Stabilization of Capacitated Matching Games

Matthew Gerstbrein[1], Laura Sanità[2], and Lucy Verberk(✉)[3]

[1] University of Waterloo, Canada.
mlgerstbrein@uwaterloo.ca
[2] Bocconi University of Milan, Italy.
laura.sanita@unibocconi.it
[3] Eindhoven University of Technology, Netherlands.
l.p.a.verberk@tue.nl

Abstract

An edge-weighted, vertex-capacitated graph G is called *stable* if the value of a maximum-weight capacity-matching equals the value of a maximum-weight *fractional* capacity-matching. Stable graphs play a key role in characterizing the existence of stable solutions for popular combinatorial games that involve the structure of matchings in graphs, such as network bargaining games and cooperative matching games.

The vertex-stabilizer problem asks to compute a minimum number of players to block (i.e., vertices of G to remove) in order to ensure stability for such games. The problem has been shown to be solvable in polynomial-time, for unit-capacity graphs. This stays true also if we impose the restriction that the set of players to block must not intersect with a given specified maximum matching of G.

In this work, we investigate these algorithmic problems in the more general setting of arbitrary capacities. We show that the vertex-stabilizer problem with the additional restriction of avoiding a given maximum matching remains polynomial-time solvable. Differently, without this restriction, the vertex-stabilizer problem becomes NP-hard and even hard to approximate, in contrast to the unit-capacity case.

Finally, in unit-capacity graphs there is an equivalence between the stability of a graph, existence of a stable solution for network bargaining games, and existence of a stable solution for cooperative matching games. We show that this equivalence does not extend to the capacitated case.

Keywords: Matching · Game theory · Network bargaining.

1 Introduction

Network Bargaining Games (NBG) and *Cooperative Matching Games* (CMG) are popular combinatorial games involving the structure of matchings in graphs. CMG were introduced in the seminal paper of Shapley and Shubik 50 years ago [17], and have been widely studied since then. NBG are relatively more recent, and were defined by Kleinberg and Tardos [13] as a generalization of Nash's 2-player bargaining solution [16].

Instances of these games are described by a graph $G = (V, E)$ with weights $w \in \mathbb{R}^E_{\geq 0}$, where the vertices and the edges model the players and their potential interactions, respectively. The value of a *maximum-weight matching*, denoted as $\nu(G)$, is the total value that players can collectively accumulate. The goal, roughly speaking, is to assign values to players in such a way that players have no incentive to deviate from the current allocation.

Formally, in an instance of a NBG, players want to enter in a deal with one of their neighbours, and agree on how to split the value of the deal given by the weight of the corresponding edge. Hence, an outcome is naturally associated with a matching M of G representing the deals, and allocation vector $y \in \mathbb{R}^V_{\geq 0}$ with $y_u + y_v = w_{uv}$ if $uv \in M$, and $y_v = 0$ if v is not matched. An outcome (M, y) is *stable* if each player's allocation y_u is at least as large as their *outside option*, formally defined as $\max_{v:uv \in E \setminus M} \{w_{uv} - y_v\}$.

In an instance of a CMG, one wants to find an allocation of total value $\nu(G)$, given by a vector $y \in \mathbb{R}^V_{\geq 0}$ in which no subset of players can gain more by forming a coalition. This condition is enforced by the constraint $\sum_{v \in S} y_v \geq \nu(G[S])$ for all $S \subseteq V$, where $G[S]$ indicates the subgraph of G induced by the vertices in S. Such an allocation is called *stable*, and the set of stable allocations constitutes the *core* of the game.

Despite having been defined in different contexts, there is a tight link between stable solutions of these types of games. In particular, if each game is played on the same graph G, then it has been shown that either a stable solution exists for both games, or for neither game. This follows as both games admit the same polyhedral characterization of instances with stable solutions [7, 13]. Specifically, a stable solution exists if and only if $\nu(G)$ equals the value of the standard linear programming (LP) relaxation of the maximum matching problem defined as

$$\nu_f(G) := \max \left\{ w^\top x : \sum_{u:uv \in E} x_{uv} \leq 1 \; \forall v \in V, \; x \geq 0 \right\}. \tag{1}$$

A graph G for which $\nu(G) = \nu_f(G)$ is called *stable*. As a result of this characterization, it is easy to see that there are graphs which do not admit stable solutions (to either type of game), such as odd cycles. Given that not all graphs are stable, naturally arises the *stabilization* problem of how to minimally modify a graph to turn it into a stable one. Stabilization problems attracted a lot of attention in the literature in the past years (see e.g. [1, 3–6, 12, 14, 15]).

In this context, very natural operations to stabilize graphs are edge- and vertex-removal operations. Those have an interesting interpretation: they correspond to blocking interactions and players, respectively, in order to ensure a stable outcome. While removing a minimum number of edges to stabilize a graph is NP-hard already for unit weight graphs [4], and even hard-to-approximate with a constant factor [10, 14], stabilizing the graph via vertex-removal operations turned out to be solvable in polynomial-time. Specifically, [1] and [12] showed that computing a minimum-cardinality set of players to block in order to stabilize an unweighted graph (called the *vertex-stabilizer problem*) can be done in polynomial time. Furthermore, [1] showed that computing a minimum set of players to block in order to make a given maximum matching realizable as a stable outcome (called the *M-vertex-stabilizer problem*) is also efficiently solvable. The authors of [14] showed that both results generalize to weighted graphs.

This paper focuses on *Capacitated NBG*, introduced by Bateni et al [2] as a generalization of NBG, to capture the more realistic scenario where players are allowed to enter in more than one deal. This generalization can be modeled by allowing for vertex capacities $c \in \mathbb{Z}^V_{\geq 0}$. The notion of a matching is therefore generalized to a *c-matching*, where each vertex $v \in V$ is matched with at most c_v vertices. In this case, the value of a maximum-weight c-matching of a graph G is denoted as $\nu^c(G)$, and the standard LP relaxation is given by

$$\nu^c_f(G) := \max \left\{ w^\top x : \sum_{u:uv \in E} x_{uv} \leq c_v \; \forall v \in V, \; 0 \leq x \leq 1 \right\}. \tag{2}$$

Similarly to the unit-capacity case, an outcome to the NBG is associated with a c-matching M and a vector $a \in \mathbb{R}^{2E}_{\geq 0}$ that satisfies $a_{uv} + a_{vu} = w_{uv}$ if $uv \in M$, and $a_{uv} = a_{vu} = 0$ otherwise. The concepts of outside option and stable outcome can be defined similarly as in the unit-capacity case, see [2].

