How to Secure Matchings Against Edge Failures

Felix Hommelsheim
Department of Mathematics, TU Dortmund University, Germany
felix.hommelsheim@math.tu-dortmund.de

Moritz Mühlenthaler
Department of Mathematics, TU Dortmund University, Germany
moritz.muehlenthaler@math.tu-dortmund.de

Oliver Schaudt
Department of Mathematics, RWTH Aachen University, Germany
schaudt@mathc.rwth-aachen.de

Abstract

Suppose we are given a bipartite graph that admits a perfect matching and an adversary may delete any edge from the graph with the intention of destroying all perfect matchings. We consider the task of adding a minimum cost edge-set to the graph, such that the adversary never wins. We show that this problem is equivalent to covering a digraph with non-trivial strongly connected components at minimal cost. We provide efficient exact and approximation algorithms for this task. In particular, for the unit-cost problem, we give a $\log_2 n$-factor approximation algorithm and a polynomial-time algorithm for chordal-bipartite graphs. Furthermore, we give a fixed parameter algorithm for the problem parameterized by the treewidth of the input graph. For general non-negative weights we give tight upper and lower approximation bounds relative to the Directed Steiner Forest problem. Additionally we prove a dichotomy theorem characterizing minor-closed graph classes which allow for a polynomial-time algorithm. To obtain our results, we exploit a close relation to the classical Strong Connectivity Augmentation problem as well as directed Steiner problems.

2012 ACM Subject Classification Hardware → Robustness; Mathematics of computing → Matchings and factors; Mathematics of computing → Graph algorithms; Mathematics of computing → Approximation algorithms; Theory of computation → Fixed parameter tractability; Mathematics of computing → Mathematical optimization

Keywords and phrases Matchings, Robustness, Connectivity Augmentation, Graph Algorithms, Treewidth

Digital Object Identifier 10.4230/LIPIcs.STACS.2019.38

Related Version A full version of the paper is available at `https://arxiv.org/abs/1805.01299`.

Funding *Felix Hommelsheim*: Research partially supported by the German Research Foundation (DFG), RTG 1855.
Moritz Mühlenthaler: Research partially supported by the German Research Foundation (DFG), RTG 1855.

Acknowledgements We would like to thank Viktor Bindewald for the fruitful discussions about some of the results in this paper.

1 Introduction

We say that a bipartite graph is *robust* if it admits a perfect matching after the removal of any single edge. Our goal is to make a bipartite graph robust at minimal cost by adding edges from its bipartite complement and we study the complexity of the corresponding optimization problem. We refer to this problem informally as *robust matching augmentation*. In general, an *augmentation problem* asks for a minimum-cost set of edges to be added to a graph in order to establish a certain property. In our context this property is robustness. As a motivation, consider some assignment-type application, such as staff or task scheduling. The application requires that we choose a perfect matching that assigns, say, tasks to machines. By buying additional edges, we would like to ensure that after the failure of any single edge the resulting graph has a perfect matching, i.e., we may continue our operation. Buying edges may correspond for example to training staff or upgrading machines. Note that in many situations some kind of infrastructure is already available, so it may make sense to upgrade it instead of designing robust infrastructure from scratch.

A *design problem* asks for a minimum-cost subgraph with a certain property, for instance a minimum-cost k-edge-connected subgraph [12, 22]. Robust matching augmentation can be stated also as a design problem, where the given infrastructure is available at zero cost and the host graph is a complete bipartite graph. In fact, our problem is a special case of the bulk-robust assignment problem, a design problem introduced in [2]. Bulk-robustness is a redundancy-based robustness concept proposed by Adjiashvili, Stiller, and Zenklusen [3]. Roughly speaking, the input of a bulk-robust design problem is a host graph and a list of sets of edges, the *failure scenarios*. If a failure scenario emerges then the corresponding edges are deleted from the host graph. The task is to select a minimum-cost subgraph of the host graph that has a certain property (e.g., it contains an assignment [2], a spanning tree [3], or an st-path [4]), no matter which failure scenario emerges. Bulk-robust design problems are notoriously hard. For example, the bulk-robust assignment problem is known to be NP-hard even if only one of two fixed edges may fail [2]. Here, we consider the setting that any single edge of the host graph may fail.

We provide a detailed study of the complexity of the robust matching augmentation problem. For the unweighted problem we give a tight $\log n$-factor approximation algorithm as well as polynomial-time algorithms for chordal-bipartite graphs and graphs of bounded treewidth. For the weighted problem we give a characterization of minor-closed graph classes for which the problem admits a polynomial-time algorithm. Our algorithmic results are based on the following reformulation of the robust matching augmentation problem: Given a digraph, find a minimum-cost superset of its arcs, such that each vertex is contained in some non-trivial strongly connected component. In contrast, the classical *strong connectivity augmentation* problem asks for the minimal number of arcs that are needed to have all vertices covered by a *single* strongly connected component. It was shown by Eswaran and Tarjan that this problem admits a polynomial-time algorithm, but its edge-weighted variant is NP-hard [16]. It turns out that the classical algorithm for strong connectivity augmentation is useful in order to satisfy our more relaxed strong connectivity requirements at minimal cost.

Our Contribution

Recall that we call a bipartite graph *robust* if it admits a perfect matching after the removal of any single edge. For a bipartite graph (V, E), we denote by $\overline{E}$ the edge-set of its bipartite complement. We provide algorithms and hardness results for variants of the following problem.

Robust Matching Augmentation
instance: Undirected bipartite graph $G = (U + W, E)$ that admits a perfect matching.
task: Find a set $L \subseteq \overline{E}$ of minimum cardinality, such that the graph $G + L$ is robust.

Based on a characterization of robustness in terms of strong connectivity, we provide a deterministic $\log_2 n$-factor approximation for Robust Matching Augmentation, as well as a fixed parameter tractable (FPT) algorithm for the same problem parameterized by the treewidth of the input graph. We also give a polynomial-time algorithm for instances on chordal-bipartite graphs, which are bipartite graphs without induced cycles of length at least six. Furthermore, we show that Robust Matching Augmentation admits no polynomial-time sublogarithmic-factor approximation algorithm unless $\mathsf{P} = \mathsf{NP}$, so our approximation guarantee is essentially tight. We also consider the following more general setting. Let us call a bipartite graph k-robust, if it admits a matching of cardinality k after the removal of any single edge. By a simple reduction we show that our algorithmic results carry over to the task of making a bipartite graph k-robust.

