Stabilizing Weighted Graphs

Zhuan Khye Koh
Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Canada
zkkoh@uwaterloo.ca

Laura Sanità
Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Canada
lsanita@uwaterloo.ca

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 are of independent interest. In particular, one of the main ingredients of our result is the development of a polynomial-time algorithm to compute a basic maximum-weight fractional matching with minimum number of *odd cycles* in its support. This generalizes a fundamental and classical result on unweighted matchings given by Balas more than 30 years ago, which we expect to prove useful beyond this particular application.

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.

2012 ACM Subject Classification Mathematics of computing → Matchings and factors, Mathematics of computing → Approximation algorithms, Mathematics of computing → Graph algorithms, Theory of computation → Discrete optimization, Theory of computation → Algorithmic game theory, Theory of computation → Network games

Keywords and phrases combinatorial optimization, network bargaining, cooperative game

Digital Object Identifier 10.4230/LIPIcs.ICALP.2018.83

Related Version A full version of this paper is available at `https://arxiv.org/abs/1709.01982`.

Funding This work was supported by the NSERC Discovery Grant Program and an Early Researcher Award by the Province of Ontario.

45th International Colloquium on Automata, Languages, and Programming (ICALP 2018).
Editors: Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella;
Article No. 83; pp. 83:1–83:13
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

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 *cooperative matching* games, introduced by Shapley and Shubik [17], and *network bargaining* games, defined by Kleinberg and Tardos [13], both extensively studied in the game theory community. Instances of such games are described by a graph $G = (V, E)$ with edge weights $w \in \mathbb{R}^E_{\geq 0}$, 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

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

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 [17], the goal is to find an allocation of the value $\nu(G)$ among the vertices, given as a vector $y \in \mathbb{R}^V_{\geq 0}$, 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. [8] 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 solution to the dual of (P) and has 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 [13], 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 $y \in \mathbb{R}^V_{\geq 0}$ 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 [6, 1, 10, 7, 14, 4, 5]. 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 paper, 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. [6] have showed that finding a minimum-cardinality edge-stabilizer is hard to approximate within a factor of $(2-\varepsilon)$, assuming Unique Game Conjecture (UGC) [11]. 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. [10] 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 [6, 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, and in fact, the development of algorithmic results for weighted graphs requires substantial new ideas.

Our results and techniques

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 which 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-complete 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 $\frac{1}{3}$, i.e. $\nu(G \setminus S) \leq \left(\frac{2}{3} + \varepsilon\right)\nu(G)$ (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 above result is based on two main ingredients. The first one is giving a lower bound on the cardinality 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 this generalization is not obvious (in fact, as we will show later, the same generalization does *not* hold for edge-stabilizers). Consequently, our proof is much more involved, and requires different ideas. The second main ingredient is giving a polynomial-time algorithm for computing an optimal basic solution to (P) with the smallest number of odd-cycles in its support, which is of independent interest, as highlighted in the next paragraph.

Computing maximum fractional matchings with minimum cycle support. The fractional matching polytope given by (P) has been extensively studied in the literature, and characterizing instances for which a maximum fractional matching equals an integral one is a natural graph theory question (see [6, 1]). It is well-known that basic solutions of (P) are half-integral, and the support of a basic solution is a disjoint union of a matching (given by 1-valued entries) and a set of odd-cycles (given by half-valued entries). Balas [2] gave a polynomial-time algorithm to compute a basic maximum fractional matching in an unweighted graph, with minimum number of odd-cycles in its support. This is a classical result on matching theory, which has been known for more than 30 years. In this paper, we generalize this result to arbitrary weighted instances, exploiting structural properties of basic fractional matchings. Our algorithm is based on combinatorial techniques, and we expect that this result will prove useful beyond this particular application.

Edge-stabilizers. When dealing with edge-removal operations, the stabilizer problem becomes harder, already in the unweighted setting. It is shown in [6] 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 [6] that a lower-bound on the cardinality of a minimum edge-stabilizer for unweighted graphs is again given by the minimum number of odd-cycles in the support of an optimal 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 [12].

