Math. Program., Ser. B
DOI 10.1007/s10107-017-1177-9

FULL LENGTH PAPER

Stabilizing network bargaining games by blocking players

Sara Ahmadian[1] · **Hamideh Hosseinzadeh**[2] · **Laura Sanità**[1]

Received: 30 April 2016 / Accepted: 8 July 2017

Abstract Cooperative matching games (Shapley and Shubik) and Network bargaining games (Kleinberg and Tardos) are games described by an undirected graph, where the vertices represent players. An important role in such games is played by *stable* graphs, that are graphs whose set of inessential vertices (those that are exposed by at least one maximum matching) are pairwise non adjacent. In fact, stable graphs characterize instances of such games that admit the existence of stable outcomes. In this paper, we focus on stabilizing instances of the above games by *blocking* as few players as possible. Formally, given a graph G we want to find a minimum cardinality set of vertices such that its removal from G yields a stable graph. We give a combinatorial polynomial-time algorithm for this problem, and develop approximation algorithms for some NP-hard *weighted* variants, where each vertex has an associated non-negative weight. Our approximation algorithms are LP-based, and we show that our analysis are almost tight by giving suitable lower bounds on the integrality gap of the used LP relaxations.

Mathematics Subject Classification 05C07 · 05C70 · 05C75 · 05C85 · 68Q15 · 68Q17 · 68R05 · 68R10 · 68W25 · 90C05 · 90C10 · 90C27 · 90C35 · 90C59 · 90C60 · 91A12 · 91A40 · 91A46 · 91A80 · 91B26

✉ Laura Sanità
lsanita@uwaterloo.ca

Sara Ahmadian
sahmadian@uwaterloo.ca

Hamideh Hosseinzadeh
hamideh.hosseinzadeh@gmail.com

1 Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Canada

2 Faculty of Mathematical Science, Alzahra University, Tehran, Iran

Published online: 25 July 2017

1 Introduction

Game theory is an active and important area of research in the field of Theoretical Computer Science, and combinatorial optimization techniques are often crucially employed in solving game theory problems [17]. For several games defined on networks, studying the structure of the underlying graph that describes the network setting is important to identify the existence of good outcomes for the corresponding games. Prominent examples are *cooperative matching games* introduced by Shapley and Shubik [19] and *network bargaining games* studied by Kleinberg and Tardos [11]. These are games described by an undirected graph $G = (V, E)$, where the vertices represent players, and the cardinality of a maximum matching in G, denoted by $\nu(G)$, represents a total *value* that the players could gain by interacting with each other.

In an instance of a cooperative matching game [19], one seeks for an *allocation* of the value $\nu(G)$ among players, described by a vector $y \in \mathbb{R}^V_{\geq 0}$, in which no subset of players S has an incentive to form a coalition to deviate. This is formally described by the constraint $\sum_{v \in S} y_v \geq \nu(G[S])$ for all subsets S, where $G[S]$ denotes the subgraph induced by the vertices in S. Such allocation is called *stable*. It is well-known (see e.g. [6]) that cooperative matching game instances that admit the existence of a stable allocation are precisely the set of instances described by *stable graphs*: these are graphs whose set of *inessential vertices* are pairwise non adjacent. We recall here that a vertex v of a graph G is called inessential if there exists at least one maximum matching M in G that *exposes* v, that is, v is not an endpoint of M, and it is called essential otherwise.

Network bargaining games described by Kleinberg and Tardos [11] are network extensions of the classical Nash bargaining games [16]. In an instance of a network bargaining game described by a graph G, the edges represent a set of potential *deals* of unit value that the players (vertices) could make. An outcome of the game is given by a matching M of G (representing the set of deals that the players made) together with a value allocation $y \in \mathbb{R}^V_{\geq 0}$ on each vertex (representing how the players decided to split the values of the deals they made, if any). Kleinberg and Tardos [11] introduced the notion of *stable* outcomes for such games, that are outcomes where no player has an incentive to deviate, as well as the notion of *balanced* outcomes, that are stable outcomes in which, in addition, the values are "fairly" split among the players. The authors proved that a balanced outcome exists if and only if a stable outcome exists, and this happens if and only if the graph G describing the instance is *stable*.

Since not all graphs are stable, there are instances of both network bargaining games and cooperative matching games that do not admit stable solutions. This motivated many authors in past years to address the algorithmic problem of *stabilizing* such instances by minimally modifying the underlying graph. Two very natural ways to modify a graph in order to achieve some desired properties are via *edge-removal* or *vertex-removal* operations. The authors in [5] looked at edge-removal operations, that is, stabilizing instances of the above games by blocking potential deals that the players could make. In this paper, we look at the vertex-removal counterpart, that is, stabilizing instances by *blocking players*. Formally, this translates into the following problem:

Vertex-stabilizer problem: *Given a graph $G = (V, E)$, find a minimum cardinality vertex-stabilizer, that is a set $S \subseteq V$ whose removal from G yields a stable graph.*

We also generalize and study this problem in the *weighted* setting (a formal definition is in the next subsection).

In addition to the connection with game theory, the vertex-stabilizer problem is also of interest from a combinatorial optimization perspective. In fact, an alternative and equivalent characterization of stable graphs can be given using linear programming and the notion of *fractional* matchings and vertex covers, as we are now going to explain. For a graph $G = (V, E)$, a fractional matching is a feasible solution to the LP:

$$\nu_f(G) := \max \left\{ \sum_{e \in E} x_e : \sum_{e \in \delta(v)} x_e \leq 1 \; \forall v \in V, \; x \geq 0 \right\},$$

where $\delta(v)$ denotes the set of edges incident into v. Note that, if we add binary constraints to the above LP we obtain a formulation to find a matching of G of maximum cardinality $\nu(G)$. The dual of the above LP is:

$$\tau_f(G) := \min \left\{ \sum_{v \in V} y_v : y_u + y_v \geq 1 \; \forall \{u, v\} \in E, \; y \geq 0 \right\}.$$

Once again, note that if we add binary constraints to this dual LP we obtain the canonical formulation for finding a *vertex cover* of G of minimum cardinality $\tau(G)$, that is, a min-cardinality subset of vertices covering all edges of the graph. For this reason, fractional feasible solutions to the above dual LP are called *fractional* vertex covers.

By duality theory, we know that the following holds: $\nu(G) \leq \nu_f(G) = \tau_f(G) \leq \tau(G)$. In general, there are graphs for which all the above inequalities are strict (e.g. a triangle). However, for certain classes of graphs some of the above inequalities hold tight. In particular, the class of *König–Egerváry* graphs [14,20] is formed by all graphs G for which $\nu(G) = \tau(G)$, that is, all the above inequalities hold tight. Note that the class of König–Egerváry graphs is a proper superset of the class of bipartite graphs. It is known (see e.g. [11]) that stable graphs are exactly the class of graphs for which $\nu(G) = \nu_f(G) = \tau_f(G) \leq \tau(G)$, that is, graphs for which the cardinality of a maximum matching ($\nu(G)$) is equal to the minimum size of a *fractional* vertex cover ($\tau_f(G)$). We have therefore the following relation:

(Bipartite graphs) $\subsetneq$ (König-Egerváry graphs) $\subsetneq$ (Stable graphs) $\subsetneq$ (General graphs).

The algorithmic problems of turning a general graph into a bipartite one by removing either a set of edges or a set of vertices of minimum weight/cardinality, have been studied in the literature (see e.g. [1,9]). Similarly, the algorithmic problems of turning a given graph into a König–Egerváry one by removing a min-cardinality subset of edges or of vertices have been studied (see e.g [15]). Differently, as mentioned before, for stable graphs only the edge-removal question has been investigated so far, and this yields an additional motivation to study the vertex-removal question in this paper, both in the unweighted and in the weighted setting.

Our results and techniques We study the vertex-stabilizer problem in Sect. 2. We first show a structural property of any minimal vertex-stabilizer. Namely, we prove that removing any minimal vertex-stabilizer *does not decrease* the size of a maximum matching in the resulting graph (Theorem 1). This theorem has an interesting interpretation in network bargaining and cooperative matching games: it states that it is always possible to stabilize instances by blocking a minimum number of players *without* decreasing the total value that the players could get. An analogue of Theorem 1 has been proven by Bock et al. [5] for minimal *edge*-stabilizers,[1] however, their proof does not hold for the vertex-removal setting, and therefore our proof is different. Interestingly, despite this analogy, algorithmically the two problems appear to have a different complexity: while finding a min-cardinality edge-stabilizer is at least as hard as finding a minimum vertex cover [5], we here prove (Theorem 2) that finding a min-cardinality vertex-stabilizer is a polynomial-time solvable problem. In addition, we can prove (Theorem 3) that the problem of blocking as few players as possible in order to make a *given* set of deals realizable as a stable outcome is also polynomial-time solvable, once again in contrast with the edge-removal setting, where the analogous question has been studied by [5] and shown to be vertex cover-hard. These three theorems are proved using combinatorial techniques. Theorem 1 exploits the structure of maximum matchings in graphs, that follows from the seminal works in [3,8]. Using Theorem 1, one can compute a lower bound on the size of a minimum vertex-stabilizer (as is done in [5]) exploiting properties of the so-called *Edmonds-Gallai Decomposition* (EGD) of a graph (definition is in Sect. 2). By further exploiting the relation that interplays between matchings and EGD, we get algorithms that prove Theorems 2 and 3.
We study in Sect. 3 the *weighted* setting. In the vertex-stabilizer problem described before, players are all equally considered, that is, from an objective function perspective, we are assuming that blocking a player u is as costly as blocking a player v, independently on how u and v are connected to the rest of the players in the network. However, from a bargaining perspective, players might not all be equally powerful: as an example, players corresponding to essential vertices have more bargaining power than inessential ones. Moreover, blocking a player that is highly connected in the graph and therefore has the potential to enter in many deals might be more costly than blocking a less connected player. For this reason, blocking different players might have different costs. We can model this by assigning a *weight* $w_v \geq 0$ to each vertex v. In this setting, we could be interested in either *minimizing* the weight of the *blocked* players, or in *maximizing* the weight of the *remaining* players. Two optimization problems then arise:

Min-weight vertex-stabilizer: *Given a graph* $G = (V, E)$, *and vertex weights* $w_v \geq 0\ \forall v \in V$, *find a* vertex-stabilizer S *that minimizes* $w(S) = \sum_{v \in S} w_v$.

Max-weight vertex-stabilizer: *Given a graph* $G = (V, E)$, *and vertex weights* $w_v \geq 0\ \forall v \in V$, *find a* vertex-stabilizer S *that maximizes* $w(V \setminus S) = \sum_{v \notin S} w_v$.

This weighted setting poses more algorithmic challenges, and this is technically the most interesting part of the paper. We prove that both the above problems become

[1] These are subsets of *edges* such that their removal from G yields a stable graph.

NP-hard already if 2 different weights are involved (Theorem 7). For this reason, we focus on *approximation algorithms*. We give a 2-approximation algorithm for the max-weight vertex-stabilizer problem (Theorem 4), and a $O(\gamma)$-approximation algorithm for the min-weight vertex-stabilizer problem (Theorem 5), where γ is the size of the so-called *Tutte-set* of the graph G (a formal definition is in Sect. 2). Both our algorithms are LP-based and rely on the following strategy. As a first step, we identify a suitable LP-relaxation to use for our problems. To this extent, we show that we can reduce our problems to vertex-deletion problems in a *bipartite* graph, in which the goal is to remove a subset of vertices in order to turn some special nodes into *essential* vertices in the remaining graph. This reinterpretation of the problem allows us to write a formulation that uses a set of *flow-type* valid constraints, and exploiting the properties of this flow will be crucial to round fractional solutions into integral ones.

In addition, we show lower bounds on the integrality gap of the LP relaxations we use, that show that our analysis are almost tight. We give a $\frac{3}{2}$ lower bound on the integrality gap in the max-weight case, and a $\Omega(\gamma)$ lower bound in the min-weight case, that asymptotically matches the developed approximation factor. The lower bound for the min-weight case holds even on graph with *bounded* (constant) degree, and to construct it we rely on suitable *unbalanced bipartite expander* graphs.

We conclude by showing that we can give an algorithm for the min-weight vertex-stabilizer problem whose approximation factor *is* bounded by the maximum degree of a vertex in G, *if* we have an additional information: namely, if we know which is the set of essential vertices in the final graph (Theorem 6). From a network bargaining perspective, this corresponds to stabilize instances *enforcing* that some specific players will always be able to enter in a deal in any stable outcome. Also for this latter case we show a matching lower bound on the integrality gap of the LP relaxation we use. Our lower bounds show that to improve significantly our approximation factors a different strategy or at least different formulations have to be used.

