Efficient Stabilization of Cooperative Matching Games

Takehiro Ito[*]
Tohoku University
takehiro@ecei.tohoku.ac.jp

Naonori Kakimura[†]
University of Tokyo
kakimura@global.c.u-tokyo.ac.jp

Naoyuki Kamiyama[‡]
Kyushu University
kamiyama@imi.kyushu-u.ac.jp

Yusuke Kobayashi[§]
University of Tsukuba
kobayashi@sk.tsukuba.ac.jp

Yoshio Okamoto[¶]
University of Electro-Communications
okamotoy@uec.ac.jp

ABSTRACT

Cooperative matching games have drawn much interest partly because of the connection with bargaining solutions in the networking environment. However, it is not always guaranteed that a network under investigation gives rise to a stable bargaining outcome. To address this issue, we consider a modification process, called stabilization, that yields a network with stable outcomes, where the modification should be as small as possible. Therefore, the problem is cast to a combinatorial-optimization problem in a graph. Recently, the stabilization by edge removal was shown to be **NP**-hard. On the contrary, in this paper, we show that other possible ways of stabilization, namely, edge addition, vertex removal and vertex addition, are all polynomial-time solvable. Thus, we obtain a complete complexity-theoretic classification of the natural four variants of the network stabilization problem. We further study weighted variants and prove that the variants for edge addition and vertex removal are **NP**-hard.

1. INTRODUCTION

Matching markets play a central role in economics and game theory, and much work has been done. Among them, we concentrate on cooperative games with transferable utility derived from matchings in a network, called matching games, which are also known as stable roommate problems with side payments (when it is modeled as a cooperative game). The following example was given by Eriksson and Karlander [12] to motivate such games.

> In the professional tennis circuit, there are parallel singles and doubles tournaments. Although the prize money in the doubles is not as generous as in the singles, the sums are still impressive. The players are free to form pairs for the doubles as they choose. It is not necessarily so that the two best single players make the best team; the strength of a pair is a more complex function of the players abilities.
>
> The players know each others' strengths and weaknesses, and can make good estimates of the expected prize money for each possible constellation. In the process of forming pairs, the players negotiate how to distribute the expected income within the pair. (At least, this is what rational tennis players should do.) The objective of each player in this process is to maximize his own expected prize money.

As a cooperative game with transferable utility, in a *matching game*, we are given a network (or an undirected graph), and each vertex of the graph corresponds to a player of the game. A *matching* is a set of pairs of players such that every player is involved in at most one pair, and a pair can be formed only if there is an edge between two players. The graph is often associated with edge weights so that the utility of forming a pair can be incorporated. The characteristic function value of a coalition is defined as the maximum possible total utility of a matching over the coalition.

Matching games have attracted researchers, and solution concepts for matching games have been studied through the algorithmic lens [13, 7, 6, 18, 17, 9, 2]. In particular, there is a polynomial-time algorithm to test whether a given matching game has a non-empty core [10], where the *core* is defined as the set of payoff vectors such that no coalition has a characteristic function value larger than the total payoff allocated to the players of the coalition. Thus, a "stable" payoff exists in a game with a non-empty core. We here note

[*]Supported by JSPS KAKENHI Grant Numbers 25330003, 15H00849.

[†]Supported by JST, ERATO, Kawarabayashi Large Graph Project, and by JSPS KAKENHI Grant Numbers 25730001, 24106002.

[‡]Supported by JST, PRESTO.

[§]Supported by JST, ERATO, Kawarabayashi Large Graph Project, and by JSPS KAKENHI Grant Numbers 24106002, 24700004.

[¶]Supported by JSPS KAKENHI Grant Numbers 24106005, 24700008, 24220003, 15K00009.

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*

Table 1: The summary of our results. The mark $*$ depicts the results in this paper.

Unweighted	Edge	Vertex
Removal	**NP**-hard [8]	Poly-time [$*$]
Addition	Poly-time [$*$]	Poly-time [$*$]

Weighted	Edge	Vertex
Removal	**NP**-hard [6]	**NP**-hard [$*$]
Addition	**NP**-hard [$*$]	—

that if the underlying network is bipartite, then the game is called an *assignment game*, which is also well studied in the literature [26].

On the other hand, if the core of a game is empty, then no payoff is stable. Therefore, we often modify the game itself to achieve the core non-emptiness. The literature offers options of taxation [22, 28], cost of stability [3], and least cores [18]. In this paper, we consider a structural modification of the underlying network, which better fits another motivation from network bargaining as explained later. The modification should be as small as possible so that the games themselves do not differ much.

There are several possible ways of modification. We consider the following four processes.

Edge removal: We remove a set of edges from the network. In this case, we want to remove as few edges as possible.

Edge addition: We add a set of edges to the network. In this case, we want to add as few edges as possible.

Vertex removal: We remove a set of vertices from the network. When we remove a vertex, all the edges incident to the vertex are removed too. In this case, we want to remove as few vertices as possible.

Vertex addition: We add a set of vertices to the network. When we add a vertex, we may also add edges from the new vertex to any of the existing vertices. In this case, we want to add as few vertices as possible.

The literature only studied edge removal. Namely, the problem was to find a smallest subset of edges whose removal resulted in a network with non-empty core. Biró et al. [6] proved that the problem is **NP**-hard for the weighted case. Later, Bock et al. [8] proved that the problem is still **NP**-hard for the unweighted case (as introduced above), and hard to approximate within factor less than two assuming the unique games conjecture [19]. Some approximation algorithms have been proposed [8, 21], but the existence of a constant-factor approximation algorithm is left open.

Our Results.

We study the remaining three options, namely, edge addition, vertex removal, and vertex addition. For all of them, we prove that the problems can be solved in polynomial time (Sections 3, 4, and 5, respectively). In this way, we obtain the complete complexity classification of the problem variants, and reveal that edge removal studied in the literature is the only hard case.

Those polynomial-time algorithms are obtained by a thorough treatment of the so-called Gallai–Edmonds decompositions that possess useful information on the structure of maximum matchings in a graph. We also utilize the theory of linear programming, as explained later, to connect our problems with fractional matchings of a graph.

One may think that in vertex addition, we may add a lot of edges at the same time even though we add only one vertex. However, as it turns out, our algorithm finds a solution which adds as few edges as possible among all the possible solutions for the vertex addition variant.

We also consider a weighted variant. In the edge addition variant, each possible edge is associated with a non-negative real number. In the vertex removal variant, each vertex is associated with a non-negative real number. Those numbers represent costs of addition or removal. That models a more realistic situation where each possible modification operation is not equivalently manageable. With this setup, we prove that for the edge addition and the vertex removal, the problem is **NP**-hard (Section 6).

The results are summarized in Table 1.

Simultaneously with us, Ahmadian, Sanita, and Hosseinzadeh [1] consider the vertex removal variant, and prove its polynomial-time solvability and the **NP**-hardness of the weighted variant. Furthermore, they propose approximation algorithms for the weighted variant.