We refer by Weighted Robust Matching Augmentation to the generalization of Robust Matching Augmentation, where each edge $e \in \overline{E}$ has a non-negative cost c_e. The task is to find a minimum-cost set $L \subseteq \overline{E}$, such that $G + L$ is robust. First, we show that the approximability of Weighted Robust Matching Augmentation is closely linked to that of Directed Steiner Forest. In particular we show that an $f(n)$-factor approximation algorithm for Weighted Robust Matching Augmentation implies an $f(n + k)$-factor approximation algorithm for Directed Steiner Forest, where k is the number of terminal pairs. By a result of Halperin and Krauthgamer [23] it follows that there is no $\log^{2-\varepsilon}(n)$-factor approximation for Weighted Robust Matching Augmentation, unless $\mathsf{NP} \subseteq \mathrm{ZTIME}(n^{\mathsf{polylog}(n)})$. On the positive side, we show that an $f(k)$-factor approximation for the Directed Steiner Forest problem yields an $(f(k) + 1)$-factor approximation for Weighted Robust Matching Augmentation. Hence, the algorithms from [10, 18] give an approximation guarantee of $1+n^{\frac{1}{2}+\varepsilon}$ for Weighted Robust Matching Augmentation, for every $\varepsilon > 0$.

Second, we prove a complexity dichotomy based on graph minors. Let $\mathcal{T}$ be a class of connected graphs closed under connected minors. We show that Weighted Robust Matching Augmentation restricted to input graphs from $\mathcal{T}$ is NP-complete if $\mathcal{T}$ contains at least one of two simple graph classes, which will be defined in Section 5, and admits a polynomial-time algorithm otherwise. The polynomial-time algorithm for the remaining instance classes uses a reduction to the Directed Steiner Forest problem with a constant number of terminal pairs, which in turn admits a (slice-wise) polynomial-time algorithm due to a result by Feldman and Ruhl [17]. The terminal pairs of the instance are computed by the Eswaran-Tarjan algorithm.

Summary of Algorithmic Techniques

By close inspection, it turns out that in order to make some bipartite graph G robust at minimum cost, we may restrict our attention to failures of single edges from any fixed perfect matching M of G. We then show that the resulting problem is equivalent to augmenting a minimum-cost set A of arcs to a given digraph D, such that in the graph $D + A$, each vertex is contained in a strongly connected component and each strongly connected component contains at least two vertices. In order to satisfy these connectivity requirements, we select certain sources and sinks of the condensation of the digraph and add a minimum-cardinality set of arcs, such that the selected sources and sinks are contained in a single strongly connected component. For this purpose, we use the classical Eswaran-Tarjan algorithm. From the arcs we added we obtain an optimal set L of edges such that $G + L$ is robust, provided that the selection of sources and sinks was optimal.

We model the task of selecting sources and sinks as a variant of the Set Cover problem with some additional structure. Given an acyclic digraph, the task is to select a minimum-cardinality subset of the sources, such that each sink is reachable from at least one of the selected sources. We refer to this problem as Source Cover and remark that its complexity may be of independent interest, since it generalizes Set Cover but is a special case of

Directed Steiner Tree. We give an FPT algorithm for the Source Cover problem parameterized by the treewidth of the input graph and a polynomial-time algorithm for chordal-bipartite graphs (ignoring orientations). The FPT algorithm is single exponential in the treewidth. Our reductions from Robust Matching Augmentation to Source Cover preserve chordal-bipartiteness and bounded treewidth, so efficient algorithms for Source Cover on these graph classes imply efficient algorithms for Robust Matching Augmentation on the same classes.

As a by-product of our analysis of the Source Cover problem, we obtain FPT algorithms for the node-weighted and arc-weighted versions of the Directed Steiner Tree problem on acyclic digraphs, which are exponential in the treewidth and linear in the number of nodes of the input graph.

Related work

In [2], Adjiashvili, Bindewald and Michaels proposed an LP-based randomized algorithm for the bulk-robust assignment problem. They claim an $O(\log n)$-factor approximation guarantee for their algorithm. Since the robust assignment problem generalizes Weighted Robust Matching Augmentation, an $O(\log n)$-factor approximation for our problem is implied. However, due to our inapproximability result for Weighted Robust Matching Augmentation, this can not be true, unless $\mathsf{NP} \subseteq \mathrm{ZTIME}(n^{\mathsf{polylog}(n)})$. The authors of [2] agree that their analysis is incorrect.

A connectivity augmentation problem related to strong connectivity, but of a different flavor, is the *tree augmentation problem* (TAP). The TAP asks for a minimum-cost edge-set that increases the edge-connectivity of a given tree from one to two. In contrast to robust matching augmentation, the TAP admits a constant-factor approximation [21]. The constant has recently been lowered to $3/2 + \varepsilon$ for bounded-weight instances [1, 19]. Similar to robust matching augmentation, the input graph is available at zero cost. Let us briefly remark that there is more conceptual similarity. The *matching preclusion number* of a graph is the minimal number of edges to be removed, such that the remaining graph has no perfect matching. Robust matching augmentation can be stated as the task of finding a minimum-cost edge-set that increases the matching preclusion number of a bipartite graph from one to two, while the TAP aims to increase connectivity from one to two. The matching preclusion number is considered to be a measure of robustness of interconnect networks [9, 11]. Determining the matching preclusion number of a graph is NP-hard [14, 25].

In our reduction from robust matching augmentation problem to a connectivity augmentation problem, we construct a digraph from the input graph and a fixed perfect matching. This construction is closely related to the classical Dulmage-Mendelsohn decomposition (DM-decomposition) introduced in [15]. In fact the digraph from our reduction can be obtained from the auxiliary digraph that is used for computing the DM-decomposition of a graph by contracting the edges of the perfect matching. In [5], the authors consider the problem of making a bipartite graph DM-irreducible, which means that its DM-decomposition consists of a single component. They show that the unweighted variant of this problem admits a polynomial-time algorithm. For balanced bipartite graphs that admit a perfect matching, the problem reduces to the strong connectivity augmentation problem. Hence, DM-irreducibility of such graphs implies robustness, but not vice versa.

Robust perfect matchings with a given recovery budget were studied by Dourado et al. in [14]. Our notion of robustness corresponds to *1-robust ∞-recoverable* in their terminology. They provide hardness results and structural insights mainly for fixed recovery budgets, which bound the number of edges that can be changed in order to repair a matching, after a certain number of edges has been removed from the graph.