Related work. Biró et al. [4] were the first to consider the edge-stabilizer problem in weighted graphs, and they showed NP-hardness for this case. Stabilizing a graph via different operations on the input graph (other than removing edges/vertices) has also been studied. In particular, Ito et al. [10] have given polynomial-time algorithms to stabilize an unweighted graph by adding edges and by adding vertices. Chandrasekaran et al. [7] have recently studied the problem of stabilizing unweighted graphs by fractionally increasing edge weights. 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, and gave approximation algorithms for this case.

Biró et al. [5] and Könemann et al. [14] 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 yield the existence of a fractional vertex cover of size at most $\nu(G)$ (but note that the resulting graph might not be stable). Mishra et al. [15] 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.

Paper Organization. In Section 2, we give some preliminaries and discuss notation. In Section 3, we give a polynomial-time algorithm to compute an optimal basic solution to (P) with minimum number of odd cycles in its support. This algorithm will be crucially used in Section 4, where we give our results on vertex-stabilizers. The sections on edge-stabilizers and additional results can be found in the full version of this paper. All missing proofs also appear in the full version.

2 Preliminaries and notation

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

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

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 $\operatorname{supp}(x) := \{e \in E : x_e \neq 0\}$ as the support of x. For any positive integer k, $[k]$ represents the set $\{1, 2, \ldots, k\}$.

Given an undirected graph G, we denote by n the number of vertices and by m the number of edges. For 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. For edge weights $w \in \mathbb{R}^m_+$, a valid M-alternating path P is called *M-augmenting* if $w(P \setminus M) > w(P \cap M)$. We will need the following classical result on the structure of basic fractional matchings:

▶ **Theorem 1** ([3]). *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

$$\mathcal{C}(\hat{x}) := \{C_1, \dots, C_q\} \quad \text{and} \quad M(\hat{x}) := \{e \in E : \hat{x}_e = 1\}$$

as the set of odd cycles such that $\hat{x}_e = \frac{1}{2}$ for all $e \in E(C_i)$ and the set of matched edges in $\hat{x}$ respectively. 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.** 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 3.** 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 $\gamma(G) = 0$.

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 [2] considered this problem on unweighted graphs, and gave the following characterization:

▶ **Theorem 4** ([2]). *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 1 for some examples):

▶ **Definition 5.** Let C be a cycle and $S_0, S_1, \dots, S_k$ be a partition of $V(C)$ such that $|S_0|$ is even and $k \geq 2$, where S_0 is allowed to be empty and $S_1, \dots, S_k$ are non-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:

▶ **Lemma 6.** *Let $C \cup M$ be a connector. For every terminal set S_i, there exists an M-alternating path in the connector from a vertex $v \in S_i$ to a vertex $u \in S_j$, for some $j \notin \{0, i\}$.*

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.

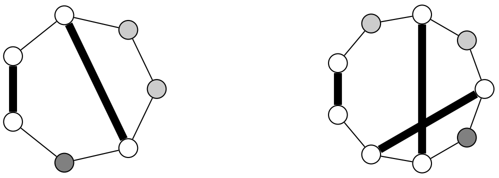

Figure 1 Two examples of connectors. Bold edges indicate M. Vertices of the same color belong to the same terminal set. White vertices are the ones in S_0.

▶ **Theorem 7.** *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, namely that alternate rounding and complementing produces a basic maximum-weight fractional matching with lesser odd cycles. 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

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

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 of $\mathcal{C}(\hat{x}) \cup \mathcal{C}(x^*)$.