Relation to Network Bargaining.

Bateni et al. [4] related the core non-emptiness of matching games with a desirable property in network bargaining by Kleinberg and Tardos [20]. We briefly look at the relationship here.

In their seminal paper, Kleinberg and Tardos [20] gave a theoretical and mathematical foundation for network bargaining. Their model assumes that each vertex of a network corresponds to a player, each edge corresponds to a possible unit-value deal between two players, and each player can be engaged in at most one deal with one of its neighbors in the network. Then, outcome consists of a matching of the network and a value obtained by each player. If a player is matched with another player, then a unit payoff is split to those two players as their values, and if a player is not matched to any other player, then the value of the player is zero. The classical two-player situation studied by Nash [24] corresponds to the two-vertex network with one edge.

Given an outcome, we may consider a deviation of a player from the current matching. Namely, a player A has an incentive to be matched with another player B if B is a neighbor of A in the network and the value of A is smaller than the value that can be obtained by a deal with B while the value of B stays the same. If no player has such an incentive, then the outcome is called stable. On the other hand, the outcome is called balanced if two players joined by an edge of the matching receive the values as Nash's bargaining solution. A main result of Kleinberg and Tardos [20] stated that a stable outcome exists if and only if a balanced outcome exists. Bateni et al. [4] proved that this is also equivalent to the property that the matching game on the same network has a non-empty core.

With this equivalence, following Bock et al. [8], we call the modification process to turn a given network into one with non-empty core *stabilization*, and if a graph yields a matching game with non-empty core, we call the graph *stable*.

We here emphasize that our structural modification is better suited than modification processes when we consider least cores or cost of stability. The modification for the latter two does not maintain the structure of characteristic functions as they arise from matching games, and thus does not give any implication to network bargaining.

Relation to Linear Programming.

It is known that theory of linear programming is a key to study the core non-emptiness of cooperative games arising from graphs and combinatorics. For a matching game, this translates to the relation of the matching number and the fractional matching number of a graph.

A matching of a graph can be seen as an assignment of zero or one to each edge of the graph so that for every vertex of the graph, the sum of the assigned values to the edges incident to that vertex is at most one. A *fractional matching* of a graph is defined as an assignment of non-negative real number to each edge of the graph so that for every vertex of the graph, the sum of the assigned values to the edges incident to that vertex is at most one. The *matching number* is defined as the maximum size of a matching in the graph, which can be seen as the maximum sum of the values assigned to the edges of the graph when the assignment corresponds to a matching. Similarly, the *fractional matching number* is defined as the maximum sum of the values assigned to the edges of the graph in a fractional matching.

The computation of the matching number can naturally be formulated as a $0/1$ integer linear program, while the computation of the fractional matching number can be formulated as a linear program without integrality constraint.

By definition, the fractional matching number is always at least as large as the matching number. A result by Deng et al. [10] stated that those two are equal if and only if the matching game has a non-empty core. Moreover, if the core is non-empty, then it is exactly the set of optimal solutions to the dual of the linear program formulating the computation of the fractional matching number. The special case of bipartite graphs (i.e., assignment game) was already proved by Shapley and Shubik [26].

In the terminology of optimization, the difference of the fractional matching number and the matching number is called *integrality gap*, which is relevant in developing optimization algorithms, both exact and approximate. Our results are concerned with smallest modifications to input graphs such that the integrality gaps will be zero, and we use this view in designing our algorithms.

2. PRELIMINARIES

Matching Number and Stable Graphs.

In this paper, we are given a simple undirected graph G with a vertex set V and an edge set E, i.e., $G = (V, E)$. We denote by $\{u, v\}$ the edge between a pair of vertices u, v in V. For each subset X of V, we define $E(X) := \{\{u, v\} \in E \mid u, v \in X\}$. For each subset X of V, the subgraph of G induced by X is denoted by $G[X]$, i.e., $G[X] = (X, E(X))$. For each vertex v in V, we denote by $\delta(v)$ the set of edges incident to v. For each subset X of V, we define $\delta(X) := \bigcup_{v\in X} \delta(v) \setminus E(X)$, i.e., $\delta(X)$ represents the set of edges leaving from X. For each subset X of V, we define the *neighborhood* $N(X)$ as the set of vertices v in $V \setminus X$ such that there is a vertex u in X with $\{u, v\} \in E$. We denote by $\binom{V}{2}$ the family of subsets X of V such that $|X| = 2$. For each subset F of $\binom{V}{2} \setminus E$, we denote by $G + F$ the graph obtained from G by adding edges in F to G, i.e., $G + F = (V, E \cup F)$. For each subset X of V, we denote by $G - X$ the graph obtained from G by removing vertices in X from G, i.e., $G - X = (V \setminus X, E \setminus \bigcup_{v\in X} \delta(v))$. For each edge e in $\binom{V}{2} \setminus E$ (resp., each vertex v in V), we define $G + e := G + \{e\}$ (resp., $G - v := G - \{v\}$). A subset M of E is called a *matching* in G, if no two edges in M share a common vertex incident to them. The *matching number* of G, denoted by $\nu(G)$, is the maximum size of a matching in G. A matching in G is *perfect* if $|M| = |V|/2$.

Finding a maximum-size matching in G can be formulated by the following integer programming problem IP(G):

$$\nu(G) = \max \left\{ \sum_{e\in E} x(e) \;\middle|\; \begin{array}{l} \sum_{e\in\delta(v)} x(e) \le 1 \ (v \in V), \\ x \in \{0,1\}^E \end{array} \right\}.$$

Consider the linear programming relaxation of IP(G), denoted by LP(G):

$$\nu_f(G) := \max \left\{ \sum_{e\in E} x(e) \;\middle|\; \begin{array}{l} \sum_{e\in\delta(v)} x(e) \le 1 \ (v \in V), \\ x \in \mathbb{R}_+^E \end{array} \right\},$$

where $\mathbb{R}_+$ is the set of non-negative real numbers. The optimal value $\nu_f(G)$ of LP(G) is called the *fractional matching number of* G. Then $\nu_f(G) \ge \nu(G)$ holds. The dual problem DP(G) of LP(G) is

$$\tau_f(G) := \min \left\{ \sum_{v\in V} y(v) \;\middle|\; \begin{array}{l} y(u) + y(v) \ge 1 \ (\{u, v\} \in E), \\ y \in \mathbb{R}_+^V \end{array} \right\}.$$

By the strong duality theorem in linear programming, $\nu_f(G) = \tau_f(G)$. Remark that the problem DP(G) is the linear programming relaxation of the problem of finding a minimum-size vertex cover in G, where a *vertex cover* is a subset X of V such that every edge of G has an incident vertex in X. The minimum size of a vertex cover is denoted by $\tau(G)$.