Notation

Undirected and directed graphs considered here are simple. For sets U, W, we denote by $U+W$ their disjoint union. For an undirected bipartite graph $G = (U+W, E)$ with bipartition (U, W), we denote by $\overline{E}$ the edge-set of its bipartite complement. Let $D = (V, A)$ be a directed graph. We refer by $\overline{A}$ to the arcs not present in D. That is, we let $\overline{A} \subseteq (V \times V) \setminus A$. By $\mathcal{U}(D)$ we refer to the underlying undirected graph of D. For $L \subseteq \overline{E}$, we write $G + L$ for the graph $G' = (V(G), E(G) \cup L)$. Simple paths in graphs are given by a sequence of vertices. For graphs G, H we write $H \subseteq G$ if H is a subgraph of G. Recall that a graph H is an *induced minor* of a graph G if it arises from G by a sequence of vertex deletions and edge contractions. Similarly, the graph H is a *minor* of G if we additionally allow edge deletion. Furthermore, the graph H is a *connected minor* of G if H is connected and a minor of G. In general, contractions may result in parallel edges or loops, which we simply discard in order to keep our graphs simple. Let $\mathcal{G}$ be a class of graphs. We will refer to the restriction of (Weighted) Robust Matching Augmentation to instances where the graph G is bipartite, admits a perfect matching, and belongs to the class $\mathcal{G}$ as *(*Weighted*)* Robust Matching Augmentation *on* $\mathcal{G}$. Given a set of items X and sets $\mathcal{S} \subseteq 2^X$, the Set Cover problems asks for a minimum-cardinality subset $C \subseteq \mathcal{S}$, such that each $x \in X$ is contained in some $s \in C$. The *incidence graph* $G(\mathrm{I})$ of a Set Cover instance $\mathrm{I} = (X, \mathcal{S})$ is an undirected bipartite graph on the vertex set $X + \mathcal{S}$ that has an edge xs if and only if the item $x \in X$ is contained in the set $s \in \mathcal{S}$.

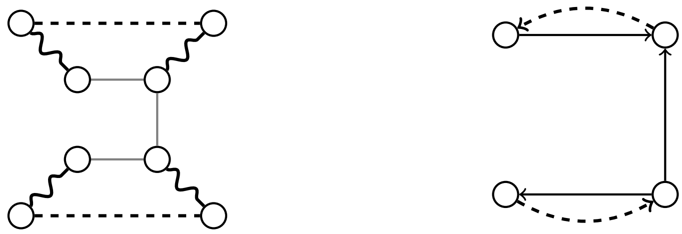
Organization of the Paper

The remainder of the paper is organized as follows. We illustrate the relation between robust matching augmentation and strong connectivity augmentation in Section 2. In Section 3 we show an even closer relation of Robust Matching Augmentation to the Source Cover problem. Algorithms for the Source Cover problem are given in Section 4 as well as our results on robust matching augmentation with unit costs. In Section 5 we give the complexity classification for the weighted version of the problem and Section 6 concludes the paper.

2 Robust Matchings and Strong Connectivity Augmentation

In this section we give some preliminary observations on the close relationship between robust matching augmentation with unit costs and strong connectivity augmentation. For this purpose, we fix an arbitrary perfect matching and construct an auxiliary digraph that is somewhat similar to the *alternating tree* used in Edmond's blossom algorithm. We show that the original graph is robust if the auxiliary graph is strongly connected (but not vice versa). Furthermore, we show that there is an optimal edge-set making the given graph robust, that corresponds to a set of arcs connecting sources and sinks in the auxiliary digraph. Finally, if no source or sink of the auxiliary digraph corresponds to a non-trivial robust part of the original graph, then we may use the algorithm for strong connectivity augmentation by Eswaran and Tarjan [16] to make the original graph robust. As a consequence, we have that Robust Matching Augmentation on trees can be solved efficiently by using the Eswaran-Tarjan algorithm. In Section 3, we will generalize this result.

Let $G = (U + W, E)$ be a bipartite graph that admits a perfect matching and let M be an arbitrary but fixed perfect matching M of G. We call an edge $e \in M$ *critical* if $G - e$ admits no perfect matching. Observe that an edge $e \in M$ is critical if and only if it is not



(a) Graph G and matching M (wiggly edges). **(b)** Digraph $D(G, M)$.

Figure 1 Illustration of the correspondence between the dotted edges of G and dotted arcs of $D(G, M)$.

contained in an M-alternating cycle. Furthermore, no edge in $E \setminus M$ is critical. Since M is perfect, each edge $e \in M$ is incident to a unique vertex u_e of U. We consider the following auxiliary digraph $D(G, M) = (U, A)$, whose arc-set A is given by

$$A := \{uu' \mid u, u' \in U : \text{there is a vertex } w \in W \text{ such that } uw \in M \text{ and } wu' \in E \setminus M\}.$$

We first note that the choice of the bipartition of G is irrelevant.

▶ **Fact 1.** *Let $G' = (U' + W', E)$, where (U', W') is a bipartition of G. Then $D(G, M)$ is isomorphic to $D(G', M)$.*

Note that we may perform the reverse construction as well. That is, from any digraph D' we may obtain a corresponding undirected graph G and a perfect matching M of G such that $D(G, M) = D'$. In fact, augmenting edges to G is equivalent to augmenting arcs to $D(G, M)$.

▶ **Fact 2.** *Let $\overline{A}$ be the set of arcs that are not present in $D(G, M)$. Then there is a 1-to-1 correspondence between $\overline{E}$ and $\overline{A}$.*

An example of the correspondence mentioned in Fact 2 is shown in Figure 1. In order to keep our notation tidy, we will make implicit use of Fact 2 and refer to $\overline{A}$ and $\overline{E}$ interchangeably. Observe that for edges $e, f \in M$ there is an M-alternating path containing e and f in G if and only if u_e is connected to u_f in $D(G, M)$. This implies the following characterization of robustness.

▶ **Fact 3.** *G is robust if and only if each strongly connected component of $D(G, M)$ is non-trivial, that is, it contains at least two vertices.*

Let D' be a digraph. A vertex of D' is called a *source* (*sink*) if it has no incoming (outgoing) arc. We refer to the set of sources (sinks) of D' by $V^+(D')$ ($V^-(D')$). Furthermore, we denote by $C(D')$ the *condensation* of D', that is, the directed acyclic graph of strongly connected components of D'. We call a source or sink of $C(D')$ *strong* if the corresponding strongly connected component of D' is non-trivial. From Fact 3 it follows that a subgraph of G that corresponds to a strong source or a strong sink is robust against the failure of a single edge. Furthermore, observe that the choice of the perfect matching M of G is irrelevant in the following sense.

▶ **Fact 4.** *Let M and M' be perfect matchings of G. Then $C(D(G, M))$ is isomorphic to $C(D(G, M'))$.*

Fact 4 is of key importance for our algorithmic results, for which we generally assume that some fixed perfect matching is given. Next, we observe that for unit costs we may restrict our attention to connecting sources and sinks of $C(D)$ in order to make G robust. It is easy to check that this does not hold for general non-negative costs.