Finally, we want to mention that recently Ito et al. [10] have given independently alternative proofs of Theorems 2 and 7.

Related works Removing vertices or edges from a graph as to satisfy certain properties has been widely studied in the literature in many variants. The paper that is most closely related to our work is [5] that studied the edge-stabilizer problem in the unweighted setting, and in addition to the results previously mentioned, they give efficient approximation algorithms for sparse graphs and for regular graphs. Biró et al. [4] also studied the edge-stabilizer problem, but considering maximum-weight matchings instead of maximum-cardinality matchings, and showed NP-hardness for this case. Könemann et al. [13] studied a related problem of computing a minimum-cardinality *blocking set*, that is a set of edges F such that $G\backslash F$ has a fractional vertex cover of size at most $\nu(G)$ (but note that $G\backslash F$ might not be stable). They give approximation algorithms for sparse graphs. Mishra et al. [15] studied vertex-removal and edge-removal problems to turn a graph into a König–Egerváry graph. Among other results, they give an $O(\log n \log \log n)$ approximation algorithm for the vertex-removal case in the unweighted setting, and show that assuming Unique Game Conjecture, both the minimum vertex-removal and edge-removal problems do not admit a constant factor approximation algorithm. We note that their hardness results do not seem to be helpful for our setting, since the graphs used in their reductions are stable. As already mentioned, Ito et al. [10] have given independently alternative proofs of Theorems 2 and 7. They also give polynomial-time algorithms to stabilize an unweighted graph by adding edges and by adding vertices.

2 Minimum cardinality vertex-stabilizers

We first prove that the removal of any inclusion-wise *minimal* vertex-stabilizer does not decrease the cardinality of a maximum matching in the resulting graph. Note that a minimum cardinality vertex-stabilizer is of course minimal. Here $G\backslash S$ denotes the graph obtained by removing from $G = (V, E)$ the subset of vertices $S \subseteq V$.

Theorem 1 *For any minimal vertex-stabilizer $S \subseteq V$ of a graph $G = (V, E)$, we have $\nu(G\backslash S) = \nu(G)$.*

Before giving a proof, we need a proposition (see [11]) that follows from standard results in matching theory, and uses the notion of *M-flower* for a maximum matching M of G. An M-flower is a subgraph of G formed by a u, v-path of even length that starts at an M-exposed vertex u and alternates edges in $E\backslash M$ and edges in M, plus a cycle containing v of $2k + 1$ edges, for some integer $k \geq 1$, in which exactly k edges are in M.

Proposition 1 [11] *Given graph G, the following are equivalent characterizations of a stable graph:* (i) *The set of inessential vertices of G are pairwise non adjacent,* (ii) $\nu(G) = \tau_f(G)$, (iii) *There is no M-flower in G for any maximum matching M. Moreover, if G is not stable, then for every maximum matching M there is an M-flower.*

Proof of Theorem 1 Let S be a minimal vertex-stabilizer of $G = (V, E)$, and M be a maximum matching of $G\backslash S$. Suppose by contradiction that $|M| < \nu(G)$. By classical results on matching theory [3], since M is not a maximum matching in G there exists an *M-augmenting path* P in G, that is, a path P that alternates edges from $E\backslash M$ and edges from M with endpoints s and t which are exposed by M. Clearly, we must have $|S \cap \{s, t\}| \geq 1$, otherwise P would be an augmenting path in $G\backslash S$, contradicting that M is a maximum matching. We distinguish two cases:

Case 1 $|S \cap \{s, t\}| = 1$. Without loss of generality, assume $s \in S$. In this case, we will show that $S' = S\backslash\{s\}$ is a vertex-stabilizer of G, which is a contradiction to the minimality of S. Consider the matching $M' = M \Delta P$, where Δ denotes the symmetric difference operator. M' is a matching of $G\backslash S'$ and $|M'| = |M| + 1$. Since adding one vertex to an arbitrary graph can increase the size of maximum matching by at most one, we deduce that M' is a maximum matching of $G\backslash S'$, hence $\nu(G\backslash S') = |M'|$. We now prove that $G\backslash S'$ is stable by showing that $\nu(G\backslash S') = \tau_f(G\backslash S')$. Let $y \in \mathbb{R}^{V\backslash S}_{\geq 0}$ be a minimum size fractional vertex cover of $G\backslash S$. By stability of $G\backslash S$, $\nu(G\backslash S) = \mathbf{1}^T y$. Define vector $y' \in \mathbb{R}^{V\backslash S'}_{\geq 0}$ as $y'_v = y_v$ for all $v \in V\backslash S$, and $y'_s = 1$. Obviously y' is a fractional vertex cover of $G\backslash S'$. So we

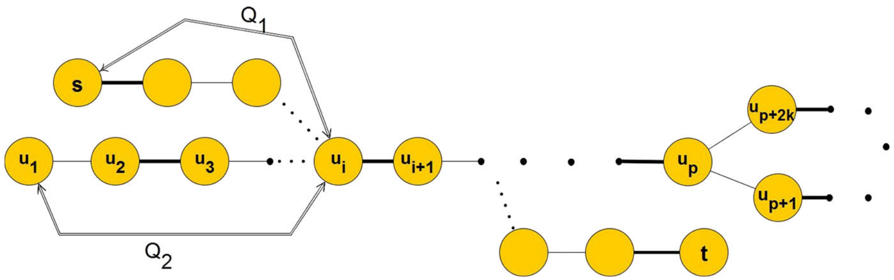

Fig. 1 M' edges are shown by bold edges. Note $M' \cap Q_2 = M \cap Q_2$ while $M \cap Q_1 = Q_1 \backslash M'$

have $\tau_f(G\backslash S') \leq \mathbf{1}^T y' = \mathbf{1}^T y + 1 = \nu(G\backslash S) + 1 = \nu(G\backslash S')$, i.e., $G\backslash S'$ is stable.

Case 2 $|S \cap \{s, t\}| = 2$. We first observe that $(G\backslash S) \cup \{s\}$ does not contain any M-augmenting path. Otherwise, by the same arguments as in Case 1, we can deduce that $S\backslash\{s\}$ is a vertex-stabilizer, and obtain a contradiction. Similarly, $(G\backslash S) \cup \{t\}$ does not contain any M-augmenting path. Let $S' = S\backslash\{s, t\}$, and $M' = M \Delta P$. We first show that M' is a maximum matching in $G\backslash S'$. If not, then $\nu(G\backslash S') \geq \nu(G\backslash S) + 2$. Let M'' be a maximum matching in $G\backslash S'$. If we remove s from $G\backslash S'$, we delete at most one edge of M''. Therefore, $\nu((G\backslash S) \cup \{s\}) \geq \nu(G\backslash S) + 1$. However, this implies that M is not a maximum matching in $(G\backslash S) \cup \{s\}$, and therefore $(G\backslash S) \cup \{s\}$ contains an M-augmenting path contradicting our first observation. Since M' is a maximum matching in $G\backslash S'$, and $G\backslash S'$ is not stable, by Proposition 1 there exists an M'-flower F, with vertex set $u_1, \ldots, u_p$, with u_1 being the M'-exposed vertex on the even-length path. Note that F cannot be vertex disjoint from P: otherwise, F would be an M-flower as well in $G\backslash S$, contradicting stability of $G\backslash S$. It follows that $F \cup P$ is a connected subgraph of G (see Fig. 1). Let u_i be the node with the smallest index i that belongs to both F and P. Note that $i \neq 1$, since u_1 is M'-exposed and every node in P is instead M'-covered. Moreover, i is necessarily an even number: if odd, then the edge $\{u_{i-1}, u_i\}$ is in both P and F, contradicting our choice of i. Furthermore, note that the edge $\{u_i, u_{i+1}\}$ belongs to both P and F. Consider the path Q_1 that is the subpath of P connecting u_i to either s or t in $P\backslash\{u_i, u_{i+1}\}$, and the path Q_2 that is the subpath in F with vertex set $u_1, \ldots, u_i$. Their union $Q_1 \cup Q_2$ forms a path from u_1 to either s or t, say s (the other case is similar). In this case, $Q_1 \cup Q_2$ is an M-augmenting path in $(G\backslash S) \cup \{s\}$, contradicting our first observation. □

We will now state a useful lower bound on the size of a minimum cardinality vertex-stabilizer, which relies on the notion of *Edmonds-Gallai Decomposition* (EGD) of a graph. The EGD of a graph $G = (V, E)$ is a partition of the set of vertices V into 3 sets (B, C, D) where B is the set of inessential vertices of G, C is the set of essential vertices of G that have at least one adjacent vertex in B, and D is the set of the remaining essential vertices of G (see Fig. 2). The set C is called the *Tutte-set* of G. The following lemma states some known properties of EGD of graphs. For a proof, see e.g. [18].

Lemma 1 [18] *Let* (B, C, D) *be the Edmonds-Gallai Decomposition of a graph* G. *Let* $G_1, G_2, \ldots, G_r$ *be the components of the graph* $G[B]$ *induced by* B. *Then, each*

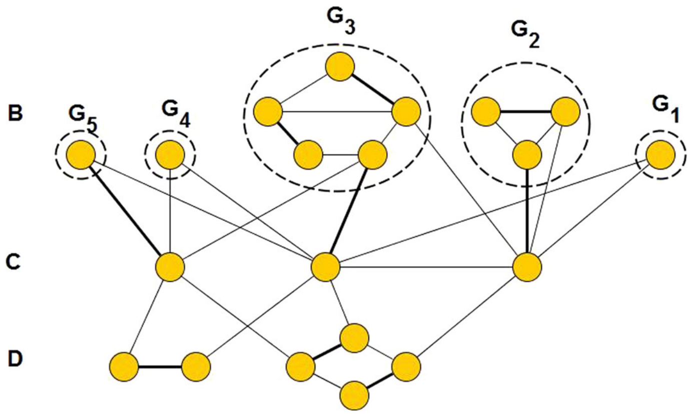

Fig. 2 Example of Edmonds-Gallai decomposition for graph G with maximum matching M showed by bold edges

non-singleton component $G_i = (V_i, E_i)$ is factor-critical, i.e., for each $v \in V_i$, the graph $G_i \setminus v$ admits a perfect matching. Furthermore, if M is a maximum matching in G, then: (a) M induces a perfect matching between vertices of D; (b) M matches C into distinct components of $G[B]$; (c) M induces a near-perfect matching in each G_i, i.e., $|M \cap E_i| = \frac{|V_i|-1}{2}$.

A lower bound on the size of a minimum stabilizer is given in the following lemma, that has been proved by Bock et al. [5] for edge-stabilizers, but their proof easily extends to vertex-stabilizers, assuming Theorem 1. We report here a proof for completeness.

Lemma 2 [5] *Let (B, C, D) be the EGD of a graph G, and let M be a maximum matching of G that covers the maximum possible number of isolated vertices (i.e., singletons) in the graph $G[B]$. Let k be the number of non-singleton components of $G[B]$ with one vertex exposed by M. Then for any vertex-stabilizer S, we have $|S| \geq k$.*

Proof Let $G_1, \ldots, G_k$ denote the non-singleton components in $G[B]$ with at least one vertex exposed by M. Let S be a minimal vertex-stabilizer and $H = G \setminus S$. For each component $G_1, \ldots, G_k$, at least one vertex $v_i \in G_i$ becomes essential in H. If not, either we removed at least two vertices in G_i (but this is not possible since, according to Lemma 1(c), it would decrease the size of maximum matching, contradicting Theorem 1) or it means that some inessential vertices are adjacent, again a contradiction with H being stable.

Pick a maximum matching N in H. Then, N will cover all these vertices $v_1, \ldots, v_k$ that are essential in H. Since G_i is factor-critical and M matches all but one vertex in G_i using edges in G_i, we may assume without loss of generality, that M exposes all these vertices. The graph $M \Delta N$ is a vertex-disjoint union of even cycles and even paths since $|M| = |N| = \nu(G)$. Consider the k disjoint paths starting at the vertices $v_1, \ldots, v_k$ in $M \Delta N$. We observe that at least one vertex in each of these paths should belong to S, otherwise we can obtain a maximum matching in H that exposes the starting vertex v_i, thus contradicting v_i being an essential vertex in H. Hence $|S| \geq k$. □

We now state our algorithm to find a minimum cardinality vertex-stabilizer.

