A Strongly Polynomial-Time Algorithm for Weighted General Factors with Three Feasible Degrees

Shuai Shao
School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing, China

Stanislav Živný
Department of Computer Science, University of Oxford, UK

Abstract

General factors are a generalization of matchings. Given a graph G with a set $\pi(v)$ of feasible degrees, called a degree constraint, for each vertex v of G, the general factor problem is to find a (spanning) subgraph F of G such that $\deg_F(v) \in \pi(v)$ for every v of G. When all degree constraints are symmetric Δ-matroids, the problem is solvable in polynomial time. The weighted general factor problem is to find a general factor of the maximum total weight in an edge-weighted graph. Strongly polynomial-time algorithms are only known for weighted general factor problems that are reducible to the weighted matching problem by gadget constructions.

In this paper, we present a strongly polynomial-time algorithm for a type of weighted general factor problems with real-valued edge weights that is provably not reducible to the weighted matching problem by gadget constructions. As an application, we obtain a strongly polynomial-time algorithm for the terminal backup problem by reducing it to the weighted general factor problem.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases matchings, factors, edge constraint satisfaction problems, terminal backup problem, delta matroids

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2023.57

Related Version *Full Version*: https://arxiv.org/abs/2301.11761

Funding The research leading to these results has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 714532). This research was also funded by UKRI EP/X024431/1. All data is provided in full in the results section of this paper. Part of the work was done while the first author was a postdoctoral research associate at the University of Oxford.

1 Introduction

A matching in an undirected graph is a subset of the edges that have no vertices in common, and it is perfect if its edges cover all vertices of the graph. Graph matching is one of the most studied problems both in graph theory and combinatorial optimization, with beautiful structural results and efficient algorithms described, e.g., in the monograph of Lovász and Plummer [38] and in relevant chapters of standard textbooks [43, 34]. In particular, the *weighted (perfect) matching problem* is to find a (perfect) matching of the maximum total weight for a given graph of which each edge is assigned a weight. This problem can be solved in polynomial time by the celebrated Edmonds' blossom algorithm [14, 15]. Since then, a number of more efficient algorithms have been developed [20, 35, 31, 8, 22, 27, 24, 23, 26, 29]. Table III of [10] gives a detailed review of these algorithms.

The *f-factor problem* is a generalization of the perfect matching problem in which one is given a non-negative integer $f(v)$ for each vertex $v \in V$ of $G = (V, E)$. The task is to find a (spanning) subgraph $F = (V_F, E_F)$ of G such that $\deg_F(v) = f(v)$ for every $v \in V$.[1] The case $f(v) = 1$ for every $v \in V$ is the perfect matching problem. This problem, as well as the weighted version, can be solved efficiently by a gadget reduction to the perfect matching problem [16]. In addition, Tutte gave a characterization of graphs having an f-factor [47], which generalizes his characterization theorem for perfect matchings [46]. Subsequently, the study of graph factors has attracted much attention with many variants of graph factors, e.g., b-matchings, $[a, b]$-factors, (g, f)-factors, parity (g, f)-factors, and anti-factors introduced, and various types of characterization theorems proved for the existence of such factors. We refer the reader to the book [1] and the survey [40] for a comprehensive treatment of the developments on the topic of graph factors.

In the early 1970s, Lovász introduced a generalization of the above factor problems [36, 37], for which we will need a few definitions. For any nonnegative integer n, let $[n]$ denote $\{0, 1, \dots, n\}$. A degree constraint D of arity n is a subset of $[n]$.[2] We say that a degree constraint D has a gap of length k if there exists $p \in D$ such that $p + 1, \dots, p + k \notin D$ and $p + k + 1 \in D$. An instance of the *general factor problem* (GFP) [36, 37] is given by a graph $G = (V, E)$ and a mapping π that maps every vertex $v \in V$ to a degree constraint $\pi(v) \subseteq [\deg_G(v)]$ of arity $\deg_G(v)$. The task is to find a subgraph, if one exists, F of G such that $\deg_F(v) \in \pi(v)$ for every $v \in V$. The case $\pi(v) = \{0, 1\}$ for every $v \in V$ is the matching problem, and the case $\pi(v) = \{1\}$ for every $v \in V$ is the perfect matching problem. Cornuéjols showed that the GFP is solvable in polynomial time if each degree constraint has gaps of length at most 1 [7]. When a degree constraint having a gap of length at least 2 occurs, the GFP is NP-complete [37, 7] except for the case when all constraints are either 0-*valid* or 1-*valid*. A degree constraint D of arity k is 0-valid if $0 \in D$, and 1-valid if $k \in D$. When all constraints are 0-*valid*, the empty graph is a factor. When all constraints are 1-*valid*, the underlying graph is a factor of itself. In both cases, the GFP is trivially tractable.