▶ **Fact 5.** *Let $L \subseteq \overline{E}$ such that $G + L$ is robust. Then there is some $L' \subseteq \overline{E}$ of cardinality at most $|L|$, such that $G + L'$ is robust and L' connects only sinks to sources of $C(D(G, M))$.*

We remark that the construction of L' given in the proof of Fact 5 can be performed in polynomial time.

We denote by $\gamma(D')$ the minimal number of arcs to be added to a digraph D' in order to make it strongly connected. Eswaran an Tarjan have proved the following min-max relation [16].

▶ **Fact 6.** *Let D' be a digraph. Then* $\gamma(D') = \max\{|V^+(D')|, |V^-(D')|\}$.

From the proof of Fact 6 it is easy to obtain a polynomial-time algorithm that, given a digraph D', computes an arc-set L of cardinality $\gamma(D')$ such that $D' + L$ is strongly connected [20]. We will refer to this algorithm by Eswaran-Tarjan. The following proposition illustrates the usefulness of the algorithm Eswaran-Tarjan for Robust Matching Augmentation, and at the same time its limitations.

▶ **Fact 7.** *Suppose that $C(D(G, M))$ contains no strong sources or sinks. Then* Eswaran-Tarjan *computes a set $L \subseteq \overline{E}$ of minimum cardinality such that $G + L$ is robust.*

Fact 7 implies that Eswaran-Tarjan solves Robust Matching Augmentation on trees. If strong sources or sinks are present in $D(G, M)$, then we may or may not need to consider them in order to make G robust. This is precisely what makes the problem Robust Matching Augmentation hard. This close connection will be presented in Section 3. We will formalize the task of selecting strong sources and sinks in terms of the Source Cover problem, which is discussed in Section 4.

3 Robust Matching Augmentation

In this section we present our main technical tool for solving the problem Robust Matching Augmentation. By combining it with the results in Section 4 we obtain our algorithmic results. Let us first restate the problem in a slightly different way.

Robust Matching Augmentation
instance: Bipartite graph $G = (U + W, E)$ and perfect matching M of G.
task: Find a minimum-cardinality set $L \subseteq \overline{E}$ such that $G + L$ is robust.

Fixing the perfect matching M in the instance is just for notational convenience, since we can compute a perfect matching in polynomial time and our results do not depend on the exact choice of M, according to the discussion in Section 2. For the main theorem of this section we need to introduce the Source Cover problem. Given an acyclic digraph, the Source Cover problem asks for a minimum-cardinality subset of its sources, such that each sink is reachable from at least one selected source. The Source Cover problem is formally defined as follows.

Source Cover
instance: Weakly connected acyclic digraph $D = (V, A)$.
task: Find a minimum-cardinality subset S of the sources $V^+(D)$ of D, such that for each sink $t \in V^-(D)$ there is an S-t-path in D.

Note that the assumption that D is connected is needed only for technical reasons. Our main technical result is the following.

▶ **Theorem 8.** *There is a polynomial-time algorithm that, given an instance* $\mathrm{I} = (G, M)$ *of* Robust Matching Augmentation*, computes two instances* $\mathcal{A}_1 = (S_1)$ *and* $\mathcal{A}_2 = (S_2)$ *of* Source Cover *such that the following holds.*

1. $\mathcal{U}(S_1)$ *and* $\mathcal{U}(S_2)$ *are induced minors of* $\mathcal{U}(D(G, M))$.
2. $\mathrm{OPT}(\mathrm{I}) = \max\{\mathrm{OPT}(\mathcal{A}_1), \mathrm{OPT}(\mathcal{A}_2)\}$
3. *From a solution* C_1 *of* $\mathcal{A}_1$ *and a solution* C_2 *of* $\mathcal{A}_2$ *we can construct in polynomial time a solution* L *of* I *of cardinality* $\max\{|C_1|, |C_2|\}$.

Proof. Let $\mathrm{I} = (G, M)$ be an instance of Robust Matching Augmentation, where $G = (U + W, E)$. Our goal is to obtain from solutions of the Source Cover instances a suitable selection of sources and sinks of $C(D(G, M))$, such that we can make M robust by connecting the selected sources and sinks, using the algorithm Eswaran-Tarjan. Let us denote by u_e the vertex in U that is incident to an edge $e \in M$. Furthermore, let $D := D(G, M)$. We construct the Source Cover instance $\mathcal{A}_1$ as follows. For each critical edge $e \in M$, we remove from D each vertex $v \in U - u_e$, such that v is reachable from u_e in D. Let D' be the resulting graph and let the Source Cover instance $\mathcal{A}_1$ be given by $\mathcal{A}_1 := (C(D'))$. The construction of $\mathcal{A}_2$ is as for $\mathcal{A}_1$, but with the arcs of D reversed. This turns the sources of D into sinks. Clearly, the acyclic digraphs of $\mathcal{A}_1$ and $\mathcal{A}_2$ are induced minors of $\mathcal{U}(D)$, since they were constructed by deleting vertices of $\mathcal{U}(D)$ and contracting strong components. By Fact 3, the set of critical edges can be obtained efficiently by Tarjan's classical algorithm for computing strongly connected components. In order to generate $\mathcal{A}_1$ and $\mathcal{A}_2$, observe that D' and $C(D')$ can both be obtained by applying a breadth-first search starting at each vertex of D or D', respectively. So it remains to prove Statement 2 and 3.

Let C_1 (C_2) be a solution to $\mathcal{A}_1$ ($\mathcal{A}_2$). We show how to construct in polynomial time a solution L of I of cardinality $\max\{|C_1|, |C_2|\}$. Let $X \subseteq V(\widehat{D})$ be the set of vertices incident to critical edges. Moreover, let $\widehat{D} \subseteq C(D)$ be the graph induced by the vertices of $C(D)$ that are on C_1X-paths or on XC_2-paths in $C(D)$. Note that $\widehat{D}$ can be computed by a depth-first search applied on each source and sink. By running Eswaran-Tarjan on $\widehat{D}$ we obtain an arc-set L^* such that $\widehat{D} + L^*$ is strongly connected. Hence, each $u \in X$ is on some directed cycle in $\widehat{D} + L^*$. From L^* we can obtain in a straight-forward way an arc-set L of the same cardinality, such that each $u \in X$ is on some directed cycle of $D + L$. For each $ss' \in L^*$, we add to L an arc uu', where u (u') is some vertex in the strong component s (s') of D. By the construction of L, each $u \in X$ is on some directed cycle of D. By Fact 2 and 6 we have constructed a solution L of I of cardinality $|L| = |L^*| = \max\{|C_1|, |C_2|\}$. This completes the proof of Statement 3.