Recall that a graph G is called stable if the network bargaining on G has a stable outcome, equivalently, the matching game on G has a non-empty core. As mentioned before, the stability of a graph G can be characterized as follows.

PROPOSITION 1 ([10]). *A graph G is stable if and only if $\nu_f(G) = \nu(G)$.*

Therefore, stabilization of a graph G can be seen as the problem of modifying G so that the resulting graph G' satisfies $\nu_f(G') = \nu(G')$. For example, in the edge addition variant, we aim to find a minimum-size subset F of $\binom{V}{2} \setminus E$ such that $\nu_f(G + F) = \nu(G + F)$.

Gallai–Edmonds Decomposition.

For each subset M of E, we denote by ∂M the set of end vertices of M, i.e., $\partial M = \bigcup_{e\in M} e$. We say that a vertex v is *covered by* M if $v \in \partial M$, and *exposed* by M if $v \notin \partial M$. A vertex v in V is said to be *essential* if v is covered by every maximum-size matching in G, and *inessential* otherwise, i.e., if some maximum-size matching in G does not cover v.

Definition 1. (Gallai–Edmonds Decomposition [11, 14, 15]) Define a partition (B, A, D) of V as follows:

- B is the set of the inessential vertices;
- $A = N(B)$, i.e., A is the neighborhood of B;
- $D = V \setminus (B \cup A)$.

The partition (B, A, D) is called the *Gallai–Edmonds decomposition* of G.

It is known that the Gallai–Edmonds decomposition (B, A, D) of G uniquely exists and can be found in polynomial time (see e.g., [23]). By definition, every vertex in $A \cup D$ is covered by a maximum-size matching M in G, and vertices exposed by M are contained only in B.

Furthermore, the well-known Tutte–Berge formula can be written in terms of the Gallai–Edmonds decomposition.

PROPOSITION 2. (Tutte–Berge Formula [5, 27]) *We have*

$$\nu(G) = \frac{1}{2}\left(|V| - \mathrm{comp}(G[B]) + |A|\right),$$

where comp$(G[B])$ *is the number of connected components in* $G[B]$.

Bock et al. [8] characterize the difference $\nu_f(G) - \nu(G)$ using the Gallai–Edmonds decomposition to bound the size of an edge-removal stabilizer from below. In the rest of this section, we provide explicit optimal solutions for LP(G) and DP(G) using the Gallai–Edmonds decomposition (Proposition 4). This will be used to obtain a minimum-size stabilizer for the other variants. Note that the case of LP(G) was given in [25]. For the purpose, we first summarize fundamental properties of the Gallai–Edmonds decomposition (see e.g., [23]):

(a) Each connected component H in $G[B]$ is factor-critical, where a graph G' is *factor-critical* if there is a perfect matching in $G' - v$ for every vertex v of G'. Thus H has odd number of vertices.

(b) For a maximum-size matching M, each connected component H in $G[B]$ satisfies either (i) H has one vertex exposed by M and no edge in M leaving from H, or (ii) H has no vertex exposed by M and one edge in M leaving from H.

Let B_1 be the vertex set of singleton components in $G[B]$. Let $B_3 = B \setminus B_1$ be the vertex set of the other *non-trivial* connected components (i.e., connected components with size at least three) in $G[B]$. In this paper, we sometimes call the partition (B_1, B_3, A, D) the Gallai–Edmonds decomposition.

For each maximum-size matching M in G, an *M-alternating* cycle C is a cycle such that edges of M appear alternately in C (except for one vertex if the length is odd). An *M-alternating* path is defined similarly. From the structure of the Gallai–Edmonds decomposition, we have the following lemma.

LEMMA 1. *If a connected component H in $G[B_3]$ has a vertex v exposed by M, then H contains an M-alternating cycle of odd length through v.*

Let $G[B_1, A]$ be the bipartite graph induced by edges between B_1 and A. A maximum-size matching in G is said to be *B_1-optimal*, if $|\partial M \cap B_1| = \nu(G[B_1, A])$, i.e., the number of edges of M connected to B_1 is maximized. By an augmenting-type algorithm for finding a maximum-size matching, we can find a B_1-optimal matching in polynomial time.

PROPOSITION 3. *A B_1-optimal matching in G always exists, and can be found in polynomial time.*

With help of a B_1-optimal matching, we can give optimal solutions of LP(G) and DP(G).

PROPOSITION 4. *Assume that M is a B_1-optimal matching in G, and $v_1, v_2, \ldots, v_p$ are the vertices exposed by M in $G[B_3]$.*

1. *Let $C_1, C_2, \ldots, C_p$ be pairwise vertex-disjoint odd M-alternating cycles such that C_i has v_i for every $i = 1, 2, \ldots, p$. Then, the vector x in $\mathbb{R}_+^E$ defined by*

$$x(e) = \begin{cases} 1 & \text{if } e \in M \setminus \bigcup_{i=1}^p EC_i, \\ 1/2 & \text{if } e \in \bigcup_{i=1}^p EC_i, \\ 0 & \text{otherwise} \end{cases}$$

is an optimal solution for LP(G)*, where EC_i is the edge set of C_i for each $i = 1, 2, \ldots, p$.*

2. *Let Y be a minimum vertex cover in $G[B_1, A]$. Then, the vector y in $\mathbb{R}_+^V$ defined by*

$$y(v) = \begin{cases} 1 & \text{if } v \in A \cap Y, \\ 0 & \text{if } v \in B_1 \setminus Y, \\ 1/2 & \text{otherwise} \end{cases}$$

is an optimal solution for DP(G).

3. *The optimal values of* LP(G) *and* DP(G) *are equal to*

$$\frac{1}{2}\big(|V \setminus B_1| + \nu(G[B_1, A])\big). \tag{1}$$

PROOF. We will show that x and y defined in the statements are feasible solutions for LP(G) and DP(G), respectively, with the same objective value (1).

It is clear that x is feasible for LP(G) since $\sum_{e \in \delta(v)} x(e) \in \{0, 1\}$ for every vertex v in V. Since $\sum_{e \in \delta(v)} x(e) = 0$ if and only if $v \in B_1 \setminus \partial M$, the objective value of x is

$$\sum_{e \in E} x(e) = \frac{1}{2} \sum_{v \in V} \sum_{e \in \delta(v)} x(e) = \frac{1}{2}\big(|D| + |A| + |B_3| + |B_1 \cap \partial M|\big).$$

Thus, since $|V| = |D| + |A| + |B_3| + |B_1|$ and $|B_1 \cap \partial M| = \nu(G[B_1, A])$, it is equal to (1).