The authors of [2] proved that the LP characterization of stable solutions generalize, i.e., there exist a stable outcome for the capacitated NBG on G if and only if $\nu^c(G) = \nu^c_f(G)$ (i.e., G is *stable*). Farczadi et al [9] show that some other important properties of NBG extend to this capacitated generalization, such as the possibility to efficiently compute a so-called *balanced* solution (we refer to [9] for details).

The goal of this paper is to investigate whether the other two significant features of NBG mentioned before generalize to the capacitated setting. Namely:

(i) Can one still efficiently stabilize instances via vertex-removal operations?

(ii) Does the equivalence between existence of stable allocations for capacitated CMG and existence of stable solutions for capacitated NBG still hold?

Our Results. In this paper we provide an answer to the above questions.

We investigate the M-vertex-stabilizer problem and the vertex-stabilizer problem in the capacitated setting in sections 3 and 4, respectively. While for unit-capacity graphs both problems are efficiently solvable, we show that adding capacities makes the complexity status of the vertex-stabilizer problem diverge. In particular, we prove that the vertex-stabilizer problem is NP-complete, and no $n^{1-\varepsilon}$-approximation is possible, for any $\varepsilon > 0$, unless P=NP. Note that a trivial n-approximation algorithm can be easily developed.

In contrast, we show that the M-vertex-stabilizer problem is still polynomial-time solvable in the capacitated setting. Our results here extend those of [14] for unit-capacity graphs, and builds upon an auxiliary construction of [9].

Finally, in section 5 we show that the equivalence between stability of a graph, existence of a stable allocation for CMG and existence of a stable outcome for NBG does *not* extend in the capacitated setting. In particular, we provide an unstable graph which does attain a stable allocation for the capacitated CMG[1].

2 Preliminaries and Notation

Problem definition. A set $S \subseteq V$ is called a **vertex-stabilizer** if $G \setminus S$ is stable, where $G \setminus S$ is the subgraph induced by the vertices $V \setminus S$. We say that a vertex-stabilizer S *preserves* a matching M of G if M is a matching in $G \setminus S$.

We now formally define the stabilization problems considered in this paper.

Vertex-stabilizer problem: given $G = (V, E)$ with edge weights $w \in \mathbb{R}^E_{\geq 0}$ and vertex capacities $c \in \mathbb{Z}^V_{\geq 0}$, find a vertex-stabilizer of minimum cardinality.

M-vertex-stabilizer problem: given $G = (V, E)$ with edge weights $w \in \mathbb{R}^E_{\geq 0}$, vertex capacities $c \in \mathbb{Z}^V_{\geq 0}$, and a maximum-weight c-matching M, find a vertex-stabilizer of minimum cardinality among the ones preserving M.

An instance of the vertex-stabilizer problem is stable if G is stable. An instance of the M-vertex-stabilizer problem is stable if G is stable, and M is a maximum-weight c-matching in G. Without loss of generality, we can assume that c_v is bounded by the degree of $v \in V$.

Notation. We refer to a graph with edge weights and vertex capacities as (G, w, c), and to a graph with a c-matching M as $[(G, w, c), M]$. For a vertex v, we let $\delta(v)$ be the set of edges of G incident into it, we let $N(v)$ be the set of its adjacent neighbours, and $N^+(v) = N(v) \cup \{v\}$. For $F \subseteq E$, we denote by d^F_v the degree of v in G with respect to the edges in F. We define $w(F) := \sum_{e \in F} w_e$. Given a c-matching M, we say that $v \in V$ is *exposed* if $d^M_v = 0$, *covered* if $d^M_v > 0$, *unsaturated* if $d^M_v < c_v$ and *saturated* if $d^M_v = c_v$. We also use these terms for feasible solutions x of (2), called *fractional c-matchings*, e.g., v is exposed if $\sum_{e \in \delta(v)} x_e = 0$. We let $n := |V|$, and $\triangle$ denote the symmetric difference operator.

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

Duality and augmenting structures. The dual of (2) is given by

$$\tau^c_f(G) := \min \left\{ c^\top y + \mathbf{1}^\top z : y_u + y_v + z_{uv} \geq w_{uv} \forall uv \in E, y \geq 0, z \geq 0 \right\}, \tag{3}$$

where $\mathbf{1}$ is the all-one vector of appropriate size. A solution (y, z) feasible for (3) is called a *fractional vertex cover*. By LP theory, we have $\nu^c(G) \leq \nu^c_f(G) = \tau^c_f(G)$.

Definition 1. We say that a walk W is M-alternating (w.r.t. a matching M) if it alternates edges in M and edges not in M. We say W is M-augmenting if it is M-alternating and $w(W \setminus M) > w(W \cap M)$. An M-alternating uv-walk W is *proper* if $d^{W \triangle M}_u \leq c_u$ and $d^{W \triangle M}_v \leq c_v$.

[1] It is stated in [8] (theorem 2.3.9) that a stable allocation for capacitated CMG exists iff G is stable, but our example shows this statement is not correct.

Definition 2. Given an M-alternating walk $W = (u; e_1, \ldots, e_k; v)$ and an $\varepsilon > 0$, the *ε-augmentation* of W is the vector $x^{M/W}(\varepsilon) \in \mathbb{R}^E$ given by

$$x_e^{M/W}(\varepsilon) = \begin{cases} 1 - \kappa(e)\varepsilon & \text{if } e \in M, \\ \kappa(e)\varepsilon & \text{if } e \notin M, \end{cases} \tag{4}$$

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

Remark 1. A feasible M-alternating walk with distinct endpoints is proper.

Definition 3. An odd cycle $C = (v; e_1, \ldots, e_k; v)$ is called an *M-blossom* if it is M-alternating such that either e_1 and e_k are both in M, or are both not in M. The vertex v is called the *base* of the blossom.

Definition 4. An *M-flower* $C \cup P$ consists of an M-blossom C with base u and an M-alternating path $P = (u; e_1, \ldots, e_k; v)$ such that (P, C, P^{-1}) is M-alternating and feasible. The vertex v is called the *root* of the flower. The flower is *M-augmenting* if

$$w(C \setminus M) + 2w(P \setminus M) > w(C \cap M) + 2w(P \cap M). \tag{5}$$

Definition 5. An *M-bi-cycle* $C \cup P \cup D$ consists of two M-blossoms C and D with bases u and v, respectively, and an M-alternating path $P = (u; e_1, \ldots, e_k; v)$ such that (P, D, P^{-1}, C) is M-alternating. The bi-cycle is *M-augmenting* if

$$w(C \setminus M) + 2w(P \setminus M) + w(D \setminus M) > w(C \cap M) + 2w(P \cap M) + w(D \cap M). \tag{6}$$

Note that, in the last two definitions, it may happen that P has no edges. In the unit-capacity case it is well-known that a matching M is maximum-weight if and only if there do not exist any proper M-augmenting paths or cycles. This generalizes to the capacitated case. We report a proof for completeness.