It remains to prove that $\mathrm{OPT}(\mathrm{I}) \geq \max\{\mathrm{OPT}(\mathcal{A}_1), \mathrm{OPT}(\mathcal{A}_2)\}$. Suppose for a contradiction that $\mathrm{OPT}(\mathrm{I}) < \max\{\mathrm{OPT}(\mathcal{A}_1), \mathrm{OPT}(\mathcal{A}_2)\}$. Without loss of generality, let $\mathrm{OPT}(\mathcal{A}_1)$ attain the maximum. Due to Fact 5, we may assume that an optimal solution L of I connects sources and sinks of $C(D)$. Let $R \subseteq V(C(D))$ be the corresponding sources of $C(D)$. Then for each critical edge $e \in M$, the vertex u_e must be reachable from some source $s \in R$. But then R is a solution of $\mathcal{A}_1$ of cardinality $|R| = \mathrm{OPT}(\mathrm{I}) < \mathrm{OPT}(\mathcal{A}_1)$, a contradiction. ◀

By Theorem 8, in order to solve Robust Matching Augmentation, is suffices to solve two instances of Source Cover. Due to Statement 1 of the theorem, structural features of the input graph, such as bounded treewidth and chordal-bipartiteness, are passed on to the digraphs of the source cover instances. We now consider the following more general setting.

We call a bipartite graph *k-robust* if it admits a matching of cardinality k after the removal of any single edge.

> ROBUST k-MATCHING AUGMENTATION
> **instance:** Bipartite graph $G = (U + W, E)$ that admits a matching of size k.
> **task:** Find a minimum-cardinality set $L \subseteq \overline{E}$ such the graph $G + L$ is k-robust.

Note that if k is less than the size of a maximum matching then $L = \emptyset$ is optimal due to the existence of a larger matching. We give a polynomial-time reduction from ROBUST k-MATCHING AUGMENTATION to ROBUST MATCHING AUGMENTATION. Let (G, M) be an instance of ROBUST k-MATCHING AUGMENTATION, where the input graph G is given by $G = (V, E)$. Without loss of generality, we assume that M is U-perfect, so $|U| \leq |W|$. Otherwise, adding an edge joining two unmatched vertices solves the problem. We construct an instance (G', M') of ROBUST MATCHING AUGMENTATION as follows. Let G' be a copy of G to which we add a leaf to each unmatched vertex of W. We then add a vertex z to U joined to each vertex of the other part of the bipartition. Finally, we add a vertex z' joined to z and each leaf added in the previous step. Furthermore, we extend the matching M of G to a perfect matching M' of G' by adding the edges between the leaves and the previously unmatched vertices to M'. Note that by construction, if e is a critical edge of G' then $G - e$ does not admit a matching of cardinality $|M|$.

Note that the construction increases the treewidth by at most two, but does not preserve chordal-bipartiteness of the input graph. However, the corresponding digraph contains no induced cycle of length at least six, so all our algorithmic results for ROBUST MATCHING AUGMENTATION carry over to ROBUST k-MATCHING AUGMENTATION.

▶ **Proposition 9.** *There is a polynomial-time reduction f from* ROBUST k-MATCHING AUGMENTATION *to* ROBUST MATCHING AUGMENTATION*, such that the following holds. Let $\mathrm{I} := (G)$ be an instance of* ROBUST k-MATCHING AUGMENTATION *and let $f(\mathrm{I}) = (G')$. Then*

1. $\mathrm{OPT}(f(\mathrm{I})) = \mathrm{OPT}(\mathrm{I})$ *and from a solution L' of $f(\mathrm{I})$ we can construct in polynomial time a solution L of I such that $|L| \leq |L'|$.*
2. $tw(G') \leq tw(G) + 2$
3. *If G is chordal-bipartite then $\mathcal{U}(D(G', M'))$ has no induced cycle of length at least six.*

4 The Source Cover Problem

In Section 3 we made precise the close relation between ROBUST MATCHING AUGMENTATION and the SOURCE COVER problem. In this section we present our algorithmic results for the SOURCE COVER problem as well as their consequences consequences for ROBUST k-MATCHING AUGMENTATION. Recall that the SOURCE COVER problem asks for a minimum-cardinality subset of the source of a given digraph, such that each sink is reachable from at least one selected source. It is easy to see that SOURCE COVER is a special case of the DIRECTED STEINER TREE problem and that it generalizes SET COVER. We give a simple polynomial-time algorithm for SOURCE COVER if the input graph is chordal-bipartite (ignoring orientations). Furthermore, we show that SOURCE COVER parameterized by treewidth (again ignoring orientations) is FPT. As a by-product, we obtain a simple FPT algorithm for the arc-weighted and node-weighted versions of the DIRECTED STEINER TREE problem on acyclic digraphs, whose running time is linear in the size of the input graph and exponential in the treewidth of the underlying undirected graph. To the best of our knowledge, the parameterized complexity of the general DIRECTED STEINER TREE problem with respect to treewidth is open. For the corresponding *undirected* STEINER TREE problem, an FPT algorithm was given by Bodlaender et al. in [8].

(a) A digraph D such that $\mathcal{U}(D)$ is balanced, but $\mathcal{U}(F(D))$ is not.

(b) Digraphs D such that $\mathcal{U}(D)$ has treewidth one, but the treewidth of $\mathcal{U}(F(D))$ is unbounded.

Figure 2 Examples showing that flattening does not preserve balancedness or bounded treewidth.

By "flattening" the input digraph, we can transform an instance $\mathrm{I} = (D)$ of Source Cover into a Set Cover instance as follows. Let $F(D) = (V^+(D) \cup V^-(D), A')$ be an acyclic digraph, where A' is given by

$$A' := \{st \mid s \in V^+(D),\, t \in V^-(D),\, t \text{ is reachable from } s \text{ in } D\}.$$

Then $\mathcal{U}(F(D))$ is the incidence graph of a Set Cover instance $\mathcal{A}$ on $V^-(F(D))$, such that the feasible solutions of I and $\mathcal{A}$ are in 1-to-1 correspondence.

As a first consequence of Theorem 8, Proposition 9, and this "flattening" we may use the classic Greedy-Algorithm for Set Cover obtain a simple $\log_2 n$-factor approximation algorithm for Robust k-Matching Augmentation.

► **Corollary 10.** Robust k-Matching Augmentation *admits a polynomial-time* $\log_2 n$*-factor approximation algorithm, where n is the number of vertices of the input graph.*

However, as illustrated in Figure 2, some useful properties of the input digraph may not be preserved by the "flattening" operation. In particular, if $\mathcal{U}(D)$ has treewidth at most r, then the treewidth of $\mathcal{U}(F(D))$ cannot be bounded by a constant in general. Furthermore, the graph $\mathcal{U}(F(D))$ is not necessarily balanced[1] (or planar) if $\mathcal{U}(D)$ is. Therefore, we cannot take advantage of polynomial-time algorithms for Set Cover on balanced incidence graphs or incidence graphs of bounded treewidth. Motivated by the example in Figure 2b we leave as an open question, whether Source Cover on balanced graphs admits a polynomial-time algorithm. By Theorem 8, the existence of such an algorithm implies a polynomial-time algorithm for Robust Matching Augmentation on balanced graphs.