On the other hand, the vector y is feasible for DP(G). Indeed, if an edge has an end vertex $v \in B_1 \setminus Y$ with $y(v) = 0$, the other end vertex u is in $A \cap Y$, since Y is a vertex cover in $G[B_1, A]$, and hence $y(u) = 1$. Thus, every edge $\{u, v\}$ satisfies $y(u) + y(v) \geq 1$. The objective value of y is

$$\sum_{v \in V} y(v) = \frac{|D|}{2} + \frac{|A \setminus Y|}{2} + |A \cap Y| + \frac{|B_1 \cap Y|}{2} + \frac{|B_3|}{2} = \frac{1}{2}\big(|D| + |A| + \nu(G[B_1, A]) + |B_3|\big),$$

where the last equality holds by $|A \cap Y| + |B_1 \cap Y| = |Y| = \nu(G[B_1, A])$. Hence the objective value is equal to (1).

Thus they are equal to (1), which means that they are optimal by the duality theorem of linear programming. □

Denote the difference between $\nu_f(G)$ and $\nu(G)$ by

$$d(G) := \nu_f(G) - \nu(G).$$

Combining Propositions 2 and 4, we can express $d(G)$ by the Gallai–Edmonds decomposition.

PROPOSITION 5. *It holds that*

$$d(G) = \frac{1}{2}\big(\mathrm{comp}(G[B_3]) - |A| + \nu(G[B_1, A])\big). \tag{2}$$

Bock et al. [8] showed that $2d(G)$ is equal to the number of components having exposed vertices in $G[B_3]$ for a B_1-optimal matching. By Proposition 5, it is equal to $\mathrm{comp}(G[B_3]) - |A| + \nu(G[B_1, A])$.

3. EDGE ADDITION

A subset F of $\binom{V}{2} \setminus E$ is called an *edge-addition stabilizer*, if $\nu(G+F) = \nu_f(G+F)$. In this section, we discuss the problem of deciding whether there exists an edge-addition stabilizer, and find a minimum-size edge-addition stabilizer if one exists.

We first describe a necessary condition that G has an edge-addition stabilizer.

THEOREM 1. *If $|V|$ is odd and $\nu(G[B_1, A]) = |B_1|$, then G has no edge-addition stabilizer.*

PROOF. Since $\nu(G[B_1, A]) = |B_1|$, the optimal value (1) in Proposition 4 is equal to $n/2$, where $n = |V|$. Hence, for every subset F of $\binom{V}{2} \setminus E$,

$$\nu_f(G+F) \geq \nu_f(G) = \frac{n}{2}.$$

On the other hand, since n is odd,

$$\nu(G+F) \leq \left\lfloor \frac{n}{2} \right\rfloor = \frac{n-1}{2}$$

for every subset F of $\binom{V}{2} \setminus E$. Thus, G does not have an edge-addition stabilizer. □

We next show that, if G does not satisfy the conditions in Theorem 1, then the minimum size of an edge-addition stabilizer is determined by $d(G)$.

THEOREM 2. *If $|V|$ is even or $\nu(G[B_1, A]) < |B_1|$, then G has an edge-addition stabilizer, and the minimum size of an edge-addition stabilizer is equal to $\lceil d(G) \rceil$.*

For the proof, we first consider adding one edge to a given graph in the following lemma, and the proof of Theorem 2 follows.

LEMMA 2. *For every edge e in $\binom{V}{2} \setminus E$,*

$$d(G) - d(G+e) \leq 1.$$

PROOF. It holds that

$$\nu(G+e) \leq \nu(G) + 1, \tag{3}$$

since otherwise a maximum-size matching in $G+e$ would yield a matching of size at least $\nu(G+e) - 1 > \nu(G)$ in G. On the other hand, since a feasible solution x of $\mathrm{LP}(G)$ is feasible for $\mathrm{LP}(G+e)$ by setting $x(e) = 0$,

$$\nu_f(G+e) \geq \nu_f(G). \tag{4}$$

Therefore, it follows from (3) and (4) that

$$\begin{aligned} &d(G) - d(G+e) \\ &\quad = (\nu_f(G) - \nu_f(G+e)) - (\nu(G) - \nu(G+e)) \leq 1. \end{aligned}$$

This completes the proof. □

PROOF OF THEOREM 2. We first prove that it is necessary to add at least $\lceil d(G) \rceil$ edges. Let $F = \{e_1, e_2, \ldots, e_p\}$ be an edge-addition stabilizer of size p. Furthermore, define $F_i := \{e_1, e_2, \ldots, e_i\}$ for each $i = 1, 2, \ldots, p$, and $F_0 := \emptyset$. Then, by applying Lemma 2 repeatedly,

$$d(G) - d(G+F) = \sum_{i=1}^{p} \left(d(G+F_{i-1}) - d(G+F_i)\right) \leq p.$$

Since $d(G+F) = 0$ and p is an integer, we see $\lceil d(G) \rceil \leq p$. Thus, the minimum size of an edge-addition stabilizer is at least $\lceil d(G) \rceil$.

Let M be a B_1-optimal matching in G. Then, $G[B_3]$ has $2d(G)$ vertices exposed by M (see the end of Section 2). We will find an edge-addition stabilizer of size $\lceil d(G) \rceil$ by considering the following two cases.

First suppose that $2d(G)$ is even. We make $d(G)$ disjoint pairs of exposed vertices in $G[B_3]$ arbitrarily, and denote the set of the pairs by F^*. The size of F^* is $d(G)$. Since $M \cup F^*$ is a matching of size $\nu(G) + |F^*|$ in $G + F^*$, we observe that

$$\nu(G+F^*) \geq \nu(G) + |F^*|. \tag{5}$$

Define a half-integral optimal dual solution y of $\mathrm{DP}(G)$ as in Proposition 4. Since the end vertices of each pair in F^* have value $1/2$, the vector y is feasible also for $\mathrm{DP}(G+F^*)$. Thus, $\tau_f(G+F^*) \leq \tau_f(G)$, and by the duality theorem in linear programming, we have

$$\nu_f(G+F^*) \leq \nu_f(G). \tag{6}$$

Therefore, it follows from (5), (6), and $|F^*| = d(G)$ that

$$\begin{aligned} d(G+F^*) &= \nu_f(G+F^*) - \nu(G+F^*) \\ &\leq \nu_f(G) - (\nu(G) + |F^*|) = 0. \end{aligned}$$

Thus, F^* is an edge-addition stabilizer.

Next suppose that $2d(G)$ is odd. In this case,

$$\nu(G[B_1, A]) < |B_1|,$$

since otherwise $|V|$ has to be even by the assumption but $|V| = 2|M| + 2d(G)$, which is a contradiction. We make $d(G)$ arbitrary pairs using the exposed vertices in $G[B_3]$ and one exposed vertex s in B_1, and the set of the pairs is denoted by F^*. The size of F^* is equal to $(2d(G)+1)/2 = \lceil d(G) \rceil$. Similarly to the first case, since $M \cup F^*$ is a matching in $G + F^*$, we observe that

$$\nu(G+F^*) \geq \nu(G) + |F^*|. \tag{7}$$

Moreover, define a half-integral optimal dual solution y of $\mathrm{DP}(G)$ as in Proposition 4. By increasing the value $y(s)$ by $1/2$, the vector becomes feasible for $\mathrm{DP}(G+F^*)$. Hence

$$\begin{aligned} \nu_f(G+F^*) &= \tau_f(G+F^*) \\ &\leq \tau_f(G) + 1/2 = \nu_f(G) + 1/2. \end{aligned} \tag{8}$$

Therefore, (7) and (8) imply that

$$\begin{aligned} d(G+F^*) &= \nu_f(G+F^*) - \nu(G+F^*) \\ &\leq (\nu_f(G) + 1/2) - (\nu(G) + |F^*|) = 0, \end{aligned}$$

since $|F^*| = d(G) + 1/2$. Thus, F^* is an edge-addition stabilizer.