(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 1.** *Every cycle in $\mathcal{C}(\hat{x})$ is adjacent to a cycle in $\mathcal{C}(x^*)$.*

Proof. Let C be an odd cycle in $\mathcal{C}(\hat{x})$. For every vertex $v \in V(C)$, 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 $\mathcal{C}(x^*)$. ◀

Recall that we assumed no two cycles in $\mathcal{C}(\hat{x})$ are adjacent. We also know that no two cycles in $\mathcal{C}(x^*)$ are adjacent. Since $|\mathcal{C}(\hat{x})| > |\mathcal{C}(x^*)|$, by the previous claim there exists an odd cycle in $\mathcal{C}(x^*)$ which is adjacent to at least two odd cycles in $\mathcal{C}(\hat{x})$. 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 $S_0 := V(C^*) \setminus \cup_{i=1}^k S_i$. Note that $y_v > 0$ for every vertex $v \in V(C^*)$. Hence, by complementary slackness every vertex in S_0 is $M(\hat{x})$-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 $C^* \cup \mathcal{P}$ where $i \neq j$ and $i, j \neq 0$. 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. Then, observe that the reduced graph along with the matching $M(\hat{x})$ forms a connector. By Lemma 6, there exists an $M(\hat{x})$-alternating path P from S_i to S_j in $C^* \cup \mathcal{P}$.

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. ◀

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 7. 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 2):

1. Delete all non-tight edges.
2. Add a vertex z.
3. For every vertex $v \in V$ where $\hat{x}(\delta(v)) = 1$ and $y_v = 0$, add the edge vz.
4. For every vertex $v \in V$ where $\hat{x}(\delta(v)) = 0$ and $y_v = 0$, add the vertex v' and the edges $vv', v'z$.
5. Shrink every odd cycle $C_i \in \mathcal{C}(\hat{x})$ into a pseudonode i.

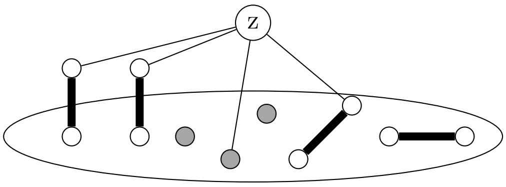

Figure 2 The auxiliary graph G' and the matching M'. Vertices in the ellipse are from the original graph G. Gray vertices represent pseudonodes.

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:

▶ **Lemma 8.** *M' is a maximum matching in G' if and only if $|\mathcal{C}(\hat{x})| = \gamma(G)$.*

Thus, searching for the structures in Theorem 7 is equivalent to searching for an M'-augmenting path in G'. This immediately gives us the following algorithm.

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 [9]
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 $\mathcal{C}(\bar{x}) = \mathcal{C}(\hat{x}) \backslash \{C_r, C_s\}$. 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 9.** *Algorithm 1 computes a basic maximum-weight fractional matching with $\gamma(G)$ odd cycles in polynomial time.*

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 [16], which involves solving the problem on a suitable bipartite graph.

4 Computing vertex-stabilizers

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

▶ **Theorem 10.** *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. Recall that a graph is stable if and only if $\gamma(G) = 0$. One strategy to achieve this is by showing that $\gamma(G)$ drops by at most 1 when a vertex is deleted (Lemma 12). We first develop a couple of claims.

▶ **Claim 2.** *Let $\hat{x}$ be a basic maximum-weight fractional matching and y be a minimum fractional w-vertex cover in G. Pick a vertex s from any 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 $G \setminus s$.*

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

▶ **Definition 11.** 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 3.** *Given two basic fractional matchings x and x', let H be the subgraph of G induced by* $\mathrm{supp}(x + x')$. *For any component K in H, switching on $E(K)$ from x to x' yields a basic fractional matching in G.*

▶ **Lemma 12.** *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 Claim 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 7 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.

For notational convenience, we can use $\mathcal{C}(\bar{x})$ and $M(\bar{x})$ to refer to the odd cycles and matched edges of $\bar{x}_{-\delta(u)}$ respectively because $\mathcal{C}(\bar{x}) = \mathcal{C}\left(\bar{x}_{-\delta(u)}\right)$ and $M(\bar{x}) = M\left(\bar{x}_{-\delta(u)}\right)$. Since $|\mathcal{C}(\bar{x})| = |\mathcal{C}(x^*)| - 1 = \gamma(G) - 1 > \gamma(G \setminus u)$, Theorem 7 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 7 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. So, let $s = V(C_i) \cap V(P)$ and 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 7.