Theorem 1. *A c-matching M in (G, w, c) is maximum-weight if and only if G does not contain a proper M-augmenting trail.*

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

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

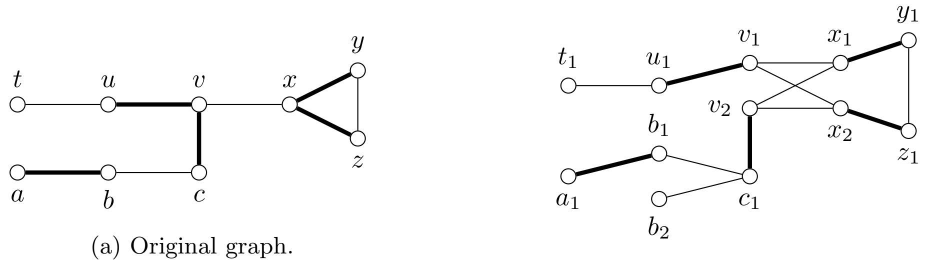
1. For each $v \in V$, define $b_v := \max \left\{ d_v^{M \setminus N}, d_v^{N \setminus M} \right\}$, create copies $v_1, \ldots, v_{b_v}$ and add them to $V(\hat{G})$. Initialize $J_M(v) = J_N(v) = \{1, \ldots, b_v\}$.

2. For each $uv \in M \setminus N$, add a single edge $u_i v_j$ to both $E(\hat{G})$ and $\hat{M}$ with edge-weight w_{uv}, where $i \in J_M(u)$ and $j \in J_M(v)$ are chosen arbitrarily. Remove i and j from $J_M(u)$ and $J_M(v)$, respectively.

3. Likewise for each $uv \in N \setminus M$.

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

Case 1: $\hat{T}$ is a proper $\hat{M}$-augmenting path. If $u = v$, then provided that at least one of e_1 and e_k is in M, then $d_u^{T \triangle M} \leq d_u^M \leq c_u$. If on the other hand neither of e_1 and e_k is in M, then the corresponding edges $\hat{e}_1$ and $\hat{e}_k$ in $\hat{T}$ are not in $\hat{M}$, and must therefore be in $\hat{N}$. Let u_i and u_j be the first



(a) Original graph.

(b) Auxiliary graph.

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

and last vertices of $\hat{T}$, incident with $\hat{e}_1$ and $\hat{e}_k$, respectively. Note that u_i and u_j are distinct, since $\hat{T}$ is a path. Furthermore, since $\hat{T}$ is proper, u_i and u_j are not incident with edges from $\hat{M}$. Observe that by construction of $\hat{G}$, either all vertices in $\{u_1, \ldots, u_{b_u}\}$ are $\hat{M}$-covered or $\hat{N}$-covered. Since u_i and u_j are not $\hat{M}$-covered, all copies of u must be $\hat{N}$-covered. Hence, $d_u^{M \setminus N} \leq d_u^{N \setminus M} - 2$, which means $d_u^M \leq d_u^N - 2$. Finally, $d_u^{T \triangle M} = d_u^M + 2 \leq d_u^N \leq c_u$.

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

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

Auxiliary Construction. We will use a construction given in [9], to transform a pair $[(G, w, c), M]$ into another one $[(G', w', \mathbf{1}), M']$, where G' is an auxiliary unit-capacity graph.

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

1. For each $v \in V$, create the set $C_v = \{v_1, \ldots, v_{c_v}\}$ of c_v copies of v, add C_v to $V(G')$, and initialize $J(v) = \{1, \ldots, c_v\}$.

2. For each $uv \in M$, add a single edge $u_i v_j$ to both $E(G')$ and M' with edge-weight w_{uv}, where $i \in J(u)$ and $j \in J(v)$ are chosen arbitrarily. Remove i and j from $J(u)$ and $J(v)$, respectively.

3. For each edge $uv \in E \setminus M$, add an edge $u_i v_j$ to $E(G')$ with edge-weight w_{uv}, for all $u_i \in C_u$ and $v_j \in C_v$.

See figure 1 for an example. In this figure it is easy to see that the matching M' in G' is not maximum, even though M is maximum in G.[2]

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

Remark 2. If $[(G, w, c), M]$ has auxiliary $[(G', w', \mathbf{1}), M']$, and $X \subseteq V$ is any set of vertices which avoids M, then $(G \setminus X)' = G' \setminus X'$, where $X' = \cup_{v \in X} C_v$.

The following easy lemma will be useful.

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

[2]It was stated in [9, corollary 1] that M is maximum if and only if M' is maximum, but this example shows this to be false.

Proof. Let $e_1 = uv$ and $e_2 = vw$ be two consecutive edges on P. Then $\eta(e_1)$ and $\eta(e_2)$ are the corresponding edges on $\eta(P)$, and they are both incident with $\eta(v)$. Hence, $\eta(P)$ is a walk.

For any edge e on P, we have $e \in M'$ if and only if $\eta(e) \in M$. In addition, $w'_e = w_{\eta(e)}$. So, $\eta(P)$ is an M-augmenting walk.

Suppose $P = (u; e_1, \ldots, e_k; v)$. Feasibility of P means that either $e_1 \in M'$, or u is M'-exposed. Likewise for e_k and v. It follows that either $\eta(e_1) \in M$, or $\eta(u)$ is M-unsaturated. Likewise for $\eta(e_k)$ and $\eta(v)$. This means $\eta(P)$ is feasible. $\square$

We will need the following theorem.