4.1 Source Cover on Chordal Bipartite Graphs

We show that in contrast to the treewidth and balancedness, chordal-bipartiteness is indeed preserved by the flattening operation introduced above. From this we obtain the following result.

► **Theorem 11.** Source Cover *on chordal-bipartite graphs admits a polynomial-time algorithm.*

[1] A graph is called *balanced* if the length of each induced cycle is divisible by four.

To prove the theorem, we show that if $\mathcal{U}(D)$ is chordal-bipartite, so is $\mathcal{U}(F(D))$. The graph $\mathcal{U}(F(D))$ is the incidence graph of a Set Cover instance, whose optimal solutions correspond canonically to the optimal solutions of the Source Cover instance (D). It is known that Set Cover on chordal-bipartite incidence graphs (and more generally, balanced graphs) admits a polynomial-time algorithm: It is possible to use LP-methods and the fact that covering polyhedra of balanced matrices are integral, see [26, pp. 562-573]. Alternatively we can use a combinatorial algorithm by Hoffman et al. [24]. By combining Theorem 8, Proposition 9, and Theorem 11 we obtain the following result.

▶ **Corollary 12.** Robust k-Matching Augmentation *admits a polynomial-time algorithm on chordal-bipartite graphs.*

4.2 Source Cover on Graphs of Bounded Treewidth

We provide a fixed-parameter algorithm for Node Weighted Directed Steiner Tree on acyclic digraphs that is single-exponential in the treewidth of the underlying undirected graph and linear in the instance size. Since Source Cover is a restriction of Node Weighted Directed Steiner Tree on acyclic graphs, this implies a polynomial-time algorithm for Source Cover parameterized by the treewidth of the underlying undirected graph. Let us first recall some definitions related to Steiner problems and tree decompositions.

Node Weighted Directed Steiner Tree
instance: Acyclic digraph $D = (V, A)$, costs $c \in \mathbb{R}^V_{\geq 0}$, terminals $T \subseteq V$, root $r \in V$.
task: Find a minimum-cost subset $F \subseteq V$, such that r is connected to each terminal in $(F, E(F))$.

Arc Weighted Directed Steiner Tree is the corresponding problem, where the costs are on the arcs of the graph. A *tree decomposition* of a graph $G = (V, E)$ is a tree T as follows. Each node $x \in V(T)$ of T has a *bag* $B_x \subseteq V$ of vertices of G such that the following properties hold.

- $\bigcup_{x \in V(T)} B_x = V$.
- If B_x and B_y both contain a vertex $v \in V$, then the bags of all nodes of T in the path between x and y contain v as well. Equivalently, the tree nodes containing vertex v form a connected subtree of T.
- For each edge vw in G there is some bag that contains both v and w. That is, for vertices adjacent in G, the corresponding subtrees have a node in common.

The *width* of a tree decomposition is the size of its largest bag minus one. The *treewidth* $tw(G)$ of G is the minimum width among all possible tree decompositions of G.

To solve the Node Weighted Directed Steiner Tree on acyclic digraphs, we use a simple dynamic-programming algorithm over the tree decomposition of the underlying undirected graph of the input digraph D with n vertices.

▶ **Theorem 13.** Node Weighted Directed Steiner Tree *on acyclic digraphs can be solved in time $O(5^w \cdot w \cdot n)$ if a tree decomposition of $\mathcal{U}(D)$ of width w is provided.*

Note that an optimal tree-decomposition of a graph G can be computed in time $O(2^{O(tw(G)^3)} \cdot n)$ by Bodlaender's famous theorem [7]. Our algorithm intuitively works in the following way and is similar to the dynamic programming algorithm for Dominating Set (see, e.g., [13, Section 7.3.2]). We interpret a solution to Node Weighted Directed Steiner Tree as follows: each vertex of D may be *active* or not. Each active vertex needs a predecessor that is also active, unless it is the root. The cost to activate a vertex is given by the cost function of the Node Weighted Directed Steiner Tree instance. Starting with all terminals active, it is easy to see that Node Weighted Directed Steiner Tree on acyclic graphs is equivalent to the problem of finding a minimum cost active vertex set satisfying the above conditions. We compute an optimal solution in a bottom-up fashion using a so-called *nice* tree decomposition of the input graph.

By a simple reduction, we also obtain an FPT-time algorithm for Arc Weighted Directed Steiner Tree on acyclic digraphs. We just subdivide each arc and assign the cost of the arc to the corresponding new vertex. Each old vertex receives cost zero. This transformation does not increase the treewidth.

Furthermore, we can reduce Source Cover to Node Weighted Directed Steiner Tree by adding a new vertex r and connecting r to each source by an arc. The sources have cost one, while all other vertices have cost zero. The root vertex is r and the set of terminals is the set of sinks. Adding a single new vertex increases the treewidth by at most one. As a consequence of this reduction and Theorem 13, we obtain the following result.

▶ **Corollary 14.** Source Cover *can be solved in time $O(5^w \cdot w \cdot n)$ if a tree-decomposition of $\mathcal{U}(D)$ of width w is provided.*

By combining Theorem 8, Proposition 9, Corollary 14, and the observation that treewidth is monotone under taking minors, we have:

▶ **Corollary 15.** Robust k-Matching Augmentation *parameterized by the treewidth of the input graph is FPT.*

5 Weighted Robust Matching Augmentation

We first demonstrate that the edge-weighted version of Robust Matching Augmentation is substantially more involved than the unit-cost version. To this end, we show that the approximability of Weighted Robust Matching Augmentation essentially corresponds to the approximability of Directed Steiner Forest. The latter problem is defined as follows:

Directed Steiner Forest
instance: Directed graph $G = (V, A)$, k terminal pairs $(s_i, t_i)_{1 \leq i \leq k}$, costs $w \in \mathbf{Z}^A_{\geq 0}$.
task: Find a minimum-cost subgraph $G' \subseteq G$ such that for each $1 \leq i \leq k$, the vertex s_i is connected to t_i in G'.