In this paper, we consider the *weighted general factor problem* (WGFP) where each edge is assigned a real-valued weight and the task is to find a general factor of the maximum total weight. We suppose that each degree constraint has gaps of length at most 1 for which the unweighted GFP is known to be polynomial-time solvable. Some cases of the WGFP are reducible to the weighted matching or perfect matching problem by gadget constructions, and hence are polynomial-time solvable. In these cases, the degree constraints are called *matching-realizable* (see Definition 18). For instance, the degree constraint $D = [b]$ where $b > 0$, for b-matchings is matching realizable [48]. The weighted b-matching problem is interesting in its own right in combinatorial optimization and has been well studied with many elaborate algorithms developed [41, 39, 21, 3, 25]. Besides b-matchings, Cornuéjols showed that the *parity interval* constraint $D = \{g, g + 2, \dots, f\}$ where $f \geq g \geq 0$ and $f \equiv g \mod 2$, is matching realizable [7], and Szabó showed that the *interval* constraint $D = \{g, g + 1, \dots, f\}$ where $f \geq g \geq 0$, for (g, f)-factors is matching realizable [45]. Thus, the WGFP where each degree constraint is an interval or a parity interval is reducible to the weighted matching problem (with some vertices required to have degree exactly 1) and hence solvable in polynomial-time by Edmonds' algorithm, although Szabó gave a different

[1] In graph theory, a graph factor is usually a spanning subgraph. Here, without causing ambiguity, we allow F to be an arbitrary subgraph including the empty graph and we adapt the convention that $\deg_F(v) = 0$ if $v \in V \setminus V_F$.

[2] We always associate a degree constraint with an arity. Two degree constraints are different if they have different arities although they may be the same set of integers.

algorithm for this problem [45]. By reducing the WGFP with interval and parity interval constraints to the weighted (g, f)-factor problem, a faster algorithm was obtained in [11] based on Gabow's algorithm [21].

In [45], Szabó further conjectured that the WGFP is solvable in polynomial time without requiring each degree constraint should be an interval or a parity interval, as long as each degree constraint has gaps of length at most 1. To prove the conjecture, a natural question is then the following: *Are there other WGFPs that are polynomial-time solvable by a gadget reduction to weighted matchings?* In other words, *are there other degree constraints that are matching realizable?* In this paper, we show that the answer is *no*.

▶ **Theorem 1.** *A degree constraint with gaps of length at most* 1 *is matching realizable if and only if it is an interval or a parity interval.*

Previous results beyond matchings realizable degree constraints. With the answer to the above question being negative, new algorithms need to be devised for the WGFP with degree constraints that are not intervals or parity intervals. Unlike the weighted matching problem and the weighted b-matching problem for which various types of algorithms have been developed, only a few algorithms have been presented for the more general and challenging WGFP: For the cardinality version of WGFP, i.e., the WGFP where each edge is assigned weight 1, Dudycz and Paluch introduced a polynomial-time algorithm for this problem with degree constrains having gaps of length at most 1, which leads to a pseudo-polynomial-time algorithm for the WGFP with non-negative integral edge weights [11].

The algorithm in [11] is based on a structural result showing that if a factor is not optimal, then a factor of larger weight can be found by a local search, which can be done in polynomial time. However, it is not clear how much larger the weight of the new factor is. In order to get an optimal factor, the algorithm needs to repeat local searches iteratively until no better factors can be found, and the number of local searches is bounded by the total edge weight, which makes the algorithm pseudo-polynomial-time. Later, in an updated version [12], by carefully assigning edge weights, the algorithm was improved to be weakly polynomial-time with a running time $O(\log Wmn^6)$, where W is the largest edge weight, m is the number of edges and n is the number of vertices. Later, Kobayash extended the algorithm to a more general setting called jump system intersections [33].

Our main contribution. Independently of [12], in this paper, we make a step towards a *strongly* polynomial-time algorithm for the WGPF. Let $p \geq 0$ be an arbitrary integer. Consider the following two types of degree constraints $\{p, p+1, p+3\}$ and $\{p, p+2, p+3\}$ (of arbitrary arity). We will call them *type-1* and *type-2* respectively. These are the "smallest" degree constraints that are not matching realizable.