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 $J = (V, N(x^*) \triangle N(\hat{x}))$. 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, but before jumping into them, 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 Claim 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 $\text{supp}(\hat{x} + \bar{x})$. 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 Claim 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 $\text{supp}(x^* + \bar{x})$. 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 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.

Subcase 2.3: $v \notin V(\mathcal{C}(x^*) \cup \mathcal{C}(\hat{x}))$. Let H be the subgraph of G induced by $\text{supp}(x^* + \hat{x})$. 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 $u \notin V(K)$. Hence, the resulting basic fractional matching is maximum-weight in G. ◀

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

▶ **Lemma 13.** *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
Initialize $S \leftarrow \emptyset$
Compute a minimum fractional $w$-vertex cover $y$ in $G$
Compute a basic maximum-weight fractional matching $\hat{x}$ in $G$ with $\gamma(G)$ odd cycles
Let $\mathcal{C}(\hat{x}) = \{C_1, C_2, \dots, C_{\gamma(G)}\}$
for $i = 1$ to $\gamma(G)$ do
    Let $v_i = \arg\min_{v \in V(C_i)} y_v$
    $S \leftarrow S + v_i$
return $S$
```

We are now ready to prove the main theorem stated at the beginning of the section, Theorem 10.

Proof of Theorem 10. 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 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 13. 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 2, we also know that y_{-S} is a minimum fractional w-vertex cover in $G \setminus S$. Then,

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

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 an example in the full version of this paper showing that, in general, this is not possible since 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. [4].

▶ **Theorem 14.** *Deciding whether a graph has a weight-preserving vertex-stabilizer is NP-complete.*

References

1 Sara Ahmadian, Hamideh Hosseinzadeh, and Laura Sanità. Stabilizing network bargaining games by blocking players. In *Proceedings of the 18th International Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 164–177, 2016.

2 Egon Balas. Integer and fractional matchings. *North-Holland Mathematics Studies*, 59:1–13, 1981.

3 Michel Balinski. On maximum matching, minimum covering and their connections. In *Proceedings of the Princeton Symposium on Mathematical Programming*, pages 303–312, 1970.

4 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.

5 Péter Biró, Walter Kern, and Daniël Paulusma. Computing solutions for matching games. *Int. J. Game Theory*, 41(1):75–90, 2012.

6 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.

7 Karthekeyan Chandrasekaran, Corinna Gottschalk, Jochen Könemann, Britta Peis, Daniel Schmand, and Andreas Wierz. Additive stabilizers for unstable graphs. *arXiv e-prints*, Aug 2016. `arXiv:1608.06797`.

8 Xiaotie Deng, Toshihide Ibaraki, and Hiroshi Nagamochi. Algorithmic aspects of the core of combinatorial optimization games. *Math. Oper. Res.*, 24(3):751–766, 1999.

9 Jack Edmonds. Paths, trees, and flowers. *Canad. J. Math.*, 17:449–467, 1965.

10 Takehiro Ito, Naonori Kakimura, Naoyuki Kamiyama, Yusuke Kobayashi, and Yoshio Okamoto. Efficient stabilization of cooperative matching games. *Theor. Comput. Sci.*, 677:69–82, 2017.

11 Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 767–775, 2002.

12 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.

13 Jon M. Kleinberg and Éva Tardos. Balanced outcomes in social exchange networks. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 295–304, 2008.

14 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.

15 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.

16 George L. Nemhauser and Leslie E. Trotter Jr. Vertex packings: Structural properties and algorithms. *Math. Program.*, 8(1):232–248, 1975.

17 Lloyd S. Shapley and Martin Shubik. The assignment game I: The core". *International Journal of Game Theory*, 1(1):111–130, 1971.