Algorithm 1

1. Compute the EGD (B, C, D) of G, and a maximum matching M^* of G that covers the maximum possible number of isolated vertices in the graph $G[B]$.
2. Let $G_1, \ldots, G_k$ be the non-singleton components of $G[B]$ with one vertex exposed by M^*. Set $S := \bigcup_{i=1}^{k}\{v_i\}$ where v_i is the M^*-exposed vertex of G_i.

Theorem 2 *Algorithm 1 is a polynomial-time algorithm to compute a minimum cardinality vertex-stabilizer S of a given graph G.*

Proof We consider Algorithm 1. First, note that the algorithm is a polynomial-time algorithm, since the EGD can be computed using Edmonds' Blossom algorithm (see [18]), and the matching M^* can be computed by reducing the problem to maximum weighted matching on a bipartite graph, as follows: consider the bipartite graph obtained by first taking $G[B \cup C]$, then removing the edges between C-vertices, and finally shrinking each non-singleton component of $G[B]$ into one pseudonode (ignore multiple copies of edges created by this last operation, if any). Assign zero weight to the edges incident into pseudonodes, and unit weight to all other edges. Then compute a C-perfect matching $\bar{M}$ of maximum weight on this bipartite weighted graph. It is not difficult to show that $\bar{M}$ can be extended to a maximum matching M^* of G with the desired property, by adding a perfect matching between vertices in D, and a near-perfect matching in each non-singleton component of $G[B]$ as to fulfill properties (a),(b),(c) of Lemma 1.

Second, note that $|S| = k$, matching the lower bound given in Lemma 2. Therefore, to finish the proof, we only need to show that $G \setminus S$ is stable. We do this by constructing a fractional vertex-cover y of $G \setminus S$ with $\mathbf{1}^T y = |M^*|$. Since M^* is clearly a maximum matching in $G \setminus S$, this proves stability. Partition the set B into two sets: B_1 and B_2, where B_2 contains the singleton vertices in $G[B]$ and B_1 all the rest. Note that $S \subseteq B_1$. Let $C_1 \subseteq C$ be the set of vertices that are matched to vertices in B_1 by M^*. We assign $y_v = \frac{1}{2}$ for all $v \in D \cup (B_1 \setminus S) \cup C_1$. Then we repeat the following process: for each $v \in B_2$ that is adjacent to some node in $u \in C$ with current assigned value $y_u = \frac{1}{2}$, set $y_v = \frac{1}{2}$ and $y_w = \frac{1}{2}$, where $w \in C$ is the vertex matched to v by M^*. Note that such w must exists, i.e., v cannot be a vertex in B_2 exposed by M^*: if this is the case, then it is easy to realize that there is an M^*-alternating path P of even length between $v \in B_2$ and some vertex $\bar{v} \in B_1$: taking $P \Delta M^*$ we would obtain another maximum matching in G that exposes one less singleton vertex in $G[B]$, namely v, contradicting our choice of M^*. We repeat this process until each vertex $v \in B_2$ that does not have an assigned y-value is adjacent only to vertices in C that also do not have an assigned y-value. At this point, we set $y_v = 0$ for all remaining vertices in B_2 and $y_u = 1$ for all remaining vertices in C. Note that all vertices with y-value 1 in C are matched to (a subset of) vertices of y-value 0 by M^*, and M^* induces a perfect matching on the set of vertices with y-value $\frac{1}{2}$. Therefore, by construction, $\mathbf{1}^T y = |M^*|$. Furthermore, y is a fractional vertex cover of $G \setminus S$, since vertices with y-value 0 are only adjacent to vertices with y-value 1. The result follows. □

2.1 When M is given

We here consider the optimization problem of blocking as few players as possible in order to make a *given* set of deals (represented as a maximum matching M) realizable as a stable outcome. This translates into finding a minimum vertex-stabilizer S with the additional constraint that S must be element-disjoint from the given maximum matching M. We call such S an *M-vertex-stabilizer*.

Theorem 3 *Given a graph G and a maximum matching M of G, there is a polynomial-time algorithm to compute a minimum M-vertex-stabilizer.*

We remark that this is in contrast with the edge-removal setting, where the analogous question has been studied in [5] and shown to be as hard as finding a minimum vertex-cover. Before giving the proof, we need to state two useful lemmas.

Lemma 3 *Let S be a minimal vertex-stabilizer for a given graph $G = (V, E)$, and let M be a maximum matching of $G \backslash S$. Let (B, C, D) be the EGD of G, and $G_i = (V_i, E_i)$ be a non-singleton component of $G[B]$. Then all vertices of $V_i \backslash S$ are essential in $G \backslash S$.*

Proof First, assume $V_i \cap S = \emptyset$. Suppose by contradiction that there is a vertex $v' \in V_i$ that is inessential in $G \backslash S$. Then, since M is a maximum matching of G as well, $|M \cap E_i| = \frac{|V_i|-1}{2}$ by Lemma 1. On the other hand, G_i is factor-critical and it is a subgraph of $G \backslash S$, therefore every vertex in V_i is inessential in $G \backslash S$, contradicting that inessential vertices in a stable graph must form an independent set. Assume now $V_i \cap S \neq \emptyset$. Since M exposes at most one vertex v in V_i, then $|V_i \cap S| = 1$. Since every maximum matching $\bar{M}$ of $G \backslash S$ must satisfy $|\bar{M} \cap E_i| = \frac{|V_i|-1}{2}$, it follows that every vertex of $V_i \backslash S$ is essential in $G \backslash S$. □

Lemma 4 *Let M be a maximum matching of a graph $G = (V, E)$, and $S \subseteq V$ be element-disjoint from M. Then, S is an M-vertex-stabilizer if and only if it contains every M-exposed vertex $v \in V$ such that there exists an M-alternating path in G from v to a vertex w in a non-singleton component of $G[B]$.*

Proof To see necessity, suppose that there is an M-exposed vertex $v \notin S$ with an M-alternating path to some vertex w of G_i. Then w is M-covered, because the path cannot be M-augmenting, and therefore w is an essential vertex in $G \backslash S$ by Lemma 3. However, by swapping the edges along this path, we can obtain a maximum matching of $G \backslash S$ that exposes w, a contradiction.

To see sufficiency, let S be a set satisfying the condition in the lemma. Suppose by contradiction that $G \backslash S$ is not stable. Then, there are two inessential vertices u, w that are adjacent in $G \backslash S$. Since $\nu(G) = \nu(G \backslash S)$ by construction, u and w are also inessential vertices in G, implying that there exists one component $G_i = (V_i, E_i)$ of $G[B]$ such that $u, w \in V_i$. If u is M-exposed, then w is M-covered by Lemma 1, and clearly we have an M-alternating path from u to a M-covered vertex in V_i, contradicting our assumption. So assume u is M-covered. By Lemma 1, there exists in $G \backslash S$ a maximum matching M_u exposing u. The symmetric difference $M \Delta M_u$, contains a path from u to an M-exposed vertex of $G \backslash S$ that alternates M-edges and M_u edges. However, such a path also exists in G, again contradicting our assumption. □

Proof (Proof of Theorem 3). We let S be the set of all M-exposed vertices in G with an M-alternating path to a vertex w in a non-singleton component of $G[B]$. Note that by Lemma 4, this is a necessary and sufficient condition for S to be an M-vertex-stabilizer, so S is a minimum cardinality M-vertex-stabilizer. Note that S can be computed in polytime by running a modified *breadth first search* (BFS) algorithm for each M-exposed vertex v which is only allowed to use M edges at even levels. □

At last, we point out that Theorem 3 easily extends to the weighted case, since the set S obtained as in the proof of Theorem 3 includes only those vertices that are *necessary* to remove in order to stabilize a graph while preserving a given maximum matching M (by Lemma 4). Therefore, the above arguments show that, given non-negative weights on the vertices, there is a polynomial-time algorithm to compute an M-vertex-stabilizer S the minimizes $w(S)$ (resp. maximizes $w(V \setminus S)$). For this reason, in the next section we will focus only on the general case when M is not given.

3 The weighted case

We here deal with the vertex-stabilizer problem in the weighted setting, which is much more challenging than the unweighted one. We will show in Sect. 4 that both the min-weight vertex-stabilizer problem and the max-weight vertex-stabilizer problem are NP-hard, even if there are only 2 distinct weights. Note that, if there is only one weight value, then the problems trivially reduce to the unweighted setting, and therefore are polynomial-time solvable. Since the weighted variants are NP-hard, we focus on approximation algorithms. As a first step in developing our approximation results, we find a suitable reformulation of our problems in *bipartite graphs*.

3.1 Reformulation in bipartite graphs

We start with a lemma that follows easily from Theorem 1.

Lemma 5 *Let (B, C, D) be the EGD of a graph G. Let $G_1, G_2, \ldots, G_p$ be the components of $G[B]$ where $G_i = (V_i, E_i)$. Let S be an optimal solution to a min-weight vertex-stabilizer (resp. max-weight vertex-stabilizer) instance defined on G. Then,* (i) *S is a subset of B,* (ii) $|S \cap V_i| \leq 1$*,* (iii) *if* $|S \cap V_i| = 1$*, then the vertex $g_i \in S$ of G_i is a minimum weight vertex in G_i.*

Proof Since any min-weight (resp. max-weight) vertex-stabilizer is of course a *minimal* stabilizer, we can apply Theorem 1 and say that $\nu(G \setminus S) = \nu(G)$, so $S \subseteq B$ as the other vertices are all essential, therefore (i) holds. Note that if G_i is a singleton, (ii) and (iii) hold trivially. So let us assume G_i is a non-singleton component of $G[B]$, and let M be maximum matching of $G \setminus S$. By Lemma 1(c), we know that any maximum matching matches all but at most one vertex of V_i, and therefore (ii) holds. Now suppose $V_i \cap S = \{v\}$, and v is not a vertex with minimum weight in G_i. Let g_i be a vertex in G_i with minimum weight. We claim $S' = S \setminus \{v\} \cup \{g_i\}$ is a vertex-stabilizer, contradicting optimality of S. Suppose that $G \setminus S'$ is not stable. Then, there are at least two inessential vertices u, w in $G \setminus S'$ that are adjacent. Note that u, w cannot be in

V_i, because Lemma 1(c) implies that vertices in V_i must be covered in any maximum matching. Therefore, u, w are also adjacent in $G\backslash S$. Since $G\backslash S$ is stable, at least one if the two vertices, say w, is essential in $G\backslash S$. Take a matching $\bar{M}$ of $G\backslash S'$ that exposes w. Since G_i is factor-critical, we can construct a maximum matching in $G\backslash S$ by replacing the perfect matching in $V_i\backslash\{v\}$ induced by $\bar{M}$ with a perfect matching in $V_i\backslash\{g_i\}$. However, this new maximum matching exposes w, a contradiction with w being essential in $G\backslash S$. The result follows. □

We can use Lemma 5 to simplify our input. If S contains a vertex from a component G_i, then it must be one of the vertices with minimum weight in G_i. Therefore, we shrink each non-singleton component G_i to a vertex g_i with minimum weight among the vertices in the component, and we call it a *pseudonode* (we remove multiple copies of the same edge created with this operation, if any). Additionally, we know that $S \cap D = \emptyset$, so we can safely ignore these vertices and temporarily remove them from G. Lastly, we know by Lemma 1(b) that every maximum matching of G matches all vertices in C to vertices in different components of $G[B]$; therefore, we ignore and remove edges between vertices in C from G. In this way we construct from G a weighted bipartite graph $G_b = (\tilde{B} \cup C, \tilde{E})$, where $\tilde{E} \subseteq E$, and $\tilde{B}$ consists of two sets of vertices: the set of pseudonodes, call this set B_1, and vertices corresponding to singletons in $G[B]$, call this set B_2. By construction and Lemma 1(b), $\nu(G_b) = |C|$ and S naturally corresponds to a subset of $\tilde{B}$ of the same weight.

Definition 1 Let $H = (U \cup W, F)$ be a bipartite graph and $U_1 \subseteq U$. We call $S \subseteq U$ a U_1-*essentializer* if all vertices in $U_1\backslash S$ are essential in the graph $H\backslash S$.

The next lemma basically shows that there is an approximation preserving reduction between the min-weight (resp. max-weight) vertex-stabilizer problem defined on G, and the problem of finding a suitable B_1-essentializer S that minimizes $\sum_{v\in S} w_v$ (resp. maximizes $\sum_{v\notin S} w_v$) in the weighted bipartite graph G_b.