▶ **Theorem 2.** *There is a strongly polynomial-time algorithm for the* WGFP *with real-valued edge weights where each degree constraint is an interval, a parity interval, a type-1, or a type-2 (of arbitrary arities). The algorithm runs in time $O(n^6)$ for a graph with n vertices.*

The requirement of degree constraints in our result may look overly specific. However, the scope of our algorithm is not narrow. First, our result implies a complexity dichotomy for the WGFP on all subcubic graphs (see the following Theorem 3), which for many is a large and interesting class of graphs. More importantly, there are interesting problems arising from applied areas that are encompassed by the WGPF with constraints considered in this paper.

For instance, the *terminal backup problem* from network design is the following problem. Given a graph consisting of terminal nodes, non-terminal nodes, and edges with non-negative costs. The goal is to find a subgraph with the minimum total cost such that each terminal node is connected to at least one other terminal node (for the purpose of backup in applications). It is known that an optimal solution of the terminal backup problem consists of edge-disjoint paths containing 2 terminals and stars containing 3 terminals [49]. In other words, in an optimal subgraph of the terminal backup problem, each terminal node has degree 1 and each non-terminal node has degree 0, 2 or 3. Thus, the terminal backup problem can be expressed as a WGFP with degree constraints $\{1\}$ and $\{0, 2, 3\}$ (both of arbitrary arities). A weakly polynomial-time algorithm was given for the terminal backup problem in [2]. Our result gives a strongly polynomial-time algorithm for this problem.

In addition, our algorithm gives a tractability result for the WGFP with degree constraints that are provably not matching realizable, thus going beyond existing algorithms. The algorithm is a recursive algorithm, reducing the problem to a smaller sub-problem of itself by fixing the parity of degree constraints on vertices. Its correctness is based on a delicate structural result, which is stronger than that of [12].[3] Equipped with this result, our algorithm can directly find an *optimal* factor (not just a better one) of an instance of a larger size by performing only one local search from an optimal factor of a smaller instance. Here, the important part is not how to find a better factor by local search (the main result of [12]) but rather how to ensure that the better factor obtained by only one local search is actually optimal under certain assumptions. This is the key to making our algorithm strongly polynomial. In addition, as a by-product, we give a simple proof of the result of [12] for the special case of WGFP with interval, parity interval, type-1 and type-2 degree constraints by reducing the problem to WGFP on subcubic graphs and utilizing the equivalence between 2-vertex connectivity and 2-edge connectivity of subcubic graphs.

Relation with (edge) constraint satisfaction problems. The graph factor problem is encompassed by a special case of the Boolean constraint satisfaction problem (CSP), called edge-CSP, in which every variable appears in exactly two constraints [30, 17]. When every constraint is symmetric (i.e, the value of the constraint only depends on the Hamming weight of its input), the Boolean edge-CSP is a graph factor problem.

For general Boolean edge-CSPs, Feder showed that the problem is NP-complete if a constraint that is not a Δ-matroid occurs, except for those that are tractable by Schaefer's dichotomy theorem for Boolean CSPs [42]. In a subsequent line of work [9, 28, 18, 13], tractability of Boolean edge-CSPs has been established for special classes of Δ-matroids, most recently for even Δ-matroids [32]. A complete complexity classification of Boolean edge-CSPs is still open with the conjecture that all Δ-matroids are tractable. A degree constraint (i.e., a symmetric constraint) is a Δ-matroid if and only if it has gaps of length at most 1. Thus, the above conjecture holds for symmetric Boolean edge-CSPs by Cornuéjols' result on the general factor problem [7]. A complexity classification for weighted Boolean edge-CSPs is certainly a more challenging goal: The complexity of weighted Boolean edge-CSPs with even Δ-matroids as constraints is still open. Our result in Theorem 2 gives a tractability result for weighted Boolean edge-CSPs with certain symmetric Δ-matroids as constraints. Combining our main result with known results on Boolean valued CSPs [6], we obtain a complexity dichotomy for weighted Boolean edge-CSPs with symmetric constraints of arity no more than 3, i.e., the WGFP on subcubic graphs.

[3] The result in [12] holds for the more general WGFP with all degree constraints having gaps of length at most 1, while our result only works for the WGFP with interval, parity interval, type-1 and type-2 degree constraints.