Therefore, F^* is an edge-addition stabilizer of size $\lceil d(G) \rceil$ in either case, which is minimum. □

Theorems 1 and 2 lead to a polynomial-time algorithm for finding a minimum-size edge-addition stabilizer.

Algorithm for finding an edge-addition stabilizer

Step 0. Find the Gallai–Edmonds decomposition (B_1, B_3, A, D) of G.

Step 1. Find a maximum-size matching M^* in $G[B_1, A]$. If $|V|$ is odd and $\nu(G[B_1, A]) = |B_1|$, then return that "G has no edge-addition stabilizer." Otherwise, go to Step 2.

Step 2. Find a B_1-optimal matching M.

Step 3. Return $\lceil d(G) \rceil$ pairs of vertices from the exposed vertices in B_3 and one exposed vertex in B_1 (if $2d(G)$ is odd).

The Gallai–Edmonds decomposition and a maximum-size matching M^* in $G[B_1, A]$ can be found in polynomial time. Moreover, by Proposition 3, a B_1-optimal matching M in Step 2 can be obtained in polynomial time. Since Step 3 is easy, we obtain the following theorem.

THEOREM 3. *We can decide whether an edge-addition stabilizer exists, and find a minimum-size edge-addition stabilizer if one exists, in polynomial time.* □

4. VERTEX REMOVAL

A subset X of V is called *a vertex-removal stabilizer*, if $\nu(G - X) = \nu_f(G - X)$. Note that since $\nu(G - V) = 0 = \nu_f(G - V)$, a vertex-removal stabilizer always exists. In this section, we discuss the problem of finding a minimum-size vertex-removal stabilizer.

The main theorem in this section is the following.

THEOREM 4. *The minimum size of a vertex-removal stabilizer is equal to* $2d(G)$.

To prove Theorem 4, we first consider removing one vertex from a graph. Let (B_1, B_3, A, D) be the Gallai–Edmonds decomposition of G, and $B = B_1 \cup B_3$.

LEMMA 3. *For every vertex v in V,*

$$d(G) - d(G - v) \leq \begin{cases} 0 & \text{if } v \in V \setminus B, \\ 1/2 & \text{if } v \in B, \end{cases}$$

PROOF. The case when $v \in V \setminus B$ is not difficult to see by the definition of the Gallai–Edmonds decomposition. We here prove only when $v \in B$.

Since v is inessential, it follows that

$$\nu(G - v) = \nu(G). \tag{9}$$

We will show that

$$\nu_f(G - v) \geq \nu_f(G) - \frac{1}{2}. \tag{10}$$

If (10) is true, together with (9), we have $d(G) - d(G - v) = (\nu_f(G) - \nu_f(G - v)) - (\nu(G) - \nu(G - v)) \leq 1/2$, which completes the proof.

It remains to show (10). Let M be a B_1-optimal matching in G. Define a half-integral optimal solution x of LP(G) as in Proposition 4. Note that $\sum_{e \in \delta(u)} x(e) \in \{0, 1\}$ for every $u \in V$. If $\sum_{e \in \delta(v)} x(e) = 0$, then restricting x to $E \setminus \delta(v)$ implies a feasible solution of LP($G - v$), whose objective value is equal to $\nu_f(G)$. Hence $\nu_f(G - v) \geq \nu_f(G)$, and thus we can assume that $\sum_{e \in \delta(v)} x(e) = 1$. Then v has either two incident edges e, f with $x(e) = x(f) = 1/2$ or exactly one edge e in M such that $x(e) = 1$. We shall consider both cases separately.

First assume that v has two incident edges e, f with $x(e) = x(f) = 1/2$. By the structure of x, the vertex v is contained in some odd M-alternating cycle C such that $x(e) = 1/2$ for every edge e in EC, where EC is the edge set of C. The cycle C has a matching M_C of size $(|EC| - 1)/2$ that exposes v. Define a vector x' in $\mathbb{R}^E_+$ by

$$x'(e) = \begin{cases} 1 & \text{if } e \in EC \cap M_C, \\ 0 & \text{if } e \in EC \setminus M_C, \\ x(e) & \text{otherwise.} \end{cases}$$

Then, x' is feasible for LP(G), and $\sum_{e \in E} x'(e) = \nu_f(G) - 1/2$. Since restricting x' to $E \setminus \delta(v)$ implies a feasible solution of LP($G - v$) and $\sum_{e \in \delta(v)} x'(e) = 0$, the objective value with respect to LP($G - v$) is $\nu_f(G) - 1/2$. Thus, we have $\nu_f(G - v) \geq \nu_f(G) - 1/2$.

The remaining case is when v has exactly one edge e in M such that $x(e) = 1$. Since $v \in B$, there is a maximum-size matching M_v that exposes v in G. The symmetric difference $M \Delta M_v$ has an M-alternating path P of even length from v to a vertex $w \notin \partial M$. Let $C_1, \ldots, C_p$ be the M-alternating cycles as defined in Proposition 4. Recall that $x(e) = 1/2$ if and only if e is contained in some C_i.

First suppose that P intersects with no C_i's. Then $x(e) \in \{0, 1\}$ for each $e \in EP$, where EP is the edge set of P. Define a vector x' in $\mathbb{R}^E_+$ by

$$x'(e) = \begin{cases} 0 & \text{if } e \in EP \cap M, \\ 1 & \text{if } e \in EP \setminus M, \\ x(e) & \text{otherwise.} \end{cases} \tag{11}$$

Then, x' is a feasible solution for LP(G). Moreover, since P is of even length, $\sum_{e \in E} x'(e) = \sum_{e \in E} x(e) = \nu_f(G)$. Since restricting x' to $E \setminus \delta(v)$ implies a feasible solution of LP($G - v$) and $\sum_{e \in \delta(v)} x'(e) = 0$, the objective value with respect to LP($G - v$) is $\nu_f(G)$. Thus, we have $\nu_f(G - v) \geq \nu_f(G)$.

