . If P is an alternating cycle, then it has even length. So the triple sequence associated with P is either

$$\langle -1, -1, -w_{v_1 v_2} \rangle, \langle 1, 1, w_{v_2 v_3} \rangle, \dots, \langle -1, -1, -w_{v_{k-1} v_k} \rangle, \langle 1, 1, w_{v_k v_1} \rangle$$

or

$$\langle 1, 1, w_{v_1 v_2} \rangle, \langle -1, -1, -w_{v_2 v_3} \rangle, \dots, \langle 1, 1, w_{v_{k-1} v_k} \rangle, \langle -1, -1, -w_{v_k v_1} \rangle$$

depending on whether $v_1 v_2 \in M$. In both cases, $a_P + b_P = 0$ so no inequality is added. If P is a blossom, then $v_1 v_2 \notin M$ and its triple sequence is given by

$$\langle -1, -1, -w_{v_1 v_2} \rangle, \langle 1, 1, w_{v_2 v_3} \rangle, \dots, \langle 1, 1, w_{v_{k-1} v_k} \rangle, \langle -1, -1, -w_{v_k v_1} \rangle$$

The residue of P is

$$\langle -1, -1, w(P \cap M) - w(P \setminus M) \rangle$$

Since $a_C + b_C = -2$, the following inequality would be added to S

$$-2y_{v_1} \leq w(P \cap M) - w(P \setminus M)$$

□

These are the only inequalities added when constructing S' . The term in the numerator can be interpreted as the difference between the total weight of unmatched edges and matched edges in P . Note that if the total weight of matched edges is higher, then this inequality is redundant because we already have $y_v \geq 0$ in S . Lastly, we look at admissible simple loops in $G(S')$. Observe that if an admissible simple loop in $G(S')$ does not contain z , then it is also an admissible simple loop in $G(S)$. By Claim 6.14, it is either an alternating cycle or a blossom in G . Hence, it suffices to look at admissible simple loops in $G(S')$ which start at z .

Claim 6.16. *Let P be a simple loop in $G(S')$ with initial vertex z . Then, P is admissible if and only if $P \setminus z$ is a valid alternating path in G .*

Proof. (\Rightarrow) Let $P = (v_1, v_2, \dots, v_k, v_1)$ be an admissible simple loop in $G(S')$ where $v_1 = z$. Then, $P' = P \setminus z$ is an admissible path. We claim that P' is M -alternating in G . For the purpose of contradiction, assume $e_i = v_i v_{i+1}, e_{i+1} = v_{i+1} v_{i+2} \notin M$ for some i . Then there is only one edge connecting v_i to v_{i+1} and v_{i+1} to v_{i+2} in $G(S')$. Moreover, their labels are $-y_{v_i} - y_{v_{i+1}} \leq -w_{e_i}$ and $-y_{v_{i+1}} - y_{v_{i+2}} \leq -w_{e_{i+1}}$ respectively. Since the signs of b_i and a_{i+1} are the same, P' is not admissible. This is a contradiction.

It is left to prove that P' is valid. If P is a loop of length two, then v_2 is M -exposed because the inequality $y_{v_2} \leq 0$ is present in $G(S')$. So we may assume that $k \geq 3$. Without loss of generality, we will just focus on one end of the path P' . The same argument holds for the other end of the path. Let $\langle a_1, b_1, c_1 \rangle$ be the triple associated with edge $v_1 v_2$. If $b_1 > 0$, then the inequality represented by this edge is $y_{v_2} \leq 0$. This implies that v_2 is M -exposed. On the other hand, if $b_1 < 0$, then $a_2 > 0$. Hence, the inequality represented by the edge $v_2 v_3$ is $y_{v_2} + y_{v_3} \leq w_{v_2 v_3}$, implying that $v_2 v_3 \in M$.

(\Leftarrow) Let $P' = (v_1, v_2, \dots, v_k)$ be a valid M -alternating path in G . For every edge $e_i = v_i v_{i+1}$ in P , if it is matched, take the edge in $G(S)$ with label $y_{v_i} + y_{v_{i+1}} \leq w_{e_i}$. Otherwise, take the edge with label $-y_{v_i} - y_{v_{i+1}} \leq -w_{e_i}$. Since a_{i+1} and b_i have opposite signs for $1 \leq i < k$, P' is an admissible path in $G(S)$. Now, if v_1 is M -exposed, append the edge with label $y_{v_1} \leq 0$ to P . Otherwise, we have $v_1 v_2 \in M$, so append the edge with label $-y_{v_1} \leq 0$ to P . Use a similar strategy for v_k . The resulting loop $P = (z, v_1, \dots, v_\ell, z)$ is simple and admissible. □

We are now ready to prove Theorem 2.4.