Let D be a degree constraint of arity at most 3. If $D \neq \{0,3\}$, then D is an interval, a parity interval, a type-1, or a type-2. Thus, if the constraint $\{0,3\}$ (of arity 3) does not occur, then the WGFP is strongly polynomial-time solvable by our main theorem. Otherwise, the constraint $\{0,3\}$ occurs. In this case, for a vertex v labeled by $\{0,3\}$, the three edges incident to it must take the same assignments in a feasible factor (i.e., the three edges are all either present or absent in any factor). Thus, the vertex v can be viewed as a Boolean variable and it appears in three other degree constraints connected to it. By viewing all vertices with $\{0,3\}$ as variables appearing three times and the other edges as variables appearing twice, the WGFP becomes a special case of valued CSPs where some variables appear three times and the other variables appear twice. It is known that once variables are allowed to appear three times in a CSP, then they can appear arbitrarily many times [9]. Thus, the WGPF with $\{0,3\}$ occurring is equivalent to a standard (non-edge) CSP [19]. By the dichotomy theorem for valued CSPs [6], one can check that the problem is tractable if and only if for every degree constraint D of arity $k \leq 3$, $D \subseteq \{0,k\}$. Thus, we have the following complexity dichotomy.

▶ **Theorem 3.** *The* WGFP *on subcubic graphs is strongly polynomial-time solvable if*

1. *the degree constraint $\{0,3\}$ of arity 3 does not occur,*
2. *or for every degree constraint D of arity $k \leq 3$, $D \subseteq \{0,k\}$.*

Otherwise, the problem is NP-hard.

Organization. In Section 2, we present basic definitions and notation. In Section 3, we describe our algorithm and give a structural result for the WGFP that ensures the correctness and the strongly polynomial-time running time of our algorithm. In Section 4, we introduce basic augmenting subgraphs as an analogy of augmenting paths for weighed matchings and give a proof of the structural result. The proof is based on a result regarding the existence of certain basic factors for subcubic graphs, for which we give a proof sketch in Section 5. Finally, we discuss matching realizability and its relation with Δ-matroids in Appendix A. All omitted proofs can be found in the full version [44].

2 Preliminaries

Let $\mathcal{D}$ be a (possibly infinite) set of degree constraints.

▶ **Definition 4.** *The* weighted general factor problem *parameterized by $\mathcal{D}$, denoted by* WGFP($\mathcal{D}$)*, is the following computational problem. An instance is a triple $\Omega = (G, \pi, \omega)$, where $G = (V, E)$ is a graph, $\pi : V \to \mathcal{D}$ assigns to every $v \in V$ a degree constraint $D_v \in \mathcal{D}$ of arity $\deg_G(v)$, and $\omega : E \to \mathbb{R}$ assigns to every $e \in E$ a real-valued weight $w(e) \in \mathbb{R}$. The task is to find, if one exists, a general factor F of G such that the total weight of edges in F is maximized.*

The general factor problem GFP($\mathcal{D}$) *is the decision version of* WGFP($\mathcal{D}$)*; i.e., deciding whether a general factor exists or not.*

Suppose that $\Omega = (G, \pi, \omega)$ is a WGFP instance. If F is a general factor of G under π, then we say that F is a factor of Ω, denoted by $F \in \Omega$. In terms of this inclusion relation, Ω can be viewed as a set of subgraphs of G. We extend the edge weight function ω to subgraphs of G. For a subgraph H of G, its weight $\omega(H)$ is $\sum_{e \in E(H)} \omega(e)$ ($\omega(H) = 0$ if H is the empty graph). If H contains an isolated vertex v, then $\omega(H) = \omega(H')$, where H' is the graph obtained from H by removing v. Moreover, $H \in \Omega$ if and only if $H' \in \Omega$. In the following, without other specification, we always assume that a factor does not contain any isolated vertices. The optimal value of Ω, denoted by $\mathrm{Opt}(\Omega)$, is $\max_{F\in\Omega}\omega(F)$. We define $\mathrm{Opt}(\Omega) = -\infty$ if Ω has no factor. A factor F of Ω is *optimal* in Ω if $\omega(F) = \mathrm{Opt}(\Omega)$.

For a WGFP instance $\Omega' = (G', \pi', \omega')$, where $G' \subseteq G$[4] and ω' is the restriction of ω on the edges of G', we say Ω' is a *sub-instance* of Ω, denoted by $\Omega' \subseteq \Omega$, if $F \in \Omega$ for every $F \in \Omega'$. In particular, Ω' is a subset of Ω by viewing them as two sets of subgraphs of G. If $\Omega' \subseteq \Omega$, then $\mathrm{Opt}(\Omega') \leq \mathrm{Opt}(\Omega)$. For two WGFP instances $\Omega_1 = (G, \pi_1, \omega)$ and $\Omega_2 = (G, \pi_2, \omega)$, we use $\Omega_1 \cup \Omega_2$ to denote the union of factors of these two instances, i.e., $\Omega_1 \cup \Omega_2 = \{F \subseteq G \mid F \in \Omega_1 \text{ or } F \in \Omega_2\}$, and $\Omega_1 \cap \Omega_2$ to denote the intersection, i.e., $\Omega_1 \cap \Omega_2 = \{F \subseteq G \mid F \in \Omega_1 \text{ and } F \in \Omega_2\}$. Note that $\Omega_1 \cup \Omega_2$ and $\Omega_1 \cap \Omega_2$ are sets of subgraphs of G and may not define WGFP instances on G.