Lemma 6 *Let $\tilde{S} \subseteq \tilde{B}$ be a B_1-essentializer of G_b that satisfies $\nu(G_b\backslash\tilde{S}) = \nu(G_b)$. Then $\tilde{S}$ corresponds to a vertex-stabilizer of G (of the same weight). Let $S \subseteq V$ be an optimal solution to a min-weight vertex-stabilizer (resp. max-weight vertex-stabilizer) instance defined on G. Then S corresponds to a B_1-essentializer in G_b (of the same weight) that satisfies $\nu(G_b\backslash S) = \nu(G_b)$.*

Proof Suppose $\tilde{S}$ is a B_1-essentializer. By construction, $\tilde{S}$ corresponds to a subset of V, and we claim that, in fact, it is a vertex-stabilizer for G. Recall that we let (B, C, D) be the EGD of G. Since $\nu(G_b\backslash\tilde{S}) = \nu(G_b)$, there is a C-perfect matching $\bar{M}$ in $G_b\backslash\tilde{S}$. It is easy to extend this perfect matching to a maximum matching M of G, by adding an arbitrary D-perfect matching and a near-perfect matching in each non-singleton component G_i of $G[B]\backslash\tilde{S}$. Note that such matchings exist due to Lemma 1. Furthermore, since M is element-disjoint from $\tilde{S}$, M is a maximum matching in $G\backslash\tilde{S}$ as well. Finally, note that by the properties of a B_1-essentializer, every vertex in $G[B_1]\backslash\tilde{S}$ is covered by M. Suppose $\tilde{S}$ is not a vertex-stabilizer. Then, it is also not an M-vertex-stabilizer, and therefore by Lemma 4 there is an M-exposed vertex $v \notin \tilde{S}$, such that there exists an M-alternating path in G from v to a vertex w in G_i. Without loss of generality, we can assume G_i is the only non-singleton component visited by this path. Note that v must be a singleton in B_2, while w is M-covered, since otherwise we would have an M-augmenting path, contradicting that M is a maximum matching. Therefore this path exists in $G\backslash\tilde{S}$ as well. By shrinking G_i into the node g_i, this path naturally corresponds to an $\bar{M}$-alternating path in G_b from v to g_i. However, switching the edges along this path, we would get a maximum matching in G_b exposing g_i, a contradiction with g_i being essential in G_b.

For the other direction, let $S \subseteq V$ be a min-weight (resp. max-weight) vertex-stabilizer. Define $\tilde{S}$ by including the vertex g_i if $S \cap G_i \neq \emptyset$ or including $v \in B_2$ if $v \in S$. By Lemma 5 we know that if $S \cap G_i \neq \emptyset$, then S includes a minimum weight vertex of G_i. It follows that S and $\tilde{S}$ have the same weight. By Theorem 1, there exists a maximum matching M of G which lies in $G\backslash S$. By Lemma 1, M matches all C vertices to a subset of B which cannot contain two vertices from the same non-singleton component of $G[B]$, so M yields a matching $\tilde{M}$ covering all C vertices of $G_b\backslash\tilde{S}$. Therefore, $\nu(G_b\backslash\tilde{S}) = \nu(G_b) = |C|$. Suppose by contradiction that a B_1-vertex g_i is not essential in $G_b\backslash\tilde{S}$. If g_i is $\tilde{M}$-exposed, then it means that there exists an M-exposed vertex in $G_i\backslash\tilde{S}$ which is a contradiction to Lemma 3. If g_i is $\tilde{M}$-covered, but it is inessential, it means that there exists an $\tilde{M}$-alternating path from an $\tilde{M}$-exposed vertex in $G_b\backslash\tilde{S}$ to g_i. Let P denote the subpath containing exactly one B_1-vertex, say $g_{i'}$ (might be different from g_i). Note that P only uses edges between B and C and contains exactly one B_1-vertex so P translates to an M-alternating path from an M-exposed vertex in G to a vertex in $G_{i'}$, in contradiction to Lemma 4. □

3.2 Integer programming description

In this section, we give an integer programming description of the set of B_1-essentializers, whose relaxation will be at the heart of our algorithms.

Given $G_b = (\tilde{B} \cup C, \tilde{E})$, with $\tilde{B} = B_1 \cup B_2$, we introduce a binary variable z_v for $v \in \tilde{B}$ to denote if v is in a B_1-essentializer S (i.e., $z_v = 1$ if $v \in S$). We also introduce a binary variable y for $v \in \tilde{B} \cup C$ with the following meaning: for $v \in \tilde{B}$, we let $y_v = 1$ denote if v is an essential node in $G_b\backslash S$; for $v \in C$ instead, we let $y_v = 1$ denote if v is always matched to an inessential node in any maximum matching of $G_b\backslash S$. For a set of vertices T, we let $y(T) = \sum_{v \in T} y_v$, and $N(T)$ denote the set of neighbors (i.e., adjacent vertices) of T. We let

$$
\begin{aligned}
P_I := \Big\{(z, y) : \quad & y_v + z_v \geq 1, && \text{for } v \in B_1 && (1)\\
& y_v + y_u + z_v \geq 1, && \text{for } \{u, v\} \in \tilde{E}, v \in B_2, u \in C && (2)\\
& y(N(A)) \geq |A| - y(A), && \text{for } A \subseteq C && (3)\\
& y(V) = |C|, && &&\\
& z \in \{0, 1\}^{\tilde{B}},\ y \in \{0, 1\}^{\tilde{B} \cup C} \Big\}. && && (4)
\end{aligned}
$$

Let us give an intuition of the meaning of the linear constraints. Inequality (1) states that a vertex in B_1 is either essential in $G_b\backslash S$ or it is removed (as required by Definition

1). Inequality (2) states that if a vertex v in B_2 is not removed then either v is essential in $G_b \backslash S$ or all of its neighbors have to be matched to inessential vertices in $G_b \backslash S$. The reason is that, if v is inessential in $G_b \backslash S$ but some neighbor u of v is matched to an essential vertex v' in some maximum matching M of $G_b \backslash S$, then it is possible to construct an even length M-alternating path between some M-exposed vertex to v', contradicting the fact that v' is essential. Inequality (3) is a translation of Hall's theorem, and states that there exists a matching between vertices in C with y-value 0 and their neighbors with y-value 1, that covers all vertices in C with y-value 0. The reason is that such vertices will always be matched to essential vertices in $G_b \backslash S$ by any maximum matching. We would like to emphasize that Inequalities (3) are crucial to have a meaningful formulation for our problem. Equality (4) basically ensures that there is a partition of vertices in C into those that will always be matched to inessential vertices and those that will always be matched to essential vertices of $G_b \backslash S$ by any maximum matching. The next lemma makes this intuition rigourous.

Lemma 7 *P_I describes the set of B_1-essentializers of the graph G_b.*

Proof Let $\tilde{S}$ be a B_1-essentializer of G_b. We set $z_v = 1$ for each $v \in \tilde{S}$, and $z_v = 0$ otherwise. For a vertex v in $(B_1 \cup B_2) \backslash \tilde{S}$, we set $y_v = 1$ if v is essential in $G_b \backslash \tilde{S}$, and $y_v = 0$ otherwise. To set the y-value of the vertices in C, we do the following. Take an arbitrary maximum matching M in $G_b \backslash \tilde{S}$, and let $\bar{C} \subseteq C$ be the vertices matched to the essential vertices in $(B_1 \cup B_2) \backslash \tilde{S}$ by M. We set $y_v = 0$ for $v \in \bar{C}$ and $y_v = 1$ otherwise. We claim that the vector (y, z) constructed in this way is a point of P_I. For each $v \in B_1$, Inequality (1) is satisfied because either $v \in \tilde{S}$, i.e., $z_v = 1$, or v is essential since $\tilde{S}$ is a B_1-essentializer, i.e., $y_v = 1$. Consider now $v \in B_2$. If v is essential or $v \in \tilde{S}$, then Inequality (2) is satisfied because either $y_v = 1$ or $z_v = 1$. Now suppose $v \notin \tilde{S}$, and v is inessential in $G_b \backslash \tilde{S}$. Then, we claim that v cannot be adjacent to a node $w \in \bar{C}$. Suppose v is adjacent to a vertex $w \in \bar{C}$, and let u be the essential vertex in $(B_1 \cup B_2)$ such that the edge $\{u, w\}$ is in M. If v is M-exposed, then by switching the edge $\{u, w\}$ with the edge $\{v, w\}$, we would obtain a maximum matching in $G_b \backslash \tilde{S}$ exposing u, a contradiction since u is essential. Therefore, v is M-covered. However, since v is inessential, there exists another maximum matching M_v in $G_b \backslash \tilde{S}$ exposing v. The symmetric difference $M_v \Delta M$ yields an even-length path P ending at an M-exposed vertex k in B. If P contains the edge $\{u, w\}$, then the subpath of P from u to k is also an even-length M-alternating path, and by switching the edges along this path we obtain a maximum matching in $G_b \backslash \tilde{S}$ exposing u, a contradiction. So P does not contain the edge $\{u, w\}$. However, we can add to P the edges $\{u, w\}$ and $\{w, v\}$, and again obtain a path from u to k that is an even-length M-alternating path, a contradiction. It follows that v cannot be adjacent to any node $w \in \bar{C}$, and therefore Inequality (2) is satisfied as $y_w = 1$ for all neighbors of v. The vector y satisfies Inequality (3), since for any $u \in A$ with $y_u = 0$, we have $y_w = 1$ for at least one neighbor w of u. Finally, y also satisfies Equality (4) by construction.

Now let (y, z) be a point of P_I. Define the set $\tilde{S} := \{v : z_v = 1\}$. Furthermore, let $\bar{B} := \{v \in \tilde{B} : y_v = 1\}$, and $\bar{C} := \{u \in C : y_u = 0\}$. Note that for every $v \in B_1 \backslash \tilde{S}$, we have $y_v = 1$ by Inequality (1), so $v \in \bar{B}$. Moreover, for any set $A \subseteq \bar{C}$, we have $y(N(A)) \geq |A|$. Therefore, by Hall's condition, there exists a matching M in G_b which matches each vertex of $\bar{C}$ to $\bar{B}$. Since for any $u \in C \backslash \bar{C}$ we have that $y_u = 1$, by

Equality (4) it follows that $|\bar{C}| = |\bar{B}|$. Since (y, z) satisfies Inequality (2), each vertex v in $\tilde{B} \backslash \bar{B}$ which is not in $\tilde{S}$, i.e., $z_v = 0$, cannot be a neighbor of $u \in \bar{C}$. It follows that vertices in $\bar{C}$ are adjacent only to vertices in $\bar{B}$ in $G_b \backslash \tilde{S}$. Since as we already mentioned $|\bar{C}| = |\bar{B}|$, this implies that in every maximum matching of $G_b \backslash \tilde{S}$ a vertex $v \in \bar{B} \backslash \tilde{S}$ is covered, i.e., it is essential. □

Finally, we note that the problem of finding a B_1-essentializer S that maximizes $\sum_{v \notin S} w_v$, or minimizes $\sum_{v \in S} w_v$, can be formulated respectively as

$$\max \left\{ \sum_{v \in \tilde{B}} w_v(1 - z_v) : (z, y) \in P_I \right\}, \quad \text{and} \quad \min \left\{ \sum_{v \in \tilde{B}} w_v z_v : (z, y) \in P_I \right\}. \quad (5)$$

3.3 Properties of the relaxation

We denote by P_f the polytope obtained by relaxing the binary constraints of P_I, i.e., replacing them with $0 \leq z \leq 1$ and $0 \leq y \leq 1$. When dealing with fractional points, Inequality (3) does not correspond to Hall's theorem anymore, but it naturally ensures the existence of a *flow* of value $|C| - y(C)$ from vertices in C to vertices in $\tilde{B}$. Among other things, this also implies that although this set contains exponentially (in the size of G_b) many inequalities, we can separate over them in polynomial time.