▶ **Proposition 16.** *Let n' be the number of vertices of a* Weighted Robust Matching Augmentation *instance and n and k be the number of vertices and terminal pairs of a* Directed Steiner Forest *instance, respectively.*

A polynomial-time $f(n')$-factor approximation algorithm for Weighted Robust Matching Augmentation *implies a polynomial-time $f(n + k)$-factor approximation algorithm for* Directed Steiner Forest*. Furthermore, a polynomial-time $f(n)$-factor (resp., $f(k)$-factor) approximation algorithm for* Directed Steiner Forest *implies a polynomial-time $(f(n) + 1)$-factor (resp., $(f(k) + 1)$-factor) approximation algorithm for* Weighted Robust Matching Augmentation.

On the one hand, Proposition 16 implies an $(n^{1/2+\varepsilon} + 1)$-factor approximation algorithm for Weighted Robust Matching Augmentation for every $\varepsilon > 0$, due to the results in [10, 18]. On the other hand, an algorithm achieving a guarantee of $n^{1/3}$ or better for Weighted Robust Matching Augmentation would imply a better approximation

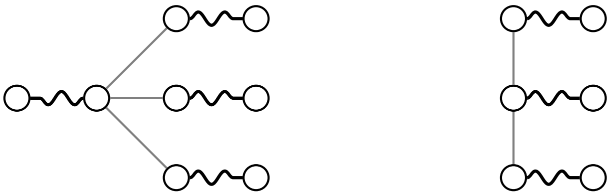


Figure 3 The graphs $K^*_{1,3}$ and P^*_3, each with its unique perfect matching.

algorithm for Directed Steiner Forest, as the number k of distinct terminal pairs is at most $O(n^2)$ and the current best approximation factor for Directed Steiner Forest in terms of n is $n^{2/3+\varepsilon}$ due to a result of Berman et al. [6]. Additionally, by a result of Halperin and Krauthgamer [23], Proposition 16 implies the following lower bound.

▶ **Corollary 17.** *For every $\varepsilon > 0$ Weighted Robust Matching Augmentation does not admit a $\log^{2-\varepsilon}(n)$-factor approximation algorithm unless $\mathsf{NP} \subseteq \mathrm{ZTIME}(n^{\mathsf{polylog}(n)})$.*

Given this negative result we proceed to identify structural features that lead to polynomial-time algorithms for Weighted Robust Matching Augmentation. The main result of this section is a classification of the complexity of the problem Weighted Robust Matching Augmentation on minor-closed graph classes. In particular we show that the problem is NP-hard on a minor-closed class $\mathcal{G}$ of graphs if and only if $\mathcal{G}$ contains at least one of the two graph classes $\mathcal{K}^*$ and $\mathcal{P}^*$, which are defined as follows. Let $K_{1,r}$ be the star graph with r leaves and let P_r be the path on r vertices. For any graph H let H^* be the graph obtained by attaching a leaf to each vertex of H. Then $\mathcal{K}^* := \{K^*_{1,r} \mid r \in \mathbb{N}\}$ and $\mathcal{P}^* := \{P^*_r \mid r \in \mathbb{N}\}$. Note that each graph in $\mathcal{K}^*$ and $\mathcal{P}^*$ has a unique perfect matching. See Figure 3 for an illustration of the graphs $K^*_{1,3}$ and P^*_3.

▶ **Lemma 18.** *Weighted Robust Matching Augmentation is NP-hard on each of the classes $\mathcal{K}^*$ and $\mathcal{P}^*$.*

We complement Lemma 18 by showing that Weighted Robust Matching Augmentation on a class $\mathcal{G}$ of graphs admits a polynomial-time algorithm if $\mathcal{G}$ contains neither $\mathcal{K}^*$ nor $\mathcal{P}^*$.

▶ **Theorem 19.** *Let $\mathcal{G}$ be a class of connected graphs that is closed under connected minors. Then Weighted Robust Matching Augmentation on $\mathcal{G}$ admits a polynomial-time algorithm if and only if there is some $r \in \mathbb{N}$ such that $\mathcal{G}$ contains neither the graph $K^*_{1,r}$ nor P^*_r. The only if part holds under the assumption that $\mathsf{P} \neq \mathsf{NP}$.*

In order to prove Lemma 18, we first show that Weighted Robust Matching Augmentation is NP-hard for graphs consisting only of a perfect matching by a reduction from Robust Matching Augmentation. The hardness of Weighted Robust Matching Augmentation on $\mathcal{K}^*$ and $\mathcal{P}^*$ follows from this result.

Before we give the proof of Theorem 19, we need the following key lemma. The polynomial-time algorithm described in the proof of the lemma uses the fact that Directed Steiner Forest can be solved in polynomial time if the number of terminal pairs is constant [17].

▶ **Lemma 20.** *Let $r \in \mathbb{N}$ be constant and let $\mathcal{T}$ be a class of perfectly matchable trees, each with at most r leaves. Then Weighted Robust Matching Augmentation on $\mathcal{T}$ admits a polynomial-time algorithm.*

We remark that the running time of the algorithm given in Lemma 20 is slicewise polynomial in the number of leaves of the input graph. We can now state the proof of our main result.

Proof of Theorem 19. According to Lemma 18, Weighted Robust Matching Augmentation is NP-hard if $\mathcal{G}$ completely contains the class $\mathcal{K} = \{K^*_{1,r} \mid r \in \mathbb{N}\}$ or the class $\mathcal{P} = \{P^*_r \mid r \in \mathbb{N}\}$. Assuming $\mathsf{P} \neq \mathsf{NP}$, this proves the *only if* statement of the theorem.

To see the *if* statement, let us consider $r \in \mathbb{N}$ such that $\mathcal{G}$ does not contain $K^*_{1,r}$ or P^*_r. First we will reduce the problem to the case when $\mathcal{G}$ contains only trees. For this, let $\mathcal{T}$ be the class of all trees in $\mathcal{G}$ that admit a perfect matching.

▷ Claim 1. There is a polynomial time reduction of Weighted Robust Matching Augmentation on $\mathcal{G}$ to Weighted Robust Matching Augmentation on $\mathcal{T}$.

The key idea for the proof is to define an equivalent instance on an arbitrary tree of G on an adapted cost function. We may hence restrict our attention to Weighted Robust Matching Augmentation on the class $\mathcal{T}$. As the next claim shows, the relevant trees contained in $\mathcal{T}$ have a bounded number of leaves.

▷ Claim 2. There is some number $f(r)$ depending only on r such that every tree in $\mathcal{T}$ has at most $f(r)$ many leaves.