We use $\mathcal{G}_1$ and $\mathcal{G}_2$ to denote the set of degree constraints that are intervals and parity intervals, respectively, and $\mathcal{T}_1$ and $\mathcal{T}_2$ to denote the set of degree constraints that are type-1 and type-2, respectively. Let $\mathcal{G} = \mathcal{G}_1 \cup \mathcal{G}_2$ and $\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2$. In this paper, we study the problem WGFP($\mathcal{G} \cup \mathcal{T}$).

Let $H_1 = (V_1, E_1)$ and $H_2 = (V_2, E_2)$ be two subgraphs of G. The symmetric difference graph $H_1 \Delta H_2$ is the induced subgraph of G induced by the edge set $E_1 \Delta E_2$. Note that there are no isolated vertices in a symmetric difference graph. When $E_1 \cap E_2 = \emptyset$, we may write $H_1 \Delta H_2$ as $H_1 \cup H_2$. When $E_2 \subseteq E_1$, we may write $H_1 \Delta H_2$ as $H_1 \backslash H_2$. A *subcubic* graph is defined to be a graph where every vertex has degree $1, 2$ or 3. Unless stated otherwise, we use V_G and E_G to denote the vertex set and the edge set of a graph G, respectively.

3 Algorithm

We give a recursive algorithm for the problem WGFP($\mathcal{G} \cup \mathcal{T}$), using the problems WGFP($\mathcal{G}$) and the decision problem GFP($\mathcal{G} \cup \mathcal{T}$) as oracles.

Given an instance $\Omega = (G, \pi, \omega)$ of WGFP($\mathcal{G} \cup \mathcal{T}$), we define the following sub-instances of $\Omega = (G, \pi, \omega)$ that will be used in the recursion. Recall that V_G denotes the vertex set of the graph G. Let T_Ω denote the set $\{v \in V_G \mid \pi(v) \in \mathcal{T}\}$. (We may omit the subscript Ω of T_Ω when it is clear from the context.)

For every vertex $v \in T_\Omega$, we split the instance Ω in two by splitting the degree constraint $\pi(v)$ in two parity intervals. More precisely, we define

$$\begin{aligned} &D_v^0 = \{p_v + 1, p_v + 3\} \text{ and } D_v^1 = \{p_v\} && \text{if} \quad \pi(v) = \{p_v, p_v + 1, p_v + 3\} \in \mathcal{T}_1; \\ &D_v^0 = \{p_v, p_v + 2\} \text{ and } D_v^1 = \{p_v + 3\} && \text{if} \quad \pi(v) = \{p_v, p_v + 2, p_v + 3\} \in \mathcal{T}_2. \end{aligned}$$

We have $D_v^0, D_v^1 \in \mathcal{G}_2$. For $i \in \{0, 1\}$ and $v \in T_\Omega$, we define $\Omega_v^i = (G, \pi_v^i, \omega)$ to be the sub-instance of Ω where $\pi_v^i(x) = \pi(x)$ for every $x \in V_G \backslash \{v\}$ and $\pi_v^i(v) = D_v^i$. Then, for every $v \in T_\Omega$, we have $\Omega_v^0 \cap \Omega_v^1 = \emptyset$ and $\Omega_v^0 \cup \Omega_v^1 = \Omega$. Moreover, $T_{\Omega_v^0} = T_{\Omega_v^1} = T_\Omega \backslash \{v\}$.

Let F be a factor of Ω. Similarly to above, one can partition Ω into $2^{|T_\Omega|}$ many sub-instances according to F such that each one is an instance of WGFP($\mathcal{G}$) – for each $v \in T_\Omega$, we choose one of the two splits of $\pi(v)$ as above. (We note that the algorithm will not consider all exponentially many sub-instances.) In detail, for every vertex $v \in T_\Omega$, we define $D_v^F = D_v^i$ where $\deg_F(v) \in D_v^i$ as follows:

[4] We use the term "subgraph" and notation $G' \subseteq G$ throughout for the standard meaning of a "normal" subgraph i.e., if $G = (V', E')$ and $G = (V, E)$ then $G' \subseteq G$ means $V' \subseteq V$ and $E' \subseteq E$.