Lemma 8 *Construct a directed network $\mathcal{N} = (V_{\mathcal{N}}, A_{\mathcal{N}})$ from graph $G_b = (\tilde{B} \cup C, \tilde{E})$ with $V_N = \tilde{B} \cup C \cup \{s, t\}$ and $A_{\mathcal{N}} = \{(s, u) : u \in C\} \cup \{(v, t) : v \in \tilde{B}\} \cup \tilde{E}$ where the edges in $\tilde{E}$ are oriented from C to $\tilde{B}$. Let $(z, y) \in P_f$. Assign y_v amount of capacity to each arc (v, t), $(1 - y_u)$ amount of capacity to each arc (s, u), and ∞ capacity to arcs in $\tilde{E}$. Then, there exists a maximum $s - t$ flow in $\mathcal{N}$ of value $y(\tilde{B}) = |C| - y(C)$.*

Proof We show an existence of a max $s - t$ flow in $\mathcal{N}$ of value $y(\tilde{B}) = |C| - y(C)$ by showing that the minimum-cut capacity in $\mathcal{N}$ is $|C| - y(C)$. Let $\mathcal{S}$ be a min-cut in $\mathcal{N}$. Since $\{s\}$ is a $s - t$ cut with capacity $|C| - y(C)$, the capacity of $\mathcal{S}$ is at most $|C| - y(C)$. Let $A = \mathcal{S} \cap C$. Since $\mathcal{S}$ cannot contain an arc in $\tilde{E}$ (they have capacity ∞), $N(A) \subseteq \mathcal{S}$, so the capacity of $\mathcal{S}$ is at least

$$\begin{aligned} \sum_{u \in C \backslash A} 1 - y_u + \sum_{v \in N(A)} y_v &= |C \backslash A| - y(C \backslash A) + y(N(A)) \\ &\geq |C \backslash A| - y(C \backslash A) + |A| - y(A) = |C| - y(C). \end{aligned}$$

Here the inequality follows from the fact that (y, z) is a feasible solution and satisfies Inequality (3). Hence a minimum cut has capacity $|C| - y(C)$ and by Max-Flow Min-Cut Theorem the desired flow exists. □

Exploiting the structure of this flow, we can derive useful properties on the extreme points of P_f. Let F be a maximum flow in $\mathcal{N}$ of value $y(\tilde{B}) = |C| - y(C)$ from s to t. Let F_e denote the flow on the arc $e = (u, v) \in A_{\mathcal{N}}$. We can assume that the support of F, ignoring orientations, induces an acyclic subgraph on the set $\tilde{E}$, as otherwise we can do the usual operation of increasing (resp. decreasing) the flow value by $\varepsilon > 0$

on the even (resp. odd) edges of a cycle, until we obtain a new acyclic flow. So the support of F, restricted to the set $\tilde{E}$ and ignoring orientations, gives a collection of disjoint trees $T_1, T_2, \ldots, T_r$ on $\tilde{B} \cup C$. The following lemma holds.

Lemma 9 *Let (y, z) be an extreme point of P_f, and let T_i be any tree with vertices in $\tilde{B} \cup C$, given by the support of the flow F in network $\mathcal{N}$. Then, there is at most one vertex v in T_i with $0 < y_v, z_v < 1$.*

Proof Suppose that there are two vertices v_1, v_2 with $0 < y_{v_1}, z_{v_1}, y_{v_2}, z_{v_2} < 1$ in T_i. Note that $v_1, v_2 \in \tilde{B}$. Using F, we will construct two feasible solutions $(y^+, z^+), (y^-, z^-)$ to P_f such that $(y, z) = \frac{(y^+,z^+)+(y^-,z^-)}{2}$ which is a contradiction with (y, z) being an extreme point. Let $P = \{e_1, e_2, \ldots, e_{2l}\}$ be unique (undirected) path in T_i from v_1 to v_2. All the arcs corresponding to edges in P have non-zero flow in F. Let $\epsilon = \min_{e \in P} F_e$. Define F^+, z^+, y^+ and F^-, z^-, y^- as follows

$$F_e^+ = \begin{cases} F_e + \epsilon & \text{if } e = e_{2i}, \\ F_e - \epsilon & \text{if } e = e_{2i-1}, \\ F_e & \text{otherwise.} \end{cases} \quad z_v^+ = \begin{cases} z_v + \epsilon & \text{if } v = v_1, \\ z_v - \epsilon & \text{if } v = v_2, \\ z_v & \text{otherwise.} \end{cases} \quad y_v^+ = \begin{cases} y_v - \epsilon & \text{if } v = vv_1, \\ y_v + \epsilon & \text{if } v = v_2, \\ y_v & \text{otherwise.} \end{cases}$$

$$F_e^- = \begin{cases} F_e - \epsilon & \text{if } e = e_{2i}, \\ F_e + \epsilon & \text{if } e = e_{2i-1}, \\ F_e & \text{otherwise.} \end{cases} \quad z_v^- = \begin{cases} z_v - \epsilon & \text{if } v = v_1, \\ z_v + \epsilon & \text{if } v = v_2, \\ z_v & \text{otherwise.} \end{cases} \quad y_v^- = \begin{cases} y_v + \epsilon & \text{if } v = v_1, \\ y_v - \epsilon & \text{if } v = v_2, \\ y_v & \text{otherwise.} \end{cases}$$

We claim that (y^+, z^+) and (y^-, z^-) are in P_f. Note that for each $v \in \tilde{B}$, $z^+ + y_v^+ = z_v^- + y_v^- = z_v + y_v$, so Inequalities (1) and (2) are satisfied. Moreover $y^+(V) = y^-(V) = y(V) = |C|$, so Equality (4) is satisfied as well. Furthermore, note that we can still send a flow from s to t of value $|C| - y(C)$, as we can send flow of $1 - y_u$ on arc (s, u) for each $u \in C$, then send flow of F_e^+ (F_e^-) on each arc $e \in \tilde{E}$, and then flow of y_v^+ (y_v^-) on arc (v, t) for each $v \in \tilde{B}$. Therefore, for any cut $\mathcal{S}$ induced by a subset $A \subseteq C$ and its neighbors $N(A)$ we have that the capacity is at least:

$$\sum_{u \in C \setminus A} 1 - y_u + \sum_{v \in N(A)} y_v = |C \setminus A| - y(C \setminus A) + y(N(A)) \geq |C| - y(C)$$

which can be rearranged as $y(N(A) \geq |A| - y(A)$, showing that Inequality (3) is satisfied. Note that $(y, z) = \frac{(y^+,z^+)+(y^-,z^-)}{2}$, which is a contradiction. □

Corollary 1 *Let (y, z) be an extreme point of P_f, and let T_i be any tree with vertices in $\tilde{B} \cup C$, given by the support of the flow F in network $\mathcal{N}$. Then, there is at most one vertex $v \in \tilde{B}$ that is a leaf of T_i.*

Proof Let v be a leaf of T_i in $\tilde{B}$ and let u be a neighbor of v in T_i which must be in C. Note that $y_v + z_v \geq 1 - y_u$ by Inequality (1) or (2). Furthermore, all the arcs from $\tilde{B}$ to t are saturated by F, as well as all the arcs from s to C. It follows that v sends a flow of y_v to t and u receives a flow of $1 - y_u$ from s. Still, since the arc (u, v) has positive flow, $y_v > 0$. Now, if $z_v = 0$ then the capacity of the arc (v, t) which is y_v, is enough for routing all the flow coming to u from s which is $1 - y_u$, and since v receives flow only from u (being a leaf of T_i), necessarily $y_v = 1 - y_u$ and u is the only other leaf of T_i. If instead u is not a leaf and it sends some flow to some other vertex $v' \in \tilde{B}$, it means that z_v must be non-zero. This argument shows that any leaf of T_i in $\tilde{B}$ with a neighbor u that is not a leaf in T_i has $y_v, z_v > 0$. Therefore there cannot exist two leaves of T_i in $\tilde{B}$ by Lemma 9. □

Lemma 10 *Let (z, y) be an extreme point of P_f. There exists a maximum matching in G_b between the set of vertices $\{v \in \tilde{B} : y_v > 0\}$ and the set of vertices $\{u \in C : y_u < 1\}$ of cardinality $|\{v \in \tilde{B} : y_v > 0\}|$.*

Proof Let $B' = \{v \in \tilde{B} : y_v > 0\}$. Each $v \in B'$ belongs to a tree T_i. By Corollary 1, there is at most one leaf that is a B'-vertex in each tree. Root each tree T_i at a leaf that is a B'-vertex, if there exists any, or at a C-vertex otherwise. Then match each B' vertex to one of its children in this rooted tree. Note that this is always possible since in each of these rooted trees, there does not exists a B'-leaf which is not a root. So there exists a matching from B' to C-vertices lying in some tree, i.e., vertices with $1 - y_u > 0$, with size $|B'|$. □

3.4 Approximation algorithm for max-weight vertex-stabilizer

Given a graph $G = (V, E)$ with weights $w_v \geq 0 \ \forall v \in V$, we construct from G a weighted bipartite graph $G_b = (\tilde{B} \cup C, \tilde{E})$, with $\tilde{B} = B_1 \cup B_2$, as described in the beginning of Sect. 3. We then apply Algorithm 2 that relies on solving the LP relaxation of the maximization IP in (5).

Algorithm 2

1. Let $(z^*, y^*) \leftarrow$ optimal extreme point of $\max\{\sum_{v \in \tilde{B}} (1 - z_v) : (z, y) \in P_f\}$.
2. Set $B_+ := \{v \in \tilde{B} : 0 < y_v^*\}$; $B_0^1 := \{v \in \tilde{B} : y_v^* = 0, z_v^* = 1\}$; $B_0^f := \{v \in \tilde{B} : y_v^* = 0, 0 < z_v^* < 1\}$.
3. **If** $w(B_+) \leq w(B_0^f)$ **then** set $S := (B_+ \cup B_0^1)$, **else** set $S := (B_0^f \cup B_0^1)$.
4. **While** $\nu(G_b \backslash S) < |C|$ **do**: find $v \in S$ such that $\nu(G_b \backslash (S \backslash \{v\})) > \nu(G_b \backslash S)$, and set $S := S \backslash \{v\}$.

Theorem 4 *There is a polynomial-time LP-based 2-approximation algorithm for the max-weight vertex-stabilizer problem.*

Proof We consider the set S output by Algorithm 2. Note that if S is a B_1-essentializer *and* $\nu(G_b \backslash S) = \nu(G_b)$, then it corresponds to a vertex-stabilizer in G by Lemma 6. Still, Lemma 6 implies that to prove the claimed approximation guarantee, it is enough to prove that S is a 2-approximated solution for the problem of finding a B_1-essentializer for G_b that maximizes the weight of the non selected vertices.

First, we show that (a) $\nu(G_b \backslash S) = \nu(G_b)$ and (b) every vertex in $B_1 \backslash S$ is essential in $G_b \backslash S$, i.e., S is a B_1-essentializer. Note that (a) holds by construction after step 4 (recall that $\nu(G_b) = |C|$ and $S \cap C = \emptyset$, therefore it is always possible to perform step 4 until the while condition is not satisfied anymore). Moreover, all vertices added in step 4 are essential vertices. We are left with (b). Define $C_f = \{u \in C : y^*_u < 1\}$. Furthermore, partition the set of vertices in $\tilde{B}$ in 4 sets: B_+, B^1_0, B^f_0 and $B^0_0 := \{v \in \tilde{B} : z^*_v = 0 \ \& \ y^*_v = 0\}$ (the definition of the first 3 sets is given in Algorithm 2). Note that the vertices in B_1 are either in B^1_0 or B_+, so if $S = B_+ \cup B^1_0$, then $G_b \backslash S$ does not contain any B_1 vertex, and we have nothing to show. Suppose instead $S = B^f_0 \cup B^1_0$. Note that there does not exist any edge between $v \in B^0_0$ and $u \in C_f$, because $y^*_v + z^*_v = 0$ holds for v and $y^*_u < 1$ holds for u, and therefore Inequality (2) will be violated for the edge $\{v, u\}$, contradicting the feasibility of (z^*, y^*). Therefore, the neighbors of vertices C_f in $G_b \backslash S$ are vertices in B_+ and by Lemma 10, we know that there is a matching between C_f and B_+ covering all vertices in B_+. Since every maximum matching in $G_b \backslash S$ covers all the vertices in C, it must cover all vertices in C_f, therefore it must be the case that $|C_f| = |B_+|$ and every maximum matching in $G_b \backslash S$ covers all the vertices in B_+, i.e., all the vertices in B_+ are essential. Since $(B_1 \backslash S) \subseteq B_+$, the result follows.