According to the above claims, there is a polynomial reduction of Weighted Robust Matching Augmentation on $\mathcal{G}$ to Weighted Robust Matching Augmentation on a class of trees with a bounded number of leaves. Hence, Lemma 20 implies that Weighted Robust Matching Augmentation on $\mathcal{G}$ can be solved in polynomial time. ◀

6 Conclusion

We presented algorithms for the task of securing matchings of a graph against the failure of a single edge. For this, we established a connection to the classical strong connectivity augmentation problem. Not surprisingly, the unit weight case is more accessible, and we were able to give a $\log_2 n$-factor approximation algorithm, as well as polynomial-time algorithms for graphs of bounded treewidth and chordal-bipartite graphs. For general non-negative weights, we showed a close relation to Directed Steiner Forest in terms of approximability and gave a dichotomy theorem characterizing minor-closed graph classes which allow a polynomial-time algorithm.

In our opinion, the case of a single edge failure is well understood now and so one might go for the case of a constant number of edge failures next. Let us remark that if the number of edge failures is a part of the input, even checking feasibility is NP-hard [14, 25].

References

1. David Adjiashvili. Beating Approximation Factor Two for Weighted Tree Augmentation with Bounded Costs. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2384–2399. Society for Industrial and Applied Mathematics, 2017. doi:10.1137/1.9781611974782.157.
2. David Adjiashvili, Viktor Bindewald, and Dennis Michaels. Robust Assignments via Ear Decompositions and Randomized Rounding. In *43rd International Colloquium on Automata, Languages, and Programming*, volume 55, pages 71:1–71:14, Dagstuhl, Germany, 2016. doi:10.4230/LIPIcs.ICALP.2016.71.

3 David Adjiashvili, Sebastian Stiller, and Rico Zenklusen. Bulk-robust combinatorial optimization. *Mathematical Programming*, 149(1-2):361–390, 2015. doi:10.1007/s10107-014-0760-6.

4 David Adjiashvili and Rico Zenklusen. An $s-t$ connection problem with adaptability. *Discrete Applied Mathematics*, 159(8):695–705, 2011. doi:10.1016/j.dam.2010.12.018.

5 Kristóf Bérczi, Satoru Iwata, Jun Kato, and Yutaro Yamaguchi. Making Bipartite Graphs DM-Irreducible. *SIAM Journal on Discrete Mathematics*, 32:560–590, 2018. doi:10.1137/16M1106717.

6 Piotr Berman, Arnab Bhattacharyya, Konstantin Makarychev, Sofya Raskhodnikova, and Grigory Yaroslavtsev. Approximation algorithms for spanner problems and Directed Steiner Forest. *Information and Computation*, 222:93–107, 2013. 38th International Colloquium on Automata, Languages and Programming (ICALP 2011). doi:10.1016/j.ic.2012.10.007.

7 Hans L. Bodlaender. A Linear-Time Algorithm for Finding Tree-Decompositions of Small Treewidth. *SIAM Journal on Computing*, 25(6):1305–1317, 1996. doi:10.1137/S0097539793251219.

8 Hans L. Bodlaender, Marek Cygan, Stefan Kratsch, and Jesper Nederlof. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. *Information and Computation*, 243:86–111, 2015. doi:10.1016/j.ic.2014.12.008.

9 Robert C. Brigham, Frank Harary, Elizabeth C. Violin, and Jay Yellen. Perfect-matching Preclusion. *Congressus Numerantium*, 174:185–192, 2005.

10 Chandra Chekuri, Guy Even, Anupam Gupta, and Danny Segev. Set Connectivity Problems in Undirected Graphs and the Directed Steiner Network Problem. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 532–541. Society for Industrial and Applied Mathematics, 2008.

11 Eddie Cheng and László Lipták. Matching preclusion for some interconnection networks. *Networks*, 50(2):173–180, 2007. doi:10.1002/net.20187.

12 Joseph Cheriyan, András Sebő, and Zoltán Szigeti. Improving on the 1.5-Approximation of a Smallest 2-Edge Connected Spanning Subgraph. *SIAM Journal on Discrete Mathematics*, 14(2):170–180, 2001. doi:10.1137/S0895480199362071.

13 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

14 Mitre C. Dourado, Dirk Meierling, Lucia D. Penso, Dieter Rautenbach, Fabio Protti, and Aline Ribeiro de Almeida. Robust recoverable perfect matchings. *Networks*, 66(3):210–213, 2015. doi:10.1002/net.21624.

15 Andrew L Dulmage and Nathan S Mendelsohn. Coverings of bipartite graphs. *Canadian Journal of Mathematics*, 10(4):516–534, 1958.

16 Kapali P. Eswaran and Robert E. Tarjan. Augmentation problems. *SIAM Journal on Computing*, 5(4):653–665, 1976. doi:10.1137/0205044.

17 Jon Feldman and Matthias Ruhl. The Directed Steiner Network Problem is Tractable for a Constant Number of Terminals. *SIAM Journal on Computing*, 36(2):543–561, 2006. doi:10.1137/S0097539704441241.

18 Moran Feldman, Guy Kortsarz, and Zeev Nutov. Improved Approximating Algorithms for Directed Steiner Forest. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 922–931. Society for Industrial and Applied Mathematics, 2009.

19 Samuel Fiorini, Martin Groß, Jochen Könemann, and Laura Sanità. Approximating Weighted Tree Augmentation via Chvátal-Gomory Cuts. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 817–831, 2018. doi:10.1137/1.9781611975031.53.

20 András Frank and Tibor Jordán. Graph Connectivity Augmentation. In *Handbook of Graph Theory, Combinatorial Optimization, and Algorithms*, chapter 14, pages 313–346. CRC Press, 2015.

21 Greg N Frederickson and Joseph Ja'Ja'. Approximation Algorithms for Several Graph Augmentation Problems. *SIAM Journal on Computing*, 10(2):270–283, 1981. doi:10.1137/0210019.

22 Harold N. Gabow, Michel X. Goemans, Éva Tardos, and David P. Williamson. Approximating the smallest k-edge connected spanning subgraph by LP-rounding. *Networks*, 53(4):345–357, 2009. doi:10.1002/net.20289.

23 Eran Halperin and Robert Krauthgamer. Polylogarithmic inapproximability. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pages 585–594, 2003. doi:10.1145/780542.780628.

24 Alan J. Hoffman, Anthonius W. J. Kolen, and Michel Sakarovitch. Totally-Balanced and Greedy Matrices. *SIAM Journal on Algebraic Discrete Methods*, 6(4):721–730, 1985. doi:10.1137/0606070.

25 Mathieu Lacroix, A. Ridha Mahjoub, Sébastien Martin, and Christophe Picouleau. On the NP-completeness of the perfect matching free subgraph problem. *Theoretical Computer Science*, 423:25–29, 2012. doi:10.1016/j.tcs.2011.12.065.

26 Laurence A. Wolsey and George L. Nemhauser. *Integer and Combinatorial Optimization*. Wiley Series in Discrete Mathematics and Optimization. Wiley, 1999.