$$
\begin{array}{llll}
D_v^F = \{p_v\} & \text{if} & \pi(v) = \{p_v, p_v+1, p_v+3\} \in \mathcal{T}_1 \text{ and } & \deg_F(v) = p_v, \\
D_v^F = \{p_v+1, p_v+3\} & \text{if} & \pi(v) = \{p_v, p_v+1, p_v+3\} \in \mathcal{T}_1 \text{ and } & \deg_F(v) \neq p_v; \\
D_v^F = \{p_v+3\} & \text{if} & \pi(v) = \{p_v, p_v+2, p_v+3\} \in \mathcal{T}_2 \text{ and } & \deg_F(v) = p_v+3, \\
D_v^F = \{p_v, p_v+2\} & \text{if} & \pi(v) = \{p_v, p_v+2, p_v+3\} \in \mathcal{T}_2 \text{ and } & \deg_F(v) \neq p_v+3.
\end{array}
$$

By definition, $\deg_F(v) \in D_v^F \subseteq \pi(v)$ and $D_v^F \in \mathcal{G}_2$. In fact, D_v^F is the maximal set such that $\deg_F(v) \in D_v^F \subseteq \pi(v)$ and $D_v^F \in \mathcal{G}_2$. One can also check that for every $v \in T_\Omega$, $\pi(v) \backslash D_v^F \in \mathcal{G}_2$, and moreover for every $p \in D_v^F$ and $q \in \pi(v) \backslash D_v^F$, $p \not\equiv q \mod 2$.

For every $W \subseteq T_\Omega$, we define $\Omega_W^F = (G, \pi_W^F, \omega)$ to be the sub-instance of Ω where $\pi_W^F(v) = \pi(v) \backslash D_v^F$ for $v \in W$, $\pi_W^F(v) = D_v^F$ for $v \in T_\Omega \backslash W$, and $\pi_W^F(v) = \pi(v)$ for $v \in V \backslash T_\Omega$. Then for every W, Ω_W^F is an instance of WGFP($\mathcal{G}$). Moreover, we have $\cup_{W \subseteq T} \Omega_W^F = \Omega$ and $\Omega_{W_1}^F \cap \Omega_{W_2}^F = \emptyset$ for every $W_1 \neq W_2$. Thus, $\{\Omega_W^F\}_{W \subseteq T_\Omega}$ is a partition of Ω (viewed as a set of subgraphs of G). When $W = \emptyset$, we write Ω_W^F as Ω^F.

Our algorithm is given in Algorithm 1.

Algorithm 1 Finding an optimal factor for an instance of WGFP($\mathcal{G} \cup \mathcal{T}$).

Function `Decision`**:**
  Input : An instance $\Omega = (G, \pi, \omega)$ of WGFP($\mathcal{G} \cup \mathcal{T}$).
  Output : A factor of Ω, or "No" if Ω has no factor.

Function `Optimization`**:**
  Input : An instance $\Omega = (G, \pi, \omega)$ of WGFP($\mathcal{G}$).
  Output : An optimal factor of Ω, or "No" if Ω has no factor.

Function `Main`**:**
  Input : An instance $\Omega = (G, \pi, \omega)$ of WGFP($\mathcal{G} \cup \mathcal{T}$).
  Output : An optimal factor $F \in \Omega$, or "No" if Ω has no factor.

  $T \leftarrow \{v \in V \mid \pi(v) \in \mathcal{T}\}$;
  if T is the empty set **then**
    return `Optimization` (Ω);
  else
    Arbitrarily pick $u \in T$;
    if `Decision` (Ω_u^0) returns "No" **then**
      return `Main` (Ω_u^1);
    else
      $F^{\text{opt}} \leftarrow$ `Main` (Ω_u^0);
      foreach $v \in T$ **do**
        `// Elements of` T `can be traversed in an arbitrary order.`
        $W \leftarrow \{u\} \cup \{v\}$;
        if `Optimization`($\Omega_W^{F^{\text{opt}}}$) $\neq$ "No" **then** $F' \leftarrow$ `Optimization`($\Omega_W^{F^{\text{opt}}}$);
        if $\omega(F') > \omega(F^{\text{opt}})$ **then** $F^{\text{opt}} \leftarrow F'$;
      end
      return F^{opt};
    end
  end

The key that makes our algorithm running strongly polynomial-time is the following structural result (Theorem 5) for the problem WGFP($\mathcal{G} \cup \mathcal{T}$). It says that given an optimal factor F of Ω_u^0 for some $u \in T_\Omega$, if F is not optimal in Ω, then we can directly find an optimal factor of Ω by searching at most n sub-instances of Ω which are in WGFP($\mathcal{G}$). Note that the number of searches is independent of the edge weights. Thus, the problem of finding an optimal factor in Ω can be reduced to finding an optimal factor in Ω_u^0, where there is one fewer vertex u with constraints in $\mathcal{T}$. By recursively reducing an instance to another with fewer vertices with constraints in $\mathcal{T}$, we eventually get an instance of WGFP($\mathcal{G}$) which can be solved in polynomial-time. This leads to a strongly polynomial-time algorithm for finding an optimal factor.