Theorem 2. *An M-vertex-stabilizer instance $[(G, w, c), M]$ is not stable if and only if the graph G' in the auxiliary construction $[(G', w', \mathbf{1}), M']$ contains at least one of the following: (i) an M'-augmenting flower; (ii) an M'-augmenting bi-cycle; (iii) a proper M'-augmenting path; (iv) an M'-augmenting cycle.*

Proof. It was proven in [9, theorem 2] that $[(G, w, c), M]$ does not correspond to a stable M-vertex-stabilizer instance if and only if $[(G', w', \mathbf{1}), M']$ does not correspond to a stable M'-vertex-stabilizer instance. We distinguish two scenarios for when the latter condition occurs. If M' is maximum-weight, then G' contains an M'-augmenting flower or bi-cycle, see [14, theorem 1]. If M' is not maximum-weight, G' must contain a proper M'-augmenting path or cycle, by standard matching theory. $\square$

We will refer to an augmenting structure of type $(i) - (iv)$ in theorem 2 as a *basic* augmenting structure. The next lemma follows from [14].

Lemma 2. *Let G' be a unit-capacity graph, and M' be any (not necessarily maximum) matching of G'.*

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

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

Proof. (a) When given a graph G', a matching M', a vertex u, and an integer k, algorithm 3 in [14] computes a feasible M'-augmenting uv-walk of length at most k, or determines none exist, for all $v \in V(G')$. Lemma 7 and 8 in [14] show correctness of the algorithm. The algorithm is polynomial time in k, $|V(G')|$, and $|E(G')|$. Since we use $k = 3\,|V(G')|$, it is polynomial time. As mentioned, algorithm 3 in [14] actually returns one uv-walk per $v \in V(G')$, if at least one exists for v. We just need one such walk, so if for at least one v a uv-walk is returned, we arbitrarily choose one, otherwise we know no such walk starting at u exists.

(b) Lemma 9 in [14] directly gives us that a feasible M'-augmenting uv-walk contains a feasible M'-augmenting uv-path, an M'-augmenting cycle, an M'-augmenting flower rooted at u or v, or an M'-augmenting bi-cycle. By remark 1 the path is proper if $u \neq v$. Lemma 9 in [14] is proven in a constructive way, hence it also gives a way to compute the augmenting structure in polynomial time. $\square$

The next theorem is standard.

Theorem 3. *An M-vertex-stabilizer instance $[(G, w, c), M]$ is stable if and only if G does not contain a feasible M-augmenting walk.*

Proof. We prove both directions by contraposition.

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

$$\nu_f^c(G) \geq w^\top x^{M/W}(\varepsilon) = w(M) - \varepsilon w(W \cap M) + \varepsilon w(W \setminus M) > w(M), \tag{7}$$

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

($\Leftarrow$) Assume the instance is not stable. Then by theorem 2, the graph G' from the auxiliary $[(G', w', \mathbf{1}), M']$ contains a basic augmenting structure, which clearly is a feasible M'-augmenting walk P. Then $\eta(P)$ is a feasible M-augmenting walk, by lemma 1. $\square$

3 M-vertex-stabilizer

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

Theorem 4. *The M-vertex-stabilizer problem on weighted, capacitated graphs can be solved in polynomial time.*

Overview of the strategy. A natural strategy would be to first apply the auxiliary construction described in section 2 to reduce to unit-capacity instances, and then apply the algorithm proposed in [14] which solves the problem exactly. However, there is a critical issue with this strategy. Namely, the auxiliary construction applied to unstable instances does *not* always preserve maximality of the corresponding matchings, as shown in figure 1. In that example, the matching M' is not maximum in G'. The algorithm of [14], if applied to an instance where the given matching is not maximum, is not guaranteed to find an optimal solution, but only a 2-approximate one (see theorem 12 in [14]). In addition, since the auxiliary construction splits a vertex into multiple ones, we may even get infeasible solutions. As a concrete example of this, the algorithm of [14] applied to the instance of figure 1b will include b_2 in its proposed solution. Mapping this solution to our capacitated instance would imply to remove b, which is clearly not allowed as b is M-covered.

To overtake this issue, we do not apply the algorithm of [14] as a black-box, but use parts of it (highlighted in lemma 2) in a careful way. In particular, we use it to compute a sequence of feasible augmenting walks in G'. We actually show that the walks in G' which might create the issue described before when mapped backed to G, are the walks in which at least one edge of G is traversed more than once in opposite directions, and that have two distinct endpoints. When this happens, we prove that we can modify the walk and get one where the endpoints coincide, which will still be feasible and augmenting. In this latter case, we can then either correctly identify a vertex to remove (the unique endpoint), or determine that the instance cannot be stabilized.

A more detailed description. We start by defining *ties*.

Definition 6. Given $[(G,w,c),M]$ with auxiliary $[(G',w',\mathbf{1}),M']$, and an M'-alternating path P', a *tie* in P' is a pair of unmatched edges $\{ab,cd\}$ on P' such that for some distinct $u,v \in V$, either $\{a,c\} \subseteq C_u$ and $\{b,d\} \subseteq C_v$ or $\{a,d\} \subseteq C_u$ and $\{b,c\} \subseteq C_v$. We say P' is *tieless* if it does not contain a tie.

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

Lemma 3. *Given $[(G,w,c),M]$ with auxiliary $[(G',w',\mathbf{1}),M']$, if M is a maximum-weight c-matching in G, then all proper M'-augmenting paths and cycles contain ties.*

Proof. We prove this by contraposition. So, suppose that there is a proper M'-augmenting path or cycle P' that is tieless. Note that P' is also feasible. By lemma 1, $P = \eta(P')$ is a feasible M-augmenting walk. Since P' is tieless, there is a bijection between $E(P')$ and $E(P)$, and so, as P' does not repeat edges, neither does P. Hence P is a feasible M-augmenting trail. We will show that P is proper.

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

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

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

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

In all cases P is a proper M-augmenting trail. It follows by theorem 1 that M is not a maximum-weight c-matching in G. ∎

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

Definition 7. Given $[(G, w, c), M]$ and an M-alternating walk $P = (u; e_1, \ldots, e_k; v)$ which repeats an edge in opposite directions, let t be the least index such that $e_t = e_s$ for some $s < t$, and e_s and e_t are traversed in opposite directions by P. Then the u-*traceback* and v-*traceback* of P are defined as the walks $\mathrm{tb}(P, u) = (e_1, \ldots, e_t, e_{s-1}, e_{s-2}, \ldots, e_1)$ and $\mathrm{tb}(P, v) = (e_k, e_{k-1} \ldots, e_s, e_{t+1}, e_{t+2}, \ldots, e_k)$.

The next lemma explains how to use the traceback operation.