To conclude the proof, we argue that the weight of the vertices in $G_b \backslash S$ is at least $\frac{1}{2}$ the optimal value of the LP. Let $w_0 = w(B^0_0)$, $w_1 = w(B_+)$, $w_2 = w(B^f_0)$. Note that the weight of the vertices in the graph $G_b \backslash S$ is at least $w_0 + \max(w_2, w_1)$ which is at least half of $w_0 + w_1 + w_2 = \sum_{v \in \tilde{B}} w_v - \sum_{v : z^*_v = 1} w_v$, which is clearly an upper bound on the optimal value of the LP. □

3.5 Approximation algorithm for min-weight vertex-stabilizer

Given a graph $G = (V, E)$ with weights $w_v \geq 0 \ \forall v \in V$, we construct a weighted bipartite graph $G_b = (\tilde{B} \cup C, \tilde{E})$, with $\tilde{B} = B_1 \cup B_2$ obtained from G as described in the beginning of Sect. 3. We then apply Algorithm 3 that relies on solving the LP relaxation of the minimization IP in (5).

Algorithm 3

1. Solve the LP: $\min \left\{ \sum_{v \in \tilde{B}} w_v z_v : (z, y) \in P_f \right\}$ to get an extreme point optimal solution (z, y), and set $S := \{v : z_v \geq \frac{1}{|C|+1}\}$.
2. **While** $\nu(G_b \backslash S) < |C|$ **do**: find $v \in S$ such that $\nu(G_b \backslash (S \setminus \{v\})) > \nu(G_b \backslash S)$, and set $S := S \backslash \{v\}$.

Theorem 5 *There is a polynomial-time LP-based $(\gamma + 1)$-approximation algorithm for the min-weight vertex-stabilizer problem, where γ is the size of the Tutte-set of G.*

Proof We consider the set S output by Algorithm 3. As for the max-weight case, due to Lemma 6 and step 2 of the algorithm, to prove the theorem it is enough to show that S is a $(|C| + 1)$-approximated solution for finding a B_1-essentializer for G_b that minimizes the weight of the selected vertices. Trivially, $w(S) \leq (|C|+1) \sum_{v \in \tilde{B}} w_v z_v$, therefore the approximation factor guarantee holds. It remains to show that S is in fact a B_1-essentializer for G_b.
Let $\tilde{S}$ be the set S *before* executing step 2 of the algorithm. We will prove that each $v \in B_1 \backslash \tilde{S}$ is essential in $G_b \backslash \tilde{S}$. This is enough, since every vertex added back in step

2 will be essential by construction, and this addition cannot make any vertex in B_1 inessential. Let us assume by contradiction that $v_0 \in B_1 \setminus \tilde{S}$ is inessential in $G_b \setminus \tilde{S}$. In this case, if we apply Edmonds' Blossom Algorithm [8] in $G_b \setminus \tilde{S}$, we can find a maximum matching M that exposes v_0 and a so-called *frustrated tree* $T = (V_T, E_T)$ containing v_0 with the following properties: (i) $|E_T \cap M| = |V_T \cap C|$, and all vertices in $V_T \setminus \{v_0\}$ are covered by M, and (ii) the neighbors of the set of vertices $V_T \cap \tilde{B}$ in $G_b \setminus \tilde{S}$ are all in the tree T (we refer to [7,8] for details). Note that the neighbors of $V_T \cap \tilde{B}$ in $G_b \setminus \tilde{S}$ are the same as the neighbors of $V_T \cap \tilde{B}$ in G_b, i.e., $N(V_T \cap \tilde{B}) = V_T \cap C$ as $\tilde{S} \subseteq \tilde{B}$. Feasibility of (z, y) implies that for each matching edge $\{u, v\} \in M$, we have $y_u + y_v + z_v \geq 1$. Since $\tilde{S}$ removed all vertices with z-value $\geq \frac{1}{|C|+1}$, for each edge $\{u, v\} \in M$, $y_u + y_v > 1 - \frac{1}{|C|+1}$. Let $M_T := M \cap E_T$. We have

$$
\begin{aligned}
y(V_T) = y_{v_0} + \sum_{\{u,v\} \in M_T} (y_u + y_v) &> \left(1 - \frac{1}{|C|+1}\right) + |M_T| \left(1 - \frac{1}{|C|+1}\right) \\
&= |M_T| + 1 - \frac{|M_T|+1}{|C|+1} \geq |M_T|,
\end{aligned}
$$

where the first inequality follows from the Inequality (1) associated to v_0, and the last inequality follows from the fact that $|M_T| \leq |C|$. Furthermore, for set $A = V_T \cap C$, since $|M_T| = |A|$ by (i), we have

$$
y(A) + y(N(A) \cap V_T) = y(V_T) > |M_T| = |A|. \tag{6}
$$

If we consider the directed network $\mathcal{N}$ and the $s-t$ flow as in Lemma 8, (6) says that the capacity $y(N(A) \cap V_T)$ of the arcs between t and $N(A) \cap V_T$ is strictly larger than the flow sent on the arcs from s to A (that can be at most $|A| - y(A)$). Since a maximum flow necessarily saturates *all* the arcs from $N(A) \cap V_T$ to t, there is a neighbor of $(N(A) \cap V_T)$ which is not in A who sends positive flow to some vertex in $N(A) \cap V_T$, but this contradicts property (ii) of T, as $N(N(A) \cap V_T) = N(\tilde{B} \cap V_T) = A$. $\square$

We remark here that in Sect. 5 we will show a tight lower bound of $\Omega(\gamma)$ on the integrality gap of the minimization IP in (5) that holds even on graphs with *constant* degree. However, we can develop an algorithm whose approximation ratio is bounded by the maximum degree of a vertex in G, if we know the set of essential vertices in the final stable graph (our reduction in Theorem 7 shows that also this problem is NP-hard). In particular, let δ be the maximum degree of an inessential vertex of G that is a singleton component in $G[B]$ (i.e., the maximum degree of a node in B_2 in the graph G_b).

Theorem 6 *There is a δ-approximation algorithm for the min-weight vertex-stabilizer problem, if we know the set of essential vertices in the final stable graph.*

This translates into knowing which subset $B' \subseteq \tilde{B}$ of vertices of G_b must have y-value 1 in our formulation (5). In this setting, we therefore add to P_f the following constraints: $y_v = 1$ for $v \in B'$, and $y_v = 0$ for $v \in \tilde{B} \setminus B'$. We call the resulting LP problem $(\mathcal{P}_1)$. To find a good integral solution to $(\mathcal{P}_1)$, we introduce a new optimization problem ($\mathcal{P}_2$) which in fact corresponds to a minimum cost flow problem, as follows. We define a new weight vector $\overline{w}$ for vertices in C as $\overline{w}_u = \sum_{v \in N(u) \cap (B_2 \setminus B')} \frac{w_v}{\delta}$. Then we remove all vertices in $\tilde{B}$ but B', obtaining graph G'_b. For a set of vertices T, let $N'(T)$ denote the set of neighbors of T in G'_b (while $N(T)$ denotes the neighbors in G_b). We introduce a variable f_u $\forall u \in C$, and let ($\mathcal{P}_2$) be the following problem:

$$\min w(B_1 \setminus B') + \sum_{u \in C} \overline{w}_u f_u \qquad (\mathcal{P}_2)$$

$$\text{s.t.} |N'(A)| \geq \sum_{u \in A} f_u, \qquad \forall A \subseteq C \qquad (7)$$

$$\sum_{u \in C} f_u = |B'|,$$

$$0 \leq f_u \leq 1, \qquad \forall u \in C. \qquad (8)$$

Apart from the constant term $(w(B_1 \setminus B'))$ in the objective function, this problem corresponds to a minimum cost flow problem in a directed network $\overline{\mathcal{N}}$ constructed from G'_b via a similar process as in Lemma 8. Formally: (i) add a vertex s and for each vertex $u \in C$ add an arc (s, u) with capacity 1 and cost $\overline{w}_u$, (ii) orient all the edges from C to B' in G'_b and set their capacity to infinity and their cost to zero, (iii) add a vertex t and for each $v \in B'$ add an arc (v, t) with capacity 1 and cost 0. An optimal solution to ($\mathcal{P}_2$) can be mapped to a minimum cost $s - t$ flow of value $|B'|$ in $\overline{\mathcal{N}}$, and vice versa. Therefore, since all the capacities are integral, we know that ($\mathcal{P}_2$) has an optimal *integral* solution.

The following lemma maps solutions of these two optimization problems.

Lemma 11 *An optimal solution of ($\mathcal{P}_1$) can be mapped into a solution of ($\mathcal{P}_2$) with no greater weight. An integral solution of ($\mathcal{P}_2$) can be mapped into an integral solution of ($\mathcal{P}_1$) whose weight is at most a δ-factor larger.*

Proof We first prove that an optimal solution of ($\mathcal{P}_1$) can be mapped into a solution of ($\mathcal{P}_2$) with no greater weight. Let (z, y) be an optimal solution of ($\mathcal{P}_1$). Define $f_u = 1 - y_u$ for $u \in C$. For $A \subseteq C$, $y(N(A)) \geq |A| - y(A) = \sum_{v \in A}(1 - y_v) = f(A)$, and since $y(N(A)) = |N(A) \cap B'| = |N'(A)|$, we have that f satisfies Inequality (7). Inequality (8) is also satisfied since $f(C) = |C| - y(C)$ which is equal to $y(\tilde{B}) = |B'|$ by Equality (4). Note that optimality of (z, y) implies $z_v = 0$ for $v \in B'$, $z_v = 1$ for $v \in B_1 \setminus B'$, and $z_v = \max_{u \in N(v)} (1 - y_u)$ for $v \in B_2 \setminus B'$. We get

$$\begin{aligned}
\sum_{v \in \tilde{B}} w_v z_v &= w(B_1 \setminus B') + \sum_{v \in B_2 \setminus B'} w_v \max_{u \in N(v)} (1 - y_u) \\
&\geq w(B_1 \setminus B') + \sum_{v \in B_2 \setminus B'} w_v \sum_{u \in N(v)} \frac{1 - y_u}{\delta} \\
&= w(B_1 \setminus B') + \sum_{u \in C} \sum_{v \in N(u) \cap (B_2 \setminus B')} \frac{w_v}{\delta}(1 - y_u) = w(B_1 \setminus B') + \sum_{u \in C} \overline{w}_u f_u.
\end{aligned}$$

We now prove that an integral solution of ($\mathcal{P}_2$) can be mapped into an integral solution of ($\mathcal{P}_1$) whose weight is at most a δ-factor larger.

Given an integral f, construct an integral solution (z, y) of ($\mathcal{P}_1$) as follows. (i) Set $y_u = 1 - f_u$ for $u \in C$, $y_v = 1$ for $v \in B'$, and $y_v = 0$ for $v \in \tilde{B} \backslash B'$. (ii) Set $z_v = \max_{u \in N(v)}(1 - y_u)$ for $v \in B_2 \backslash B'$. (iii) Set $z_v = 1 - y_v$ for $v \in B_1 \backslash B'$. (iv) Set $z_v = 0$ for $v \in B'$. Clearly this solution is integer. We now show feasibility. Inequality (1) and (2) hold trivially. Inequality (3) is satisfied, since for each $A \subseteq C$, $y(N(A)) = N'(A)$ and therefore this inequality is equivalent to inequality (7), satisfied by f. Inequality (4) holds as $y(C) = |C| - f(C) = |C| - |B'|$ and $y(\tilde{B}) = |B'|$. Finally, we get:

$$\sum_{v \in \tilde{B}} w_v z_v = \sum_{v \in B_1 \backslash B'} w_v + \sum_{v \in B_2 \backslash B'} w_v (\max_{u \in N(v)} f_u) \leq w(B_1 \backslash B') + \sum_{v \in B_2 \backslash B'} w_v \sum_{u \in N(v)} f_u$$
$$\leq \delta \left(w(B_1 \backslash B') + \sum_{u \in C} \sum_{v \in N(u) \cap (B_2 \backslash B')} \left(\frac{w_v}{\delta}\right) f_u \right) = \delta \left(w(B_1 \backslash B') + \sum_{u \in C} \overline{w}_u f_u \right).$$

□

We can now replace step 1 of Algorithm 3 with solving ($\mathcal{P}_2$) and mapping the solution into an integral solution of ($\mathcal{P}_1$) (we keep step 2). Combining Lemma 11 with Lemma 6, one can easily see that this new algorithm yields a proof of Theorem 6.

4 Hardness

In this section we will prove the following theorem.