Next suppose that P intersects with some of C_i's. We may assume that P intersects with exactly one of C_i's, say C. Indeed, suppose that we traverse P from v and consider when P intersects with some C_i for the first time. Then the union of C_i and the subpath P from v to C_i contains an M-alternating path of even length from v to the exposed vertex in C_i, and we may replace P with the path. Define a vector x' in $\mathbb{R}^E_+$ by

$$x'(e) = \begin{cases} 0 & \text{if } e \in (EP \cap M) \cup (EC' \setminus M), \\ 1 & \text{if } e \in (EP \setminus M) \cup (EC' \cap M), \\ x(e) & \text{otherwise,} \end{cases} \tag{12}$$

where EC is the edge set in C and $EC' = EC \setminus EP$. Then, x' is a feasible solution for LP(G). Moreover, since $\sum_{e \in EP \cup EC} x'(e) = \sum_{e \in EP \cup EC} x(e) - 1/2$, it holds that $\sum_{e \in E} x'(e) = \nu_f(G) - 1/2$. Since restricting x' to $E \setminus \delta(v)$

implies a feasible solution of $\mathrm{LP}(G-v)$ and $\sum_{e\in\delta(v)} x'(e) = 0$, the objective value with respect to $\mathrm{LP}(G-v)$ is $\nu_f(G) - 1/2$. Thus, we have $\nu_f(G-v) \geq \nu_f(G) - 1/2$. Therefore, in either case, it holds that $\nu_f(G-v) \geq \nu_f(G) - 1/2$. □

Proof of Theorem 4. We first claim that it is necessary to remove at least $d(G)$ vertices. Assume that we are given a vertex-removal stabilizer $X = \{x_1, x_2, \ldots, x_p\}$ of size p. Define $X_i := \{x_1, x_2, \ldots, x_i\}$ for each $i = 1, 2, \ldots, p$ and $X_0 := \emptyset$. Then, by Lemma 3, $d(G) - d(G-X) = \sum_{i=1}^{p} (d(G-X_{i-1}) - d(G-X_i)) \leq p/2$. Since $d(G-X) = 0$, we have $p \geq 2d(G)$.

Let M be a B_1-optimal matching in G. Then $G[B_3]$ has $2d(G)$ vertices exposed by M, the set of which is denoted by X^*. The size of X^* is $2d(G)$, and $\nu(G-X^*) = \nu(G)$. We will claim that X^* is a vertex-removal stabilizer.

By Proposition 4, $\mathrm{DP}(G)$ has a half-integral dual optimal solution y satisfying that $y(v) = 1/2$ for every vertex v in B_3. The vector obtained by restricting y to $V \setminus X^*$ is a feasible solution for $\mathrm{DP}(G-X^*)$, whose objective value is

$$\tau_f(G) - \frac{1}{2}|X^*| = \nu_f(G) - \frac{1}{2}|X^*| = \nu_f(G) - d(G).$$

Hence, we have $\tau_f(G-X^*) \leq \nu_f(G) - d(G)$. Since $\nu_f(G-X^*) = \tau_f(G-X^*)$, it holds that $d(G-X^*) = \nu_f(G-X^*) - \nu(G-X^*) \leq (\nu_f(G) - d(G)) - \nu(G) = 0$. This means that X^* is a vertex-removal stabilizer of size $d(G)$, and thus it is minimum. □

Theorem 4 naturally leads to a polynomial-time algorithm for finding a minimum-size vertex-removal stabilizer. The algorithm finds a B_1-optimal matching M in G, and return the set of vertices in $G[B_3]$ exposed by M. By Proposition 3, a B_1-optimal matching in G can be found in polynomial time. Therefore, we have the following theorem.

Theorem 5. *We can find a minimum-size vertex-removal stabilizer in polynomial time.* □

5. VERTEX ADDITION

For each set X of vertices, we denote by $G+X$ the graph obtained from G by adding X with some edges from X. Note that $G+X$ is not uniquely determined, but the arguments below hold for any graph obtained by the vertex addition. When $X = \{v\}$ we denote $G+v$ for simplicity. A *vertex-addition stabilizer* is a set X of vertices such that the graph obtained by adding X with some edges from X is stable.

Similarly to Sections 3 and 4, we have the following lemma for adding one vertex. We omit the proof due to space limitation.

Lemma 4. *For every vertex v, $d(G) - d(G+v) \leq 1/2$.*

With Lemma 4, we obtain a proposition similar to Theorem 4, which shows that the size of a vertex-addition stabilizer is at least $2d(G)$. Furthermore, a vertex-addition stabilizer of size $2d(G)$ can be found as follows: Look at the set X of $2d(G)$ vertices in $G[B_3]$ exposed by a B_1-optimal matching in G as in Theorem 4, and we add $2d(G)$ new vertices and connect each to a different vertex of X with a single edge. Then the resulting graph turns out to be stable.

Theorem 6. *The minimum size of a vertex-addition stabilizer is equal to $2d(G)$. Moreover, we can find a minimum-size vertex-addition stabilizer in polynomial time.* □

It should be noted that our algorithm adds $2d(G)$ edges, which is optimal even when we aim to minimize the total number of appended vertices and edges.

6. HARDNESS FOR WEIGHTED CASES

6.1 Edge addition

In this section we consider the weighted variant of finding an edge-addition stabilizer. Precisely speaking, in this problem, we are given a weight function $w\colon \binom{V}{2} \setminus E \to \mathbb{R}_+$, and we aim to find a minimum-weight edge-addition stabilizer, i.e., an edge-addition stabilizer F minimizing $\sum_{e\in F} w(e)$ among all edge-addition stabilizers.

Theorem 7. *The problem of finding a minimum-weight edge-addition stabilizer is* **NP***-hard.*

Proof. We reduce an **NP**-hard problem called EXACT COVER BY 3-SETS (X3C for short) to the problem of finding a minimum-weight edge-addition stabilizer. The problem X3C is defined as follows (see [16]).

Input. A set X with $|X| = 3n$ and a collection $\mathcal{C}$ of 3-element subsets of X.

Goal. Determine whether there is a subcollection $\mathcal{C}' \subseteq \mathcal{C}$ of size n such that every element of X occurs in exactly one member of $\mathcal{C}'$

Suppose we are given an instance of the X3C, i.e., we are given a set X and a collection $\mathcal{C}$ of 3-element subsets of X. In what follows, we construct an instance of the problem of finding a minimum-weight edge-addition stabilizer that is equivalent to the original instance of the X3C. For each element x in X, we introduce three vertices $v(x,0), v(x,1)$, and $v(x,2)$ and three edges between each pair of them. For each member $C = \{x_1, x_2, x_3\}$ in $\mathcal{C}$, we introduce nine vertices $v(C, x_i)$ for each $i = 1,2,3$ and $v(C, x_i, x_j)$ for each pair of $i, j = 1,2,3$ with $i \neq j$ and six edges between $v(C, x_i)$ and $v(C, x_i, x_j)$ for each $i, j = 1,2,3$ with $i \neq j$. The obtained graph is denoted by $G = (V, E)$. We define a weight function $w\colon \binom{V}{2} \setminus E \to \mathbb{R}_+$ as follows. For each member C in $\mathcal{C}$ and each element x in C, set $w(v(x,0), v(C,x)) = 3n + 1 (=: L)$. For each member C in $\mathcal{C}$ and each pair of elements x, y in C, set $w(v(C,x,y), v(C,y,x)) = 1$. We define the weight of the other pairs as a sufficiently large number (e.g., $nL + 3n + 1$).