▶ **Theorem 5.** *Suppose that $\Omega = (G, \pi, \omega)$ is an instance of* WGFP($\mathcal{G} \cup \mathcal{T}$)*, F is a factor of Ω and F is optimal in Ω_u^0 for some $u \in T_\Omega$. Then a factor F' is optimal in Ω if and only if $\omega(F') \geq \omega(F)$ and $\omega(F') \geq \mathrm{Opt}(\Omega_W^F)$ for every W where $u \in W \subseteq T_\Omega$ and $|W| = 1$ or 2.*

In other words, if F is not optimal in Ω, then there is an optimal factor of Ω which belongs to Ω_W^F for some W where $u \in W \subseteq T_\Omega$ and $|W| = 1$ or $|W| = 2$.

▶ Remark 6. This result is *stronger* than the main result (Theorem 2) of [12], and it is *not* simply implied by [12]. To clarify this, we give a simple proof outline of Theorem 5 here.

In order to prove Theorem 5, it suffices to prove the direction that if $\omega(F') \geq \omega(F)$ and $\omega(F') \geq \mathrm{Opt}(\Omega_W^F)$ for every W where $u \in W \subseteq T_\Omega$ and $|W| = 1$ or 2, then F' is optimal in Ω. We prove this by contradiction. Suppose that F' is not optimal in Ω, and F^* is an optimal factor of Ω. Then, $\omega(F^*) > \omega(F') \geq \mathrm{Opt}(\Omega_W^F)$ for every $W \subseteq T_\Omega$ where $|W| \leq 2$. Also, $\omega(F^*) \notin \Omega_u^0$ since $\omega(F^*) > \omega(F) = \mathrm{Opt}(\Omega_u^0)$. Thus, $\deg_{F^*}(u) \not\equiv \deg_F(u) \mod 2$.

By [12], a *canonical path* $M \subseteq F \Delta F^*$ with positive weight[5] can be found, and then $F \Delta M$ is a factor of Ω with larger weight than F and $F \Delta M \in \Omega_W^F$ for some $W \subseteq T_\Omega$ where $|W| \leq 2$. However, this does not lead to a contradiction. To get a contradiction, we need to show that the positive weighted canonical path M (a basic augmenting subgraph) further satisfies $\deg_M(u) \equiv 0 \mod 2$. Then, $\deg_{F\Delta M}(u) \equiv \deg_F(u) \mod 2$. Thus, $F \Delta M$ is a factor with larger weight than F and $F \Delta M \in \Omega_u^0$, which contradicts with F being optimal in Ω_u^0.

The existence of a basic augmenting subgraph M satisfying $\deg_M(u) \equiv 0 \mod 2$ is formally stated in the second property of Lemma 12. The main technical part of the paper (Section 5.2 of the full paper) is devoted to prove it. In Section 5 of this short version, we give an example to illustrate the proof ideas. The existence of such a basic augmenting subgraph is highly non-trivial. In fact, it does *not* hold anymore after a subtle change of the condition "F is optimal in $\Omega_u^{\mathbf{0}}$" to "F is optimal in $\Omega_u^{\mathbf{1}}$" for some $u \in T_\Omega$. We give the following example (see Figure 1) to show this.

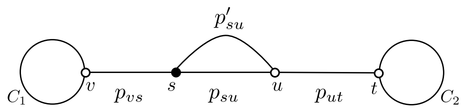

Figure 1 An example that violates Theorem 5 when F is optimal in Ω_u^1 instead of Ω_u^0.