Theorem 7 *The min-weight vertex-stabilizer problem and the max-weight vertex-stabilizer problem are NP-hard, even if there are only 2 distinct weights.*

We will show that the problems are *NP*-hard by using a reduction from *minimum weight satisfiability* (MIN SAT) problem, that is NP-hard [12]. A MIN SAT instance in the Boolean variables $x_1, x_2, \ldots, x_n$ is composed of a collection of *clauses* $K_1, \ldots, K_m$. Each clause has the form $(z_1 \vee z_2 \ldots \vee z_{k_i})$ for some $k_i \geq 1$ where each z_j is called a *literal* and is either a variable x_l or its negation $\overline{x}_l$. The goal is to assign each $x_1, x_2, \ldots, x_n$ a value $true$ or $false$ so that the total number of satisfied clauses is minimized. A clause is said to be satisfied if at least one of its literals is assigned with value $true$. We start with proving the following.

Theorem 8 *Finding a minimum weight B_1-essentializer that preserves the cardinality of a maximum matching is* NP*-hard.*

Proof Given a MIN SAT instance, we define $G_b = (\tilde{B} \cup C, \tilde{E})$ with $\tilde{B} = B_1 \cup B_2$ as follows (see Fig. 3):

– Let $C = \cup_{j=1}^{n} \{T_j, F_j\}$, i.e., C has one vertex for each possible truth assignment of every variable x_j.

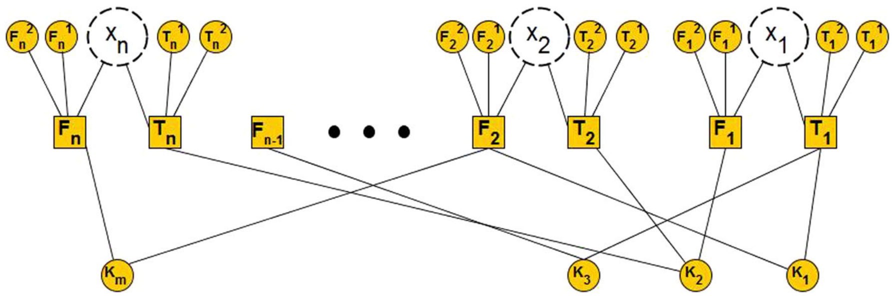

Fig. 3 Graph G_b corresponding to a MIN SAT instance with $K_1 = (x_1 \vee \overline{x}_2)$, $K_2 = (\overline{x}_1 \vee x_2 \vee x_n)$, $K_3 = (x_1 \vee \overline{x}_{n-1})$, and $K_m = (\overline{x}_2 \vee \overline{x}_n)$. The set B_1 is shown by *dotted circles*, set C is shown by *square vertices*, and set B_2 is shown by *solid circles*

- Let $B_1 = \cup_{j=1}^{n} x_j$, i.e., B_1 has one pseudonode associated with each variable.
- Let $B_2 = \cup_{i=1}^{m} K_i \cup_{j=1}^{n} \{T_j^1, T_j^2, F_j^1, F_j^2\}$, i.e., B_2 has a vertex for each clause and four extra vertices for each variable x_j.
- The edge set $\tilde{E}$ is defined as follows:
  For each T_j put edges (T_j, x_j), (T_j, T_j^1), (T_j, T_j^2) and one edge to each clause K_i containing x_j.
  For each F_j put edges (F_j, x_j), (F_j, F_j^1), (F_j, F_j^2) and one edge to each clause K_i containing $\overline{x}_j$.
- Define $w_v = 2n + m + 1$ for each $v \in B_1$, and $w_v = 1$ for each $v \in B_2$.

Claim There is a solution of the MIN SAT instance of value at most k if and only if there is a B_1-essentializer for G_b of weight at most $2n+k$ that preserves the cardinality of a maximum matching.

Suppose to be given a truth assignment satisfying at most k clauses. We construct a B_1-essentializer $\tilde{S}$ as follows. For each variable x_j, define $\mu(j) = T_j$ if the variable is set to $true$, and $\mu(j) = F_j$ otherwise. For each variable x_j, include in $\tilde{S}$ all neighbors of $\mu(j)$ except the vertex x_j. First, note that $\nu(G_b \backslash \tilde{S}) = |C| = 2n$. This is because for each $j \in [n]$, $\mu(j)$ can be matched to x_j and $\{T_j, F_j\} \backslash \{\mu(j)\}$ can be matched to one of T_j^1, T_j^2 or F_j^1, F_j^2. Furthermore, for each $j \in [n]$, x_j is the only neighbor of $\mu(j)$, so any maximum matching in $G_b \backslash \tilde{S}$ covers x_j, therefore all x_j's are essential. Hence, the set $\tilde{S}$ is a B_1-essentializer. Now let us calculate the weight of $\tilde{S}$. For each $\mu(j)$, we have to remove either $\{T_j^1, T_j^2\}$ or $\{F_j^1, F_j^2\}$, and all clauses connected to $\mu(j)$: these clauses are precisely the clauses satisfied by the assignment. So the weight of $\tilde{S}$ is at most $2n + k$.

Now let S^* be a B_1-essentializer of weight at most $2n + k$. Note that S^* does not contain any pseudonode, because the weight of a pseudonode is $> 2n + m \geq 2n + k$. Consider a maximum matching M of $G_b \backslash S^*$. Let $M(j)$ be the vertex in C that is matched to vertex x_j by M (x_j is essential so $M(j)$ is well-defined). Suppose $M(j) = T_j$ (the other case is similar). Then T_j^1 and T_j^2 must be in S^* as these two vertices are exposed by M and they have M-alternating paths to x_j. Furthermore, w.l.o.g. we can assume that any clause K_i which is a neighbor of T_j is in S^*. Suppose not, then K_i has to be essential, so it will be matched to some $u = T_{j'}$ or $u = F_{j'}$ for some $j' \neq j$. However, then either $\{T_{j'}^1, T_{j'}^2\}$ or $\{F_{j'}^1, F_{j'}^2\}$ must be in S^*. By adding K_i to S^* and removing these two vertices from S^* we obtain another B_1-essentializer with a smaller weight. This argument shows that any minimum weight B_1-essentializer yields a matching that matches each x_j to $M(j) \in \{T_j, F_j\}$, and it includes all vertices in $N(M(j))\backslash\{x_j\}$. It follows that $| \cup_j N(M(j))\backslash\{x_j\}| \leq 2n + k$, and therefore $|(\cup_j N(M(j))\backslash\{x_j\}) \cap \{K_1, \ldots, K_m\}| \leq k$. Define a truth assignment by setting x_j to true if $M(j) = T_j$ and to false otherwise. The number of clauses that are satisfied are exactly given by $|(\cup_j N(M(j))\backslash\{x_j\}) \cap \{K_1, \ldots, K_m\}| \leq k$. □

Our proof shows a reduction from MIN SAT to finding a minimum weight B_1-essentializer that preserves the cardinality of a maximum matching. It follows from Lemma 6 that finding a min-weight vertex-stabilizer is NP-hard as well. Since minimizing $\sum_{v\in S} w_v$ is equivalent to maximizing $\sum_{v\notin S} w_s$, the hardness holds as well for max-weight vertex-stabilizer. That is, we proved Theorem 7.

5 Integrality gap

In this section, we give lower bounds on the integrality gap of the LP relaxations we used, which show that our analysis are almost tight.

5.1 Integrality gap of the maximization formulation in (5)

We here prove a $\frac{3}{2}$ lower bound on the integrality gap of the maximization formulation in (5). Consider $G_b = (\tilde{B} \cup C, \tilde{E})$ which is a complete bipartite graph with $\tilde{B} = \{v_0, v_1, \ldots, v_p\}$ and $C = \{u_1, u_2\}$. We let $w_{v_0} = \sum_{i=1}^{p} w_i = W$, and $B_1 = \{v_0\}$. An optimal integral solution removes all but one vertices in $\tilde{B}\backslash\{v_0\}$, that is $y_{v_0} = y_{v_1} = 1$ and all other y-values are set to 0, $z_{v_0} = z_{v_1} = 0$ and all other z-values are set to 1, and has objective function value $W(1 + \frac{1}{p})$. However, a fractional solution can set: $y_{v_0} = 1$, $y_{u_1} = y_{u_2} = \frac{1}{2}$, $z_{v_1} = z_{v_2} = \cdots = z_{v_p} = \frac{1}{2}$, and set all other variables to 0, and has objective function value $W(1 + \frac{1}{2})$. For $p \to \infty$, the integrality gap ratio $\to \frac{3}{2}$.

5.2 Integrality gap of the minimization formulation in (5)

We here show that the integrality gap of the minimization formulation in (5) is $\Omega(n)$, for a graph G' with $20n - 1$ vertices and size of the Tutte-set $\gamma = |C| = 6n - 1$. This shows that our approximation ratio matches the integrality gap up to a constant factor.

In order to present the example achieving large integrality gap, we first need to address some results on bipartite expander graphs.

Definition 2 A bipartite multigraph G with bipartition U and V is called a (K, A) *vertex expander* if for any set $S \subseteq U$ with size at most K, the neighborhood $N(S)$ is of size at least $A \cdot |S|$

Let $Bip_{N,D}$ be the set of D-regular bipartite graphs with N vertices on each side. Bassalygo [2] proved the following theorem (for proof see Chapter 4 of [21]).

Theorem 9 [2] *For every constant D, there exists a constant $\alpha > 0$ such that for all N, a uniformly random graph from $Bip_{N,D}$ is an $(\alpha N, D-2)$ vertex expander with probability at least $1/2$.*

Using Theorem 9, we obtain the following corollary.

Corollary 2 *Given $D \geq 4$ and $N \in \mathbb{N}$, there exists a bipartite graph G' with bipartition $(U \cup U')$ and V', such that $|U| = |U'| = |V'| = N$, the maximum degree of a vertex is $2D$, and with the property that for any set $S \subseteq U \cup U'$ of size at most αN, $|N(S)| \geq |S|$ where α is a positive constant.*

Proof By Theorem 9 and choosing $D \geq 4$ and $N \in \mathbb{N}$, there exists a D-regular bipartite graph G with bipartition U and V, with $|U| = |V| = N$, which is a $(\alpha N, D-2)$ vertex expander for some constant $\alpha > 0$. Let graph G' be obtained from G by adding a copy u' of each vertex u in U to G, and connecting u' to the same set of vertices as u. Let U' denote the set containing the copies of the vertices in U, and $V' = V$ denote the other partition of G'. So the degree of each vertex in $U \cup U'$ is D and the degree of each vertex in V' is $2D$.

We claim that G' is $(\alpha N, 1)$ vertex expander. Let $S \subseteq U \cup U'$ have size at most αN. Define set S' to be obtained from S by including u in S' if at least one of u or u' is in S. Note that $\frac{|S|}{2} \leq |S'| \leq |S| \leq \alpha N$ and $N(S) = N(S')$ by construction of G'. Since G is $(\alpha N, D-2)$ vertex expander and $D \geq 4$, we have

$$|N(S)| = |N(S')| \geq |S'|(D-2) \geq 2|S'| \geq |S|.$$

□

Now, we are ready to present the integrality gap example. Let G' be a bipartite graph as in Corollary 2, for some $N = 2n \in \mathbb{N}$ and some constant $D \geq 4$. Let E' be the set of edges of G'. Note that the set of vertices of G' is given by:

$$U = \{u_i\}_{i \in [2n]}, U' = \{u'_i\}_{i \in [2n]}, V' = \{v'{}_i\}_{i \in [2n]}.$$

Using G', we construct a bipartite graph G_b with bipartition $\tilde{B} = B_1 \cup B_2$ and C, as follows. We introduce the following sets of addional vertices:

$$U'' = \{u''_i\}_{i \in [2n-1]}, V = \{v_i\}_{i \in [2n]}, W = \{w_i\}_{i \in [2n]}, W' = \{w'_i\}_{i \in [2n]}.$$

We then set $B_1 = V$, $B_2 = V' \cup W \cup W'$ and $C = U \cup U' \cup U''$. The edge set $\tilde{E}$ of G_b includes all the edges E' of G', with, in addition, the solid edges depicted in Fig. 4. Finally, let the weight of $v \in W \cup W'$ be 1, and the weight of $v \in V \cup V'$ be ∞. We would like to point out that in our example the maximum degree of a vertex in G_b is constant, and that a maximum matching of G_b has size $|C|$.