We now show that there is an edge-addition stabilizer F such that $\sum_{e\in F} w(e) \leq nL + 3n$ if and only if the original instance of the X3C has a solution.

First, assume that the original instance of the X3C has a solution $\mathcal{C}' \subseteq \mathcal{C}$. Then,

$$\begin{aligned} F := &\{\{v(x,0), v(C,x)\} \mid x \in C \in \mathcal{C}'\} \\ &\cup \{\{v(C,x,y), v(C,y,x)\} \mid x, y \in C \in \mathcal{C}'\} \end{aligned}$$

satisfies that $\nu(G+F) = \nu_f(G+F)$ and $\sum_{e\in F} w(e) = nL + 3n$, which shows the sufficiency of the claim.

Next, to show the necessity, assume that there is an edge-addition stabilizer F such that $\sum_{e\in F} w(e) \leq nL + 3n$. For every element x in X, since $v(x,0), v(x,1)$, and $v(x,2)$ form a triangle in G, which is not stable, at least one edge in F is incident to $v(x,0)$. Furthermore, since $\sum_{e\in F} w(e) \leq nL + 3n$, for every element x in X, there is a unique member $C = \{x, y, z\}$ in $\mathcal{C}$ such that $\{v(x,0), v(C,x)\} \in F$. Consider the connected component of $G+F$ that contains $v(x,0), v(x,1), v(x,2), v(C,x), v(C,x,y)$ and $v(C,x,z)$, where $C = \{x, y, z\} \in \mathcal{C}$ and $\{v(x, 0), v(C, x)\} \in F$. Since this component has to be stable, we can easily check that F contains all of the edges in

$$E_C := \{\{v(C, x, y), v(C, y, x)\},\\ \{v(C, x, z), v(C, z, x)\}, \{v(C, y, z), v(C, z, y)\}\}.$$

This shows that $\mathcal{C}' := \{C \in \mathcal{C} \mid E_C \subseteq F\}$ covers X in the original instance of the X3C. On the other hand, since $\sum_{e \in F} w(e) \leq nL + 3n$ and F contains n edges of weight L, we have $|\mathcal{C}'| = |\{C \in \mathcal{C} \mid E_C \subseteq F\}| \leq 3n/3 = n$. Thus, each element of X occurs in exactly one member of $\mathcal{C}'$, which means that $\mathcal{C}'$ is a solution of the the original instance of the X3C.

Therefore, we can reduce the X3C to the problem of finding a minimum-weight edge-addition stabilizer, which completes the proof. □

6.2 Vertex removal

In this section we consider the weighted variant of finding a vertex-removal stabilizer. Precisely speaking, in this problem, we are given a weight function $w\colon V \to \mathbb{R}_+$, and we aim to find a minimum-weight vertex-removal stabilizer, i.e., a vertex-removal stabilizer X minimizing $\sum_{v \in X} w(v)$ among all vertex-removal stabilizers. We show that the problem of finding a minimum-weight vertex-removal stabilizer is also **NP**-hard.

THEOREM 8. *The problem of finding a minimum-weight vertex-removal stabilizer is* **NP***-hard.*

We reduce the clique problem (CLIQUE), which is defined as follows.

Input. A graph G' and a positive integer p.

Goal. Determine whether there exists a set F of $p(p-1)/2$ edges of G' such that $|\partial F|$ is at most p.

CLIQUE is known to be **NP**-hard [16]. A set of p vertices covered by $p(p-1)/2$ edges is called a *clique* of size p.

For a bipartite graph $H = (U_1, U_2; F)$ with $|U_1| \geq |U_2|$, a U_2*-perfect matching* is a matching of size $|U_2|$. That is, it is a matching covering all the vertices of U_2.

LEMMA 5. *The following problem is* **NP***-complete.*

Input. *A bipartite graph $H = (U_1, U_2; F)$ that has a U_2-perfect matching, and two positive integers p, q.*

Goal. *Determine whether there exists a set $X \subseteq U_1$ of size at most p such that $\nu(G - X) \leq |U_2| - q$.*

PROOF. Suppose that we are given an instance of CLIQUE, a graph $G' = (V', E')$ and a positive integer p. We construct a bipartite graph H as follows. Define $U_1' := V'$ and $U_2 := E'$, and there is an edge between a vertex v in U_1' and a vertex e in U_2 if and only if v is an end vertex of e in G' (i.e., $v \in e$). Furthermore, defining $S_e := \{s_e^1, \ldots, s_e^L\}$ for each edge e in E', where $L > |V'|$, we connect a vertex e of U_2 to each of S_e with L edges. In summary, the resulting graph $H = (U_1, U_2; F)$ is defined by

$$U_1 := U_1' \cup \bigcup_{e \in E} S_e, \qquad U_2 := E',$$
$$F := \{\{v, e\} \mid v \in e\} \cup \{\{s_e^j, e\} \mid j = 1, \ldots, L, e \in E'\}.$$

Set $q := p(p-1)/2$. We will show that H has a desired vertex set of size at most $Lp(p-1)/2 + p$ if and only if G' has a clique of size p.

Suppose that G' has a clique K of size p. Then we delete $X = VK \cup \bigcup_{e \in EK} S_e$ from U_1, where VK (resp., EK) is the vertex set (resp., the edge set) of K. The number of the deleted vertices is $Lp(p-1)/2 + p$. Since each vertex in U_2 corresponding to one in EK has no neighbor in $G' - X$, $G' - X$ has no matching of size greater than $|U_2| - |EK| = |U_2| - q$. Thus X is a solution of the problem.

Conversely, suppose that H has a solution set X of size at most $Lp(p-1)/2 + p$. We may assume that X is minimal. Then we see by the minimality that, in $H - X$, each vertex e in U_2 has either no single edges to S_e or all the single edges remaining. Hence exactly $p(p-1)/2$ vertices of U_2's have no neighbor to S_e's in $H - X$. The edge subset corresponding to these vertices is denoted by J. Since a vertex e in $U_2 \setminus J$ has an edge to S_e, $H - X$ has a matching of size at least $|E'| - p(p-1)/2$. Therefore, a vertex in J has to have no edge to U_1' in $H - X$. Hence $N(J) \cap U_1' \subseteq X$ holds, which implies that $|N(J) \cap U_1'| \leq p$. Since $|N(J) \cap U_1'| \geq p$ as $|J| = p(p-1)/2$, the number is exactly equal to p. This means that J forms a clique of size p in G'. □

PROOF OF THEOREM 8. We reduce the problem in Lemma 5. Suppose that we are given an instance of the problem, a bipartite graph $H = (U_1, U_2; F)$ and two positive integers p, q.