Lemma 4. *Given $[(G, w, c), M]$ such that M is maximum-weight, and auxiliary $[(G', w', \mathbf{1}), M']$, let $P' = (u_i; e'_1, \ldots, e'_k; v_j)$ be a proper M'-augmenting path such that both u_i and v_j are M'-exposed and $\eta(u_i) \neq \eta(v_j)$. Then $tb(\eta(P'), \eta(u_i))$ and $tb(\eta(P'), \eta(v_j))$ are well-defined, feasible M-alternating walks, and at least one of them is M-augmenting.*

Proof. Let $P = \eta(P') = (u; e_1, \ldots, e_k, v)$. By lemma 1, P is a feasible M-augmenting walk, and even proper by remark 1, since $u \neq v$. To show that $\mathrm{tb}(P, u)$ and $\mathrm{tb}(P, v)$ are well-defined, we must show that P traverses some edge in opposite directions. By lemma 3 we already have that P' contains a tie, and hence that P traverses some edge twice. We now show that there must exist at least one edge that is traversed in opposite direction. Suppose not, let t be the least index such that $e_t = e_s$ for some $s < t$. Decompose P as $(P_1, e_s, P_2, e_t, P_3)$.

Claim 1. *If P traverses e_s and e_t in the same direction, then (P_1, e_s, P_3) is a shorter proper M-augmenting walk.*

Proof. For notation, define $P_2^+ = (P_2, e_t)$. By definition, P_2^+ is an M-alternating closed trail of even length. It follows that $d_v^{P_2^+ \triangle M} = d_v^M \leq c_v$ for all vertices v on P_2^+, and hence P_2^+ is proper. Since M is maximum-weight, theorem 1 implies that P_2^+ cannot be M-augmenting. However, P is M-augmenting, which means the augmenting part must come from $P \setminus P_2^+$. Hence, (P_1, e_s, P_3) is an M-augmenting walk. It is proper because P is proper. □

By this claim $W = (P_1, e_s, P_3)$ is a shorter proper M-augmenting walk. W also necessarily repeats an edge, because otherwise W is a proper M-augmenting trail, contradicting that M is maximum-weight, by theorem 1. Then we can apply the claim again, to find an even shorter proper M-augmenting walk. This argument can be repeated until eventually we reach a contradiction.

Thus there is at least one edge traversed in opposite direction, hence $\mathrm{tb}(P, u)$ and $\mathrm{tb}(P, v)$ are well-defined. Clearly $\mathrm{tb}(P, u)$ and $\mathrm{tb}(P, v)$ are M-alternating. Furthermore, since u_i and v_j are M'-exposed, u and v are M-unsaturated. It follows that $\mathrm{tb}(P, u)$ and $\mathrm{tb}(P, v)$ are feasible.

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

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

$$w(P_1 \setminus M) + w(P_2^{++} \setminus M) + w(P_3 \setminus M) > w(P_1 \cap M) + w(P_2^{++} \cap M) + w(P_3 \cap M). \tag{8}$$

Adding these inequalities, we obtain

$$w(P_u \setminus M) = 2w(P_1 \setminus M) + w(P_2^{++} \setminus M) > 2w(P_1 \cap M) + w(P_2^{++} \cap M) = w(P_u \cap M). \tag{9}$$

Hence, P_u is M-augmenting.

Case 2: $w(P_1 \setminus M) - w(P_3 \setminus M) < w(P_1 \cap M) - w(P_3 \cap M)$. Analogous to case 1, we find that P_v is M-augmenting.

Case 3: $w(P_1 \setminus M) - w(P_3 \setminus M) = w(P_1 \cap M) - w(P_3 \cap M)$. Analogous to case 1, we find that both P_u and P_v are M-augmenting. □

Algorithm 1: finding an M-vertex-stabilizer

input: $[(G, w, c), M]$

compute the auxiliary $[(G', w', \mathbf{1}), M']$

initialize $S \leftarrow \emptyset$, $L \leftarrow M'$-exposed vertices

while $L \neq \emptyset$ **do**

- select $u_i \in L$ and compute a feasible M'-augmenting walk starting at u_i using lemma 2(a)
- **if** *no such walk exists* **then**
  - $L \leftarrow L \setminus \{u_i\}$
- **else**
  - consider the computed feasible M'-augmenting $u_i v_j$-walk
  - **if** *both* $\eta(u_i)$ *and* $\eta(v_j)$ *are* M*-covered* **then**
    - **return** infeasible
  - **else if** $\eta(u_i)$ *is* M*-covered and* $\eta(v_j)$ *is not* **then**
    - $S \leftarrow S \cup \eta(v_j)$, $G \leftarrow G \setminus \eta(v_j)$, $G' \leftarrow G' \setminus C_{\eta(v_j)}$, $L \leftarrow L \setminus C_{\eta(v_j)}$
  - **else if** $\eta(v_j)$ *is* M*-covered and* $\eta(u_i)$ *is not* **then**
    - $S \leftarrow S \cup \eta(u_i)$, $G \leftarrow G \setminus \eta(u_i)$, $G' \leftarrow G' \setminus C_{\eta(u_i)}$, $L \leftarrow L \setminus C_{\eta(u_i)}$
  - **else**
    - **if** $\eta(u_i) = \eta(v_j)$ **then**
      - $S \leftarrow S \cup \eta(u_i)$, $G \leftarrow G \setminus \eta(u_i)$, $G' \leftarrow G' \setminus C_{\eta(u_i)}$, $L \leftarrow L \setminus C_{\eta(u_i)}$
    - **else**
      - find a basic M'-augmenting structure W contained in the $u_i v_j$-walk using lemma 2(b)
      - **if** W *is an* M'*-augmenting cycle or bi-cycle* **then**
        - **return** infeasible
      - **if** W *is an* M'*-augmenting flower rooted at* u_i **then**
        - $S \leftarrow S \cup \eta(u_i)$, $G \leftarrow G \setminus \eta(u_i)$, $G' \leftarrow G' \setminus C_{\eta(u_i)}$, $L \leftarrow L \setminus C_{\eta(u_i)}$
      - **if** W *is an* M'*-augmenting flower rooted at* v_j **then**
        - $S \leftarrow S \cup \eta(v_j)$, $G \leftarrow G \setminus \eta(v_j)$, $G' \leftarrow G' \setminus C_{\eta(v_j)}$, $L \leftarrow L \setminus C_{\eta(v_j)}$
      - **if** W *is a proper* M'*-augmenting* $u_i v_j$*-path* **then**
        - compute $\text{tb}(\eta(W), \eta(u_i))$ and $\text{tb}(\eta(W), \eta(v_j))$
        - **if** $tb(\eta(W), \eta(u_i))$ *is* M*-augmenting* **then**
          - $S \leftarrow S \cup \eta(u_i)$, $G \leftarrow G \setminus \eta(u_i)$, $G' \leftarrow G' \setminus C_{\eta(u_i)}$, $L \leftarrow L \setminus C_{\eta(u_i)}$
        - **if** $tb(\eta(W), \eta(v_j))$ *is* M*-augmenting* **then**
          - $S \leftarrow S \cup \eta(v_j)$, $G \leftarrow G \setminus \eta(v_j)$, $G' \leftarrow G' \setminus C_{\eta(v_j)}$, $L \leftarrow L \setminus C_{\eta(v_j)}$