Proof of Theorem 2.4. We start by proving the contrapositive of the first statement. Let G be a graph and M be a maximum-weight matching in G . Assume G contains an M -augmenting flower or bi-cycle. We analyze them separately:

Case (i): G contains an augmenting flower. Let $C \cup P'$ be an M -augmenting flower, where $P' = (v_1, v_2, \dots, v_k)$ is the valid alternating path and C is the blossom with base v_1 . By Lemma 6.15, the inequality associated with C is added during the construction of S' . Define the simple loop $P = (z, v_1, v_2, \dots, v_k, z)$ such that the edge (z, v_1) is labelled with $-2y_{v_1} \leq w(C \cap M) - w(C \setminus M)$. If v_k is M -covered, we take the edge (v_k, z) with label $-y_k \leq 0$. Otherwise, we take the edge (v_k, z) with label $y_k \leq 0$. Then, P is admissible. Moreover, it is infeasible because $a_P + b_P = 0$ and $2c_P = w(C \cap M) - w(C \setminus M) + 2w(P' \cap M) - 2w(P' \setminus M) < 0$.

Case (ii): G contains an augmenting bi-cycle. Let $C \cup P' \cup D$ be an M -augmenting bi-cycle, where $P' = (v_1, v_2, \dots, v_k)$ is the alternating path and C, D are the blossoms with bases v_1, v_k respectively. By Lemma 6.15, the inequalities associated with C and D are added during the construction of S' . Define the simple loop $P = (z, v_1, v_2, \dots, v_k, z)$ such that the edge (z, v_1) has label $-2y_{v_1} \leq w(C \cap M) - w(C \setminus M)$ and the edge (v_k, z) has label $-2y_{v_k} \leq w(D \cap M) - w(D \setminus M)$. Then, P is admissible. It is also infeasible because $a_P + b_P = 0$ and $2c_P = w(C \cap M) - w(C \setminus M) + 2w(P' \cap M) - 2w(P' \setminus M) + w(D \cap M) - w(D \setminus M) < 0$.

In the two cases above, $G(S')$ has an infeasible simple loop. Hence, the linear system S is infeasible by Theorem 6.13. Thus, G is unstable. Next, we prove the second statement. Let G be an unstable graph and M be a maximum-weight matching in G . Then, the linear system S is infeasible. Let S' be the Shostak extension of S . By Theorem 6.13, $G(S')$ contains an infeasible simple loop P . We claim that P starts at vertex z . For the purpose of contradiction, suppose P does not contain z . Then, it has length at least 3 because it is infeasible. Moreover, it has even length because $a_P + b_P = 0$. By Claim 6.14, P is an M -alternating cycle in G . Since $0 > c_P = w(P \cap M) - w(P \setminus M)$, it is an M -augmenting cycle, which is a contradiction. Next, we claim that P contains at least one inequality from $S' \setminus S$. For the purpose of contradiction, suppose P only contains inequalities from S . By Claim 6.16, $P' = P \setminus z$ is a valid M -alternating path in G . As $0 > c_P = c_{P'} = w(P' \cap M) - w(P' \setminus M)$, it is an M -augmenting path, which is a contradiction. Denote $P = (z, v_1, v_2, \dots, v_k, z)$, and consider the following two cases:

Case (i): P contains one inequality from $S' \setminus S$. Without loss of generality, we may assume the edge (z, v_1) in P is labelled with $-2y_{v_1} \leq w(C \cap M) - w(C \setminus M)$ for some blossom C with base v_1 . If $k = 1$, then the edge (v_1, z) in P is labelled with $y_{v_1} \leq 0$, which implies

that v_1 is M -exposed. Otherwise, the edge (v_1, v_2) in P is labelled with $y_{v_1} + y_{v_2} \leq w_{v_1v_2}$, which implies that $v_1v_2 \in M$. From Claim 6.16, we know that $P' = P \setminus z$ is a valid M -alternating path. Since $0 > 2c_P = w(C \cap M) - w(C \setminus M) + 2w(P' \cap M) - 2w(P' \setminus M)$, $C \cup P'$ is an augmenting flower in G .