We prepare q triangles, $T_1, T_2, \ldots, T_q$, and append T_i to all the vertices of U_2, that is, $N(T_i) = U_2$. The resulting graph is denoted by $G = (V, E)$. Note that the Gallai–Edmonds decomposition of G is represented by $B_1 = U_1$, $B_3 = \bigcup_{i=1}^{q} VT_i$, $A = U_2$, and $D = \emptyset$, where VT_i is the vertex set of T_i. Since H has a U_2-perfect matching, G has a B_1-optimal matching that exposes all the triangles.

We define the weight function $w\colon V \to \mathbb{R}_+$ by $w(v) = 1$ if $v \in U_1$, and $w(v) = \infty$ otherwise (i.e., $w(v) \geq |V|$).

It is observed that G has a vertex-removal stabilizer of finite weight if and only if H has a solution of size p. Thus the statement holds. □

REFERENCES

[1] S. Ahmadian, L. Sanita, and H. Hosseinzadeh. Stabilizing network bargaining games by blocking players. In *Integer Programming and Combinatorial Optimization - 18th International Conference, IPCO 2016, Liége, Belgium, June 1-3, 2016. Proceedings*, 2016. to appear.

[2] H. Aziz and B. de Keijzer. Shapley meets shapley. In E. W. Mayr and N. Portier, editors, *31st International Symposium on Theoretical Aspects of Computer Science (STACS 2014), STACS 2014, March 5-8, 2014, Lyon, France*, volume 25 of *LIPIcs*, pages 99–111. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2014.

[3] Y. Bachrach, E. Elkind, R. Meir, D. V. Pasechnik, M. Zuckerman, J. Rothe, and J. S. Rosenschein. The cost of stability in coalitional games. In M. Mavronicolas and V. G. Papadopoulou, editors, *Algorithmic Game Theory, Second International Symposium, SAGT 2009, Paphos, Cyprus, October 18-20, 2009. Proceedings*, volume 5814 of *Lecture*

Notes in Computer Science, pages 122–134. Springer, 2009.

[4] M. Bateni, M. Hajiaghayi, N. Immorlica, and H. Mahini. The cooperative game theory foundations of network bargaining games. In S. Abramsky, C. Gavoille, C. Kirchner, F. M. auf der Heide, and P. G. Spirakis, editors, *Automata, Languages and Programming, 37th International Colloquium, ICALP 2010, Bordeaux, France, July 6-10, 2010, Proceedings, Part I*, volume 6198 of *Lecture Notes in Computer Science*, pages 67–78. Springer, 2010.

[5] C. Berge. Sur le couplage maximum d'un graphe. *Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences [Paris]*, 247:258–259, 1958.

[6] P. Biró, M. Bomhoff, P. A. Golovach, W. Kern, and D. Paulusma. Solutions for the stable roommates problem with payments. *Theoretical Computer Science*, 540:53–61, 2014.

[7] P. Biró, W. Kern, and D. Paulusma. Computing solutions for matching games. *International Journal of Game Theory*, 41(1):75–90, 2012.

[8] A. Bock, K. Chandrasekaran, J. Könemann, B. Peis, and L. Sanità. Finding small stabilizers for unstable graphs. *Mathematical Programming, Series B*, 154(1-2):173–196, 2015.

[9] N. Chen, P. Lu, and H. Zhang. Computing the nucleolus of matching, cover and clique games. In J. Hoffmann and B. Selman, editors, *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada.* AAAI Press, 2012.

[10] X. Deng, T. Ibaraki, and H. Nagamochi. Algorithmic aspects of the core of combinatorial optimization games. *Mathematics of Operations Research*, 24(3):751–766, 1999.

[11] J. Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.

[12] K. Eriksson and J. Karlander. Stable outcomes of the roommate game with transferable utility. *International Journal of Game Theory*, 29(4):555–569, 2001.

[13] U. Faigle, W. Kern, S. P. Fekete, and W. Hochstättler. The nucleon of cooperative games and an algorithm for matching games. *Mathematical Programming*, 83:195–211, 1998.

[14] T. Gallai. Kritische graphen II. *A Magyar Tudományos Akadémia – Matematikai Kutató Intézetének Közleményei*, 8:373–395, 1963.

[15] T. Gallai. Maximale systeme unabhänginger kanten. *A Magyar Tudományos Akadémia – Matematikai Kutató Intézetének Közleményei*, 9:401–413, 1964.

[16] M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.

[17] W. Hochstättler, H. Jin, and R. Nickel. Note on an auction procedure for a matching game in polynomial time. In S. Cheng and C. K. Poon, editors, *Algorithmic Aspects in Information and Management, Second International Conference, AAIM 2006, Hong Kong, China, June 20-22, 2006, Proceedings*, volume 4041 of *Lecture Notes in Computer Science*, pages 387–394. Springer, 2006.

[18] W. Kern and D. Paulusma. Matching games: The least core and the nucleolus. *Mathematics of Operations Research*, 28(2):294–308, 2003.

[19] S. Khot. On the power of unique 2-prover 1-round games. In J. H. Reif, editor, *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 767–775. ACM, 2002.

[20] J. M. Kleinberg and É. Tardos. Balanced outcomes in social exchange networks. In C. Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 295–304. ACM, 2008.

[21] J. Könemann, K. Larson, and D. Steiner. Network bargaining: Using approximate blocking sets to stabilize unstable instances. *Theory of Computing Systems*, 57(3):655–672, 2015.

[22] Y. K. Kwon and P. L. Yu. Stabilization through taxation in n-person games. *Journal of Optimization Theory and Applications*, 23(2):277–284, 1977.

[23] L. Lovász and M. D. Plummer. *Matching Theory*, volume 29 of *Annals of Discrete Mathematics*. North-Holland, Amsterdam, 1986.

[24] J. F. Nash. The bargaining problem. *Econometrica*, 18(2):155–162, 1950.

[25] W. Pulleyblank. Fractional matchings and the Edmonds-Gallai theorem. *Discrete Applied Mathematics*, 16(1):51 – 58, 1987.

[26] L. Shapley and M. Shubik. The assignment game I: The core. *International Journal of Game Theory*, 1(1):111–130, 1971.

[27] W. T. Tutte. The factorization of linear graphs. *The Journal of the London Mathematical Society*, 22:107–111, 1947.

[28] Y. Zick, M. Polukarov, and N. R. Jennings. Taxation and stability in cooperative games. In M. L. Gini, O. Shehory, T. Ito, and C. M. Jonker, editors, *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS 2013, Saint Paul, MN, USA, May 6-10, 2013*, pages 523–530. IFAAMAS, 2013.