if $w(M) < \nu_f^c(G)$ **then**

- **return** infeasible

else

- **return** S

Proof of theorem 4. Let $[(G, w, c), M]$ be the input for the M-vertex-stabilizer problem, with auxiliary $[(G', w', \mathbf{1}), M']$. Algorithm 1 iteratively considers an M'-exposed vertex u_i, and computes a feasible M'-augmenting walk U starting at u_i, if one exists. Lemma 1 implies that $\eta(U)$ is a feasible M-augmenting walk in G. Theorem 3 implies that we need to remove at least one vertex of the walk $\eta(U)$ to stabilize the instance. Note that every vertex $a \neq u_i, v_j$ of U is M'-covered, and hence, $\eta(a)$ is M-covered. Therefore, the only vertices we can potentially remove are $\eta(u_i)$ or $\eta(v_j)$. Hence, if both $\eta(u_i)$ and $\eta(v_j)$ are M-covered, the instance cannot be stabilized and algorithm 1 checks this in line 9. If only one among $\eta(u_i)$ and $\eta(v_j)$ is M-covered, then necessarily we have to remove the M-exposed vertex among the two. Algorithm 1 checks this in line 11 and 13. Note that, by remark 2, instead of computing a new auxiliary for the modified G, we can just remove $C_{\eta(u_i)}$ (resp. $C_{\eta(v_j)}$) from G'. Similarly, if $\eta(u_i) = \eta(v_j)$ and $\eta(u_i)$ is M-exposed, we necessarily have to remove $\eta(u_i)$. Algorithm 1 checks this in line 16. If instead $\eta(u_i) \neq \eta(v_j)$, and both are M'-exposed, we apply lemma 2(b) to find a basic augmenting structure W contained in U. Once again, we know by lemma 1 and theorem 3 that we need to remove a vertex in $\eta(W)$. In case W is a cycle or bi-cycle, all vertices of $\eta(W)$ are M-covered so the instance cannot be stabilized and algorithm 1 checks this in line 20. In case W is a M'-augmenting flower with base u_i or v_j, algorithm 1 accordingly removes $\eta(u_i)$ or $\eta(v_j)$ as all other vertices in $\eta(W)$ are M-covered, in line 23 and 25. Finally, if W is a proper (because $\eta(u_i) \neq \eta(v_j)$) M'-augmenting path, by lemma 4 we know that we can find a feasible M-augmenting walk, where the only M-exposed vertex is either $\eta(u_i)$ or $\eta(v_j)$. Once again, this implies that this vertex must be removed. Algorithm 1 does so in lines 29 and 31.

From the discussion so far, it follows that when we exit the while loop each vertex in S is a necessary vertex to be removed from G, in order to stabilize the instance. We now argue that either removing all vertices in S is also sufficient, or the instance cannot be stabilized. Suppose that the M-vertex-stabilizer instance given by $G \setminus S$ and M is not stable. Theorem 2 implies that $(G \setminus S)'$ contains a basic augmenting structure Q. Note that Q cannot be an M'-augmenting flower with exposed root, or a proper M'-augmenting path with at least one exposed endpoint. To see this, observe that a flower and path are feasible M'-augmenting walks of length at most $3\,|V(G')|$ and $|V(G')|$, respectively. Hence, they would have been found by algorithm 1 in line 4, contradicting that Q exists in $(G \setminus S)'$. It follows that Q is a basic augmenting structure where all vertices are M'-covered. By lemma 1 $\eta(Q)$ is a feasible M-augmenting walk where all vertices are M-covered. This implies that the instance cannot be stabilized. Furthermore, using the ε-augmentation of $\eta(Q)$ we can obtain a fractional c-matching whose value is strictly greater than $w(M)$. Hence, $w(M) < \nu_f^c(G \setminus S)$. Algorithm 1 correctly determines this in line 32. This proves correctness of our algorithm.

Finally, we argue about the running time of the algorithm. Note that each operation that the algorithm performs can be done in polynomial time. Furthermore, after each iteration of the while loop, we either determine that the instance cannot be stabilized, or remove a vertex from G. Therefore, the while loop can be executed at most n times. The result follows. ☐

We close this section with a remark. The authors in [14] have also considered the following problem: given a weighted graph G and a (non necessarily maximum-weight) matching M, find a minimum-cardinality $S \subseteq V$ such that $G \setminus S$ is stable, and M is a maximum-weight matching in $G \setminus S$, i.e., such that the M-vertex-stabilizer instance given by $G \setminus S$ and M is stable. This is a generalization of our definition of the M-vertex-stabilizer problem, which essentially allows M to be not maximum-weight[3]. The authors show that this problem is NP-hard, but admits a 2-approximation algorithm (we mentioned this in the strategy overview), which is best possible assuming Unique Game Conjecture.

With a minor modification of algorithm 1, we can generalize this result to the capacitated setting. Specifically, we start the algorithm by checking if M is maximum-weight in G, and store this in the indicator variable $M_{\max}$. Then, we replace lines 26-31 by algorithm 2.

Theorem 5. *Given a weighted, capacitated graph $G = (V, E)$ and a c-matching M, the problem of computing a minimum-cardinality $S \subseteq V$ such that $G \setminus S$ is stable, and M is a maximum-weight c-matching in $G \setminus S$, admits an efficient 2-approximation algorithm.*

Proof. If M is a maximum-weight c-matching in G, the algorithm is unchanged, and the result follows from theorem 4. So suppose M is not maximum-weight. We follow the argument as in the proof of

[3] In fact, this is the way the M-vertex-stabilizer problem is defined in [14]. We instead use the original definition in [1,6] which assumes M to be maximum.