Case (ii): P contains two inequalities from $S' \setminus S$. Let the edge (z, v_1) in P be labelled with $-2y_{v_1} \leq w(C \cap M) - w(C \setminus M)$ and the edge (v_k, z) in P be labelled with $-2y_{v_k} \leq w(D \cap M) - w(D \setminus M)$, where C and D are blossoms with bases v_1 and v_k respectively. Note that k is odd because P is admissible. In particular, we have $v_1v_2, v_{k-1}v_k \in M$. Let $P' = P \setminus z$. Since $0 > 2c_P = w(C \cap M) - w(C \setminus M) + 2w(P' \cap M) - 2w(P' \setminus M) + w(D \cap M) - w(D \setminus M)$, $C \cup P' \cup D$ is an augmenting bi-cycle in G . \square

References

- [1] Sara Ahmadian, Hamideh Hosseinzadeh, and Laura Sanità. Stabilizing network bargaining games by blocking players. In *Proceedings of the 18th IPCO*, pages 164–177, 2016.
- [2] Bengt Aspvall and Yossi Shiloach. A polynomial time algorithm for solving systems of linear inequalities with two variables per inequality. *SIAM J. Comput.*, 9(4):827–845, 1980.
- [3] Egon Balas. Integer and fractional matchings. *North-Holland Mathematics Studies*, 59:1–13, 1981.
- [4] Michel Balinski. On maximum matching, minimum covering and their connections. In *Proceedings of the Princeton Symposium on Mathematical Programming*, pages 303–312, 1970.
- [5] Aditya Bhaskara, Moses Charikar, Eden Chlamtac, Uriel Feige, and Aravindan Vijayaraghavan. Detecting high log-densities: an $O(n^{1/4})$ approximation for densest k -subgraph. In *Proceedings of the 42nd STOC*, pages 201–210, 2010.
- [6] Péter Biró, Matthijs Bomhoff, Petr A. Golovach, Walter Kern, and Daniël Paulusma. Solutions for the stable roommates problem with payments. *Theor. Comput. Sci.*, 540:53–61, 2014.
- [7] Péter Biró, Walter Kern, and Daniël Paulusma. Computing solutions for matching games. *Int. J. Game Theory*, 41(1):75–90, 2012.
- [8] Adrian Bock, Karthekeyan Chandrasekaran, Jochen Könemann, Britta Peis, and Laura Sanità. Finding small stabilizers for unstable graphs. *Math. Program.*, 154(1-2):173–196, 2015.

- [9] Karthekeyan Chandrasekaran, Corinna Gottschalk, Jochen Könemann, Britta Peis, Daniel Schmand, and Andreas Wierz. Additive stabilizers for unstable graphs. *ArXiv e-prints*, Aug 2016.
- [10] William J. Cook, William H. Cunningham, William R. Pulleyblank, and Alexander Schrijver. *Combinatorial Optimization*. John Wiley & Sons, Inc., New York, NY, USA, 1998.
- [11] Xiaotie Deng, Toshihide Ibaraki, and Hiroshi Nagamochi. Algorithmic aspects of the core of combinatorial optimization games. *Math. Oper. Res.*, 24(3):751–766, 1999.
- [12] Irit Dinur and Shmuel Safra. On the hardness of approximating minimum vertex-cover. *Ann. Math.*, 162:439–485, 2005.
- [13] Takehiro Ito, Naonori Kakimura, Naoyuki Kamiyama, Yusuke Kobayashi, and Yoshio Okamoto. Efficient stabilization of cooperative matching games. *Theor. Comput. Sci.*, 677:69–82, 2017.
- [14] Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the 34th STOC*, pages 767–775, 2002.
- [15] Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within $2 - \epsilon$. *J. Comput. Syst. Sci.*, 74(3):335–349, 2008.
- [16] Jon M. Kleinberg and Éva Tardos. Balanced outcomes in social exchange networks. In *Proceedings of the 40th STOC*, pages 295–304, 2008.
- [17] Jochen Könemann, Kate Larson, and David Steiner. Network bargaining: Using approximate blocking sets to stabilize unstable instances. *Theory Comput. Syst.*, 57(3):655–672, 2015.
- [18] Sounaka Mishra, Venkatesh Raman, Saket Saurabh, Somnath Sikdar, and C. R. Subramanian. The complexity of könig subgraph problems and above-guarantee vertex cover. *Algorithmica*, 61(4):857–881, 2011.
- [19] George L. Nemhauser and Leslie E. Trotter Jr. Vertex packings: Structural properties and algorithms. *Math. Program.*, 8(1):232–248, 1975.
- [20] Oystein Ore. Note on hamilton circuits. *Amer. Math. Monthly*, 67:55, 1960.
- [21] Lloyd S. Shapley and Martin Shubik. The assignment game I: The core”. *International Journal of Game Theory*, 1(1):111–130, 1971.

- [22] Robert E. Shostak. Deciding linear inequalities by computing loop residues. *J. ACM*, 28(4):769–779, 1981.