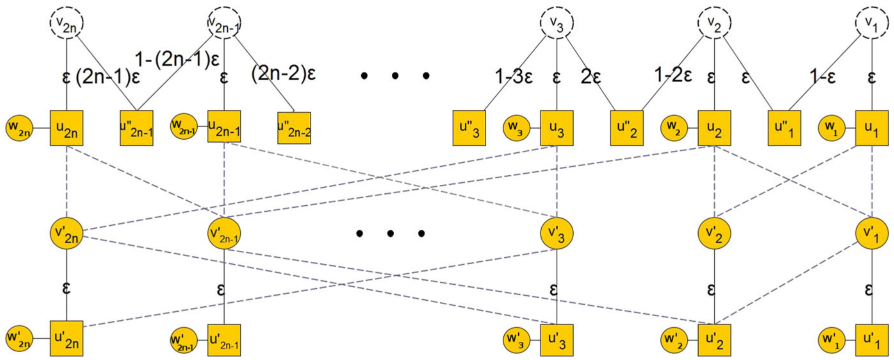

Fig. 4 Graph G_b with $\tilde{B}$ vertices shown by *circles*: B_1 vertices are *dotted circles*; B_2 vertices are *solid circles*, and C vertices are shown by *squares*. The edge set of graph includes all solid edges and the edge set of the graph G' used in the construction of G_b

Claim There is a fractional feasible solution for the minimization LP in (5) that has objective function value 2.

Proof Define solution (y, z) for $\epsilon = \frac{1}{2n}$ as follows:

- For $v \in V$, $y_v = 1$ and $z_v = 0$.
- For $v' \in V'$, $y_{v'} = \epsilon$ and $z_{v'} = 0$.
- For $w \in W \cup W'$, $y_w = 0$ and $z_w = \epsilon$.
- For $u \in U \cup U'$, $y_u = 1 - \epsilon$.
- For $u \in U''$, $y_{u''} = 0$.

Inequality (1) is satisfied as for each $v \in B_1$, as $y_v = 1$ and $z_v = 0$. Inequality (2) is satisfied as for any edge $\{u, v\}$ for $v \in B_2$ and $u \in C$, $y_u = 1 - \epsilon$ and $z_v + y_u = \epsilon$. In order to show Inequality (3) is satisfied, we show that there exists a feasible flow that sends $1 - y_u$ flow from each $u \in C$ and each $v \in \tilde{B}$ receives y_v flows. This flow is actually depicted in Fig. 4. Equality (4) is trivially satisfied as $\sum_{v \in B} y_v = 2n \times 1 + 2n \times \epsilon$ and $\sum_{u \in C} y_u = 4n(1 - \epsilon)$, which adds to $6n - 1$ for $\epsilon = \frac{1}{2n}$. Finally, since each $w \in W \cup W'$ has weight 1 and $z_w = \epsilon = \frac{1}{2n}$, the objective function value of this solution is 2. □

In order to exhibit the integrality gap, it is enough to show that the minimum B_1-essentializer in this graph has weight $\Omega(n)$. In fact, equivalently we will show that the minimum B_1-essentializer has weight at least αN where $\alpha > 0$ is a constant from Corollary 2 and $N = 2n$.

Claim The minimum weight B_1-essentializer for G_b has weight at least αN.

Proof Any B_1-essentializer S with finite cost does not remove any vertex in $V \cup V'$. Since $B_1 = V$, every vertex in V is essential in $G_b \backslash S$, so they have to be matched by a maximum matching M^* in $G_b \backslash S$. Note that $|U''| = 2n - 1$, so at least one of $v \in V$ must be matched to a vertex $u \in U$ (see Fig. 5). Since vertices in V' have ∞ cost as well, all the neighbors $N(u) \cap V'$ must be essential as well. Let U_1 denote the subset

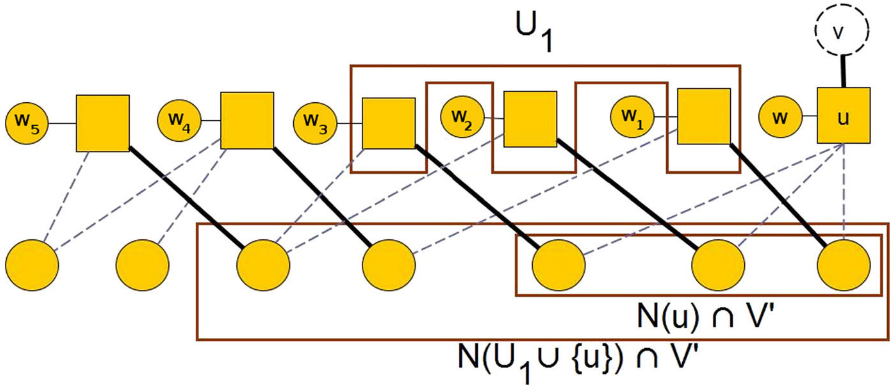

Fig. 5 Depiction of the process used in the proof of the claim

of vertices $U_1 \subseteq U \cup U'$ that are matched to vertices in $N(u) \cap V'$ by M^* (see Fig. 5 for an example). By the same argument, all vertices in $N(U_1) \cap V'$ must be matched by M^* (see Fig. 5). We can generalize this argument. More precisely, let V_e denote the set of essential nodes in V', and let U_e denote the set of vertices matched to V_e by M^* (note that U_e does not include u, as u is matched to v by M^*). All vertices in $N(U_e \cup \{u\}) \cap V'$ must be essential, so $N(U_e \cup \{u\}) \cap V' \subseteq V_e$. Since graph G is $(\alpha N, 1)$ vertex expander, if $|U_e \cup \{u\}| \leq \alpha N$, then $|N(U_e \cup \{u\}) \cap V'| \geq |U_e \cup \{u\}| > |U_e|$. On the other hand, V_e includes $N(U_e \cup \{u\}) \cap V'$ and is matched to U_e, which is not possible. So the size of U_e is at least αN. For each $u_i \in U_e$, the B_1-essentializer must include w_i or w_i' depending on whether $u_i \in U$ or $u_i \in U'$, respectively. Hence, the B_1-essentializer has size (i.e., weight) at least αN. □

5.3 Integrality gap of linear programming formulation $(\mathcal{P}_1)$

For any integer $\delta > 1$, consider a complete bipartite graph $G_b = (\tilde{B} \cup C, \tilde{E})$ where $\tilde{B} = B_1 \cup B_2$ for $B_1 = v_0$ and $B_2 = \{v_1, v_2, \ldots, v_{\delta^2}\}$, $C = \{u_1, u_2, \ldots, u_\delta\}$, and $\tilde{E} = \{(u, v) : u \in C, v \in \tilde{B}\}$. Assign weight δ^2 to v_0 and weight 1 to all the vertices in B_2. Set $B' := \{v_0\}$.

We show that the integrality gap of linear programming $(\mathcal{P}_1) := \{\min \sum_v w_v z_v : (x, y) \in P_f, y_v = 1 \ \forall v \in B', y_v = 0 \ \forall v \in \tilde{B} \backslash B'\}$ is $\Omega(\delta)$. Note that δ is the maximum degree of vertices in B_2.

Claim There is a feasible fractional solution to $(\mathcal{P}_1)$ with objective function value δ.

Proof Assign $z_{v_0} = 0$, $z_{v_i} = 1/\delta$ for all $i = 1, \ldots, \delta^2$, and $y_{u_i} = (\delta - 1)/\delta$ for all $i = 1, \ldots, \delta$. Inequality (1) is satisfied as $y_{v_0} + z_{v_0} = 1$. Inequality (2) is satisfied as for any edge $\{u, v\}$ with $v \in B_2$ and $u \in C$, $y_v + z_v + y_u = 0 + \frac{1}{\delta} + \frac{\delta-1}{\delta} \geq 1$. In order to show that Inequality (3) is satisfied, we show that there exists a maximum s-t flow of value $|C| - y(C) = 1$ in the network $\mathcal{N} = (V_{\mathcal{N}}, A_{\mathcal{N}})$ constructed in Lemma 8: send flow on $\frac{1}{\delta}$ for each $(s, u) \in A_{\mathcal{N}}$, forward the flow of $\frac{1}{\delta}$ from each $u \in C$ to v_0, and finally send flow of 1 from v_0 to t. Therefore Inequality (3) is satisfied as otherwise the maximum flow should be strictly less than 1 in network $\mathcal{N}$. Equality (4) is trivially satisfied. Finally, since each $v \in B_2$ has weight 1 and $z_v = \frac{1}{\delta}$, the objective function value of this solution is δ. $\square$

In order to exhibit the integrality gap, it remains to show that an integral solution has weight $\Omega(\delta^2)$.

Claim Any integral solution to $(\mathcal{P}_1)$ has weight $\Omega(\delta^2)$.

Proof Since $y(V) = \delta$ and $y_{v_0} = 1$, there exists at least one vertex $u \in C$ with $y_u = 0$. Since every vertex $v \in B_2$ has y-value zero, we must have $z_v = 1$ for all $v \in B_2$ in order to satisfy Inequality (2). The objective function value is then $\Omega(\delta^2)$. $\square$

References

1. Agarwal, A., Charikar, M., Makarychev, K., Makarychev, Y.: O($\sqrt{\log n}$) approximation algorithms for min UnCut, min 2CNF deletion, and directed cut problems. Proc. STOC **2005**, 573–581 (2005)
2. Bassalygo, L.A.: Asymptotically optimal switching circuits. Probl. Peredachi Inf. **17**(3), 81–88 (1981)
3. Berge, C.: Two theorems in graph theory. Proc. Natl. Acad. Sci. USA **43**(9), 842 (1957)
4. Biró, P., Bomhoff, M., Golovach, P.A., Kern, W., Paulusma, D.: Solutions for the stable roommates problem with payments. Theor. Comput. Sci. **540**, 53–61 (2014)
5. Bock, A., Chandrasekaran, K., Könemann, J., Peis, B., Sanità, L.: Finding small stabilizers for unstable graphs. Math. Program. **154**(1), 173–196 (2015)
6. Chalkiadakis, G., Elkind, E., Wooldridge, M.: Computational aspects of cooperative game theory. In: Synthesis Lectures on Artificial Intelligence and Machine Learning, vol 5. Morgan & Claypool, pp. 1–168 (2011)
7. Cook, W., Cunningham, W., Pulleyblank, W., Schrijver, A.: Combinatorial Optimization. Wiley, New York (1998)
8. Edmonds, J.: Paths, trees, and flowers. Can. J. Math. **17**, 449–467 (1965)
9. Garg, N., Vazirani, V.V., Yannakakis, M.: Approximate max-flow min-(multi)cut theorems and their applications. SIAM J. Comput. **25**, 698–707 (1993)
10. Ito, T., Kakimura, N., Kamiyama, N., Kobayashi, Y., Okamoto, Y.: Efficient stabilization of cooperative matching games. Theor. Comput. Sci. (2017). doi:10.1016/j.tcs.2017.03.020
11. Kleinberg, J., Tardos, É.: Balanced outcomes in social exchange networks. Proc. STOC **2008**, 295–304 (2008)
12. Kohli, R., Krishnamurti, R., Mirchandani, P.: The minimum satisfiability problem. SIAM J. Discrete Math. **7**(2), 275–283 (1994)
13. Könemann, J., Larson, K., Steiner, D.: Network bargaining: using approximate blocking sets to stabilize unstable instances. Theor. Comput. Sci. **57**(3), 655–672 (2015)
14. Korach, E., Nguyen, T., Peis, B.: Subgraph characterization of Red/Blue-Split Graph and König Egerváry graphs. In: Proceedings of SODA, pp. 842–850 (2006)
15. Mishra, S., Raman, V., Saurabh, S., Sikdar, S., Subramanian, C.: The complexity of König subgraph problems and above-guarantee vertex cover. Algorithmica **61**(4), 857–881 (2011)
16. Nash, J.: The bargaining problem. Econometrica **18**, 155–162 (1950)
17. Nisan, N., Roughgarden, T., Tardos, É., Vazirani, V.V.: Algorithmic Game Theory. Cambridge University Press, New York (2007)
18. Schrijver, A.: Combinatorial Optimization. Springer, New York (2003)
19. Shapley, L.S., Shubik, M.: The assignment game: the core. Int. J. Game Theory **1**(1), 111–130 (1971)
20. Sterboul, F.: A characterization of the graphs in which the transversal number equals the matching number. J. Comb. Theory Ser. B **27**, 228–229 (1979)
21. Vadhan, S.P.: Pseudorandomness. Now Publishers Inc., Hanover (2012)