Algorithm 2: modification algorithm 1, lines 26-31

1 **if** *W is a proper M'-augmenting u_iv_j-path* **then**
2   **if** $M_{\max}$ **then**
3   compute $\mathrm{tb}(\eta(W), \eta(u_i))$ and $\mathrm{tb}(\eta(W), \eta(v_j))$
4   **if** *$tb(\eta(W), \eta(u_i))$ is M-augmenting* **then**
5             $S \leftarrow S \cup \eta(u_i)$, $G \leftarrow G \setminus \eta(u_i)$, $G' \leftarrow G' \setminus C_{\eta(u_i)}$, $L \leftarrow L \setminus C_{\eta(u_i)}$
6   **if** *$tb(\eta(W), \eta(v_j))$ is M-augmenting* **then**
7             $S \leftarrow S \cup \eta(v_j)$, $G \leftarrow G \setminus \eta(v_j)$, $G' \leftarrow G' \setminus C_{\eta(v_j)}$, $L \leftarrow L \setminus C_{\eta(v_j)}$
8   **else**
9         $S \leftarrow S \cup \{\eta(u_i), \eta(v_j)\}$, $G \leftarrow G \setminus \{\eta(u_i), \eta(v_j)\}$, $G' \leftarrow G' \setminus \big(C_{\eta(u_i)} \cup C_{\eta(v_j)}\big)$, $L \leftarrow L \setminus \big(C_{\eta(u_i)} \cup C_{\eta(v_j)}\big)$

theorem 4 (theorem 3 still holds if M is not maximum-weight in G), until we reach the case where we have a proper M'-augmenting path W. We know by lemma 1 and theorem 3 that we need to remove a vertex in $\eta(W)$. We have $\eta(u_i) \neq \eta(v_j)$ and both are M-exposed. Even though it might only be necessary to remove one of them, algorithm 2 removes both vertices on line 9.

Using the same argumentation as in the proof of theorem 4 (theorem 2 still holds if M is not maximum-weight in G), we conclude that either removing all vertices in S is sufficient, or the instance cannot be stabilized, and the algorithm correctly determines this. But, in contrast to that proof, when we exit the while loop, each vertex in S is either a necessary vertex to be removed from G, in order to stabilize the instance, or it was one of two vertices for which it was necessary to remove at least one. Therefore, for any M-vertex-stabilizer S^* we have $|S^*| \geq \frac{1}{2}|S|$. It follows that algorithm 1 with the described modification is a 2-approximation.

The modifications are all operations that can be done in polynomial time. The result follows. □

4 Vertex-Stabilizer

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

Theorem 6. *The vertex-stabilizer problem on capacitated graphs is NP-complete, even if all edges have unit-weight. Furthermore, no efficient $n^{1-\varepsilon}$-approximation exists for any $\varepsilon > 0$, unless $P = NP$.*

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

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

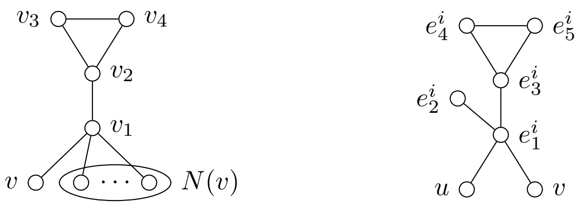
There is no efficient $n^{1-\varepsilon}$-approximation for any $\varepsilon > 0$ for the MIDS problem, unless P = NP [11, corollary 3].

Proof of theorem 6. The decision variant of the problem asks to find a vertex-stabilizer of size at most k. This problem is in NP, since if a vertex set S is given, it can be verified in polynomial time if $|S| \leq k$ and if $\nu^c(G \setminus S) = \nu^c_f(G \setminus S)$. We prove the NP-hardness and approximation factor by given an approximation-preserving reduction from the MIDS problem.

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

$$V(\Gamma_v) = N^+(v) \cup \{v_1, v_2, v_3, v_4\}\,, \tag{10}$$

$$E(\Gamma_v) = \big\{uv_1 : u \in N^+(v)\big\} \cup \{v_1v_2, v_2v_3, v_3v_4, v_2v_4\}\,. \tag{11}$$



(a) Gadget Γ_v. (b) Gadget Γ_{uv}^i.

Figure 2: Examples of gadgets.

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

$$V(\Gamma_{uv}^i) = \left\{u, v, e_1^i, e_2^i, e_3^i, e_4^i, e_5^i\right\}, \tag{12}$$

$$E(\Gamma_{uv}^i) = \left\{ue_1^i, ve_1^i, e_1^ie_2^i, e_1^ie_3^i, e_3^ie_4^i, e_4^ie_5^i, e_3^ie_5^i\right\}. \tag{13}$$

See figure 2 for an example of these gadgets. Now let G' be defined as the union of all Γ_v and all Γ_{uv}^i, such that vertices from V overlap. We set the capacity as follows: $c_v = d_v^{E(G')}$ for all $v \in V$, $c_{v_1} = d_v^E + 1$ for all $v \in V$, $c_{e_1^i} = c_{e_3^i} = 2$ for $e_1^i, e_3^i \in V(\Gamma_{uv}^i)$ for all $e = uv \in E$ and $i \in \{1, \dots, n\}$, and $c_v = 1$ for all remaining $v \in V(G')$. All edges are set to have unit-weight. The key point is:

Claim 2. *G has an independent dominating set of size at most k if and only if $(G', \mathbf{1}, c)$ has a vertex-stabilizer of size at most k.*

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

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

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

Finally, for all $e = uv \in E$ and $i \in \{1, \dots, n\}$, consider Γ_{uv}^i. Since S is dominating, at most one of u and v is in S. If neither are in S, add both ue_1^i and ve_1^i to M. If one of them is in S, without loss of generality let it be u, then add ve_1^i and $e_1^ie_2^i$ to M. Furthermore, add $e_3^ie_4^i$ and $e_3^ie_5^i$ to M. Set $y_{e_1^i} = 1$, $y_{e_2^i} = 0$, $y_{e_3^i} = y_{e_4^i} = y_{e_5^i} = 0.5$, and $z_f = 0$ for all edges f in the gadget.

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

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

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

$$x_e = \begin{cases} x_e^* & \text{if } e \in E(G' \setminus S) \setminus E[\Gamma_v], \\ 1 & \text{if } e \in \{uv_1 : u \in N^+(v)\}, \\ 0 & \text{if } e = v_1v_2, \\ 0.5 & \text{if } e \in \{v_2v_3, v_3v_4, v_2v_4\}. \end{cases} \tag{14}$$

Note that x is a fractional c-matching in $G' \setminus S$, since x^* is. However, $\sum_{e \in E[\Gamma_v]} x_e = d_v + 2.5 > \sum_{e \in E[\Gamma_u]} x_e^*$, since x^* is integral. Hence, $\mathbf{1}^\top x > \mathbf{1}^\top x^*$, contradicting the optimality of x^*.

(ii) Suppose there is some $e = uv \in E$ such that S contains both u and v. All gadgets Γ_{uv}^i are then components in $G' \setminus S$. If u and v are the only vertices in S from some component Γ_{uv}^i, then a maximum-cardinality fractional c-matching in this components is given by $x_{e_1^ie_2^i} = x_{e_1^ie_3^i} = 1$ and $x_{e_3^ie_4^i} = x_{e_4^ie_5^i} = x_{e_3^ie_5^i} = 0.5$. Which means this component is not stable, and thus $G' \setminus S$ is not stable, a contradiction. Hence, S must contain at least one vertex of each Γ^i_{uv} that is neither u nor v. Consequently, $k = |S| \geq n+2$. Since G has only n vertices, it obviously has an independent dominating set of size at most n, and hence of size at most k. Such a set can for example be obtained by a greedy approach. Hence, for the remainder of the proof we can assume that at most one of u and v is in S for each $uv \in E$.

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

$$S'_v = \begin{cases} (S_v \cup S') \cap N^+(v) & \text{if this is nonempty,} \\ v & \text{otherwise.} \end{cases} \tag{15}$$

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

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

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

Before we added S'_v to S', we had $|S'_v \setminus S'| \leq |S_v|$. Consequently, $|S'| \leq \cup_{v \in V} |S_v| \leq |S| = k$. ☐

By this claim, any minimum-cardinality vertex-stabilizer of $(G', \mathbf{1}, c)$ is of the same size as any minimum independent dominating set of G. Further, any efficient α-approximation algorithm for the vertex-stabilizer problem translates into an efficient α-approximation algorithm for the MIDS problem. Hence, the result follows from the inapproximability of the MIDS problem. ☐

5 Cooperative Matching Games

Cooperative matching games in unit-capacity graphs, defined in the introduction, extend quite easily to capacitated graphs, by replacing each ν with ν^c. In unit-capacity graphs G the following statements are equivalent [7, 13]:

(i) G is stable,

(ii) there exists an allocation in the core of the CMG on G,

(iii) there exists a stable outcome for the NBG on G.

We here note that the equivalence does not extend to capacitated graphs.

In particular, as mentioned in the introduction, we still have $(i) \iff (iii)$ proven in [2, corollary 3.3]. The implication $(i) \implies (ii)$ still holds, and follows from [2, lemma 3.4][4]. However, the graph G given in figure 3 shows that $(ii) \not\Rightarrow (i)$ (and hence $(ii) \not\Rightarrow (iii)$).

Assuming all the edges of G in figure 3 have unit weight, it is quite easy to see that $\nu^c(G) = 3$ and $\nu^c_f(G) = 3.5$, thus G is not stable. One can check that $y = (1, 1, 1, 0)$ is in the core.

References

[1] Sara Ahmadian, Hamideh Hosseinzadeh, and Laura Sanità. Stabilizing network bargaining games by blocking players. *Mathematical Programming*, 172:249–275, 2018.

[2] MohammadHossein Bateni, MohammadTaghi Hajiaghayi, Nicole Immorlica, and Hamid Mahini. The cooperative game theory foundations of network bargaining games, 2010.

[4] [2] assumes that the graph is bipartite, but bipartiteness is not needed in their proof.

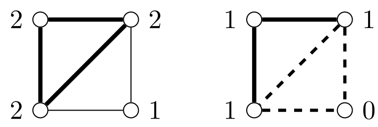


Figure 3: On the left: the graph G where the values close to the vertices indicate the capacities. Bold edges indicate a maximum c-matching. On the right: the graph G where the values close to the vertices indicate the allocation y. A maximum fractional c-matching is given by $x_e = \frac{1}{2}$ for dashed edges, $x_e = 1$ otherwise.

[3] Péter Biró, Walter Kern, and Daniël Paulusma. On solution concepts for matching games. In Jan Kratochvíl, Angsheng Li, Jiří Fiala, and Petr Kolman, editors, *Theory and Applications of Models of Computation*, pages 117–127, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[4] Adrian Bock, Karthekeyan Chandrasekaran, Jochen Könemann, Britta Peis, and Laura Sanità. Finding small stabilizers for unstable graphs. *Mathematical Programming*, 154:173–196, 2015.

[5] Karthekeyan Chandrasekaran. *Graph Stabilization: A Survey*, pages 21–41. Springer Singapore, Singapore, 2017.

[6] Karthekeyan Chandrasekaran, Corinna Gottschalk, Jochen Könemann, Britta Peis, Daniel Schmand, and Andreas Wierz. Additive stabilizers for unstable graphs. *Discrete Optimization*, 31:56–78, 2019.

[7] Xiaotie Deng, Toshihide Ibaraki, and Hiroshi Nagamochi. Algorithmic aspects of the core of combinatorial optimization games. *Mathematics of Operations Research*, 24(3):751–766, 1999.

[8] Linda Farczadi. *Matchings and games on networks*. PhD thesis, University of Waterloo, 2015.

[9] Linda Farczadi, Konstantinos Georgiou, and Jochen Könemann. Network bargaining with general capacities. arXiv preprint arXiv:1306.4302, 2013.

[10] Corinna Gottschalk. *Personal communication*, 2018.

[11] Magnús M. Halldórsson. Approximating the minimum maximal independence number. *Information Processing Letters*, 46(4):169–172, 1993.

[12] Takehiro Ito, Naonori Kakimura, Naoyuki Kamiyama, Yusuke Kobayashi, and Yoshio Okamoto. Efficient stabilization of cooperative matching games. *Theoretical Computer Science*, 677:69–82, 2017.

[13] Jon Kleinberg and Éva Tardos. Balanced outcomes in social exchange networks. In *Proceedings of the 40th STOC*, pages 295–304, 2008.

[14] Zhuan Khye Koh and Laura Sanità. Stabilizing weighted graphs. *Mathematics of Operations Research*, 45(4):1318–1341, 2020.

[15] Jochen Könemann, Kate Larson, and David Steiner. Network bargaining: Using approximate blocking sets to stabilize unstable instances. In Maria Serna, editor, *Algorithmic Game Theory*, pages 216–226, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[16] John F. Nash. The bargaining problem. *Econometrica*, 18:155–162, 1950.

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