In this instance, $\pi(u) = \pi(v) = \pi(t) = \{0, 1, 3\}$ (denoted by hollow nodes) and $\pi(s) = \{0, 2, 3\}$ (denoted by the solid node), and $\omega(C_1) = \omega(p_{vs}) = \omega(p_{su}) = \omega(p'_{su}) = \omega(p_{ut}) = \omega(C_2) = 1$. Inside the cycles C_1 and C_2, and the paths p_{vs}, p_{su}, p_{ut}, and p'_{su}, there are

[5] See definition 3 of [12]. They are defined as basic augmenting subgraphs (Definition 11) in this paper.

other vertices of degree 2 with the degree constraint $\{0, 2\}$ so that the graph G is simple. We omit these vertices of degree 2 in Figure 1. In this case, $T_\Omega = \{u, v, s, t\}$. Consider the sub-instance $\Omega_u^1 = (G, \pi_u^1, \omega)$. We have $\pi_u^1(u) = D_u^1 = \{0\}$ since $\pi(u) = \{0, 1, 3\}$. Then, the only factor F of Ω_u^1 is the empty graph (assuming there are no isolated vertices in factors), and F is not optimal in Ω. Also, the only optimal factor of Ω is the graph G and $G \in \Omega_{T_\Omega}^F$ where $|T_\Omega| = 4$. Clearly, $\deg_G(u) \not\equiv \deg_F(u) \mod 2$. One can check that for any factor F' of Ω with larger weight than F, $\deg_{F'}(u) \not\equiv \deg_F(u) \mod 2$. In other words, there is no basic augmenting subgraph M such that $\deg_M(u) \equiv 0 \mod 2$. Moreover, one can check that in this case, Theorem 5 also does not hold. In other words, the existence of a basic augmenting subgraph satisfying $\deg_M(u) \equiv 0 \mod 2$ is crucial for the correctness of Theorem 5.

Using Theorem 5, we now prove that Algorithm 1 is correct.

▶ **Lemma 7.** *Given an instance $\Omega = (G, \pi, \omega)$ of* WGFP($\mathcal{G}, \mathcal{T}$)*, Algorithm 1 returns either an optimal factor of Ω, or "No" if Ω has no factor.*

Proof. Recall that for an instance $\Omega = (G, \pi, \omega)$, we define $T_\Omega = \{v \in V_G \mid \pi(v) \in \mathcal{T}\}$ where V_G is the vertex set of G . We prove the correctness by induction on the $|T_\Omega|$.

If $|T_\Omega| = 0$, Ω is an instance of WGFP($\mathcal{G}$). Algorithm 1 simply returns `Optimization` (Ω). By the definition of the function `Optimization`, the output is correct.

Suppose that Algorithm 1 returns correct results for all instances Ω' of WGFP($\mathcal{G}, \mathcal{T}$) where $|T_{\Omega'}| = k$. We consider an instance Ω of WGFP($\mathcal{G}, \mathcal{T}$) where $|T_\Omega| = k+1$. Algorithm 1 first calls the function `Decision` (Ω_u^0) for some arbitrary $u \in T$.

We first consider the case that `Decision` (Ω_u^0) returns "No". By the definition, Ω_u^0 has no factor. Moreover, since $\Omega = \Omega_u^0 \cup \Omega_u^1$, we have $F \in \Omega$ if and only if $F \in \Omega_u^1$. Then, a factor $F \in \Omega_u^1$ is optimal in Ω if and only if it is optimal in Ω_u^1. Note that Ω_u^1 is an instance of WGFP($\mathcal{G}, \mathcal{T}$) where $|T_{\Omega_u^1}| = k$. By the induction hypothesis, Algorithm 1 returns a correct result `Main` (Ω_u^1) for the instance Ω_u^1, which is also a correct result for the instance Ω.

Now, we consider the case that `Decision` (Ω_u^0) returns a factor of Ω_u^0. Then, `Main` (Ω_u^0) returns an optimal factor F of Ω_u^0. After the loop (lines 13 to 17) in Algorithm 1, we get a factor F^{opt} of Ω such that $\omega(F^{\text{opt}}) \geq \text{Opt}(\Omega_W^F)$ for every $u \in W \subseteq T_\Omega$ where $|W| = 1$ (when $u = v$) or $|W| = 2$ (when $u \neq v$) and $\omega(F^{\text{opt}}) \geq \omega(F)$. By Theorem 5, F^{opt} is an optimal factor of Ω. Thus, Algorithm 1 returns a correct result. ◀

Now, we consider the time complexity of Algorithm 1. The size of an instance is defined to be the number of vertices of the underlying graph of the instance.

▶ **Lemma 8.** *Run Algorithm 1 on an instance $\Omega = (G, \pi, \omega)$ of size n. Then,*

- *the algorithm will stop the recursion after at most n recursive steps;*
- *the algorithm will call* `Decision` *at most n many times, call* `Optimization` *at most $\frac{n(n+1)}{2} + 1$ many times, and perform at most $\frac{n(n+1)}{2}$ many comparisons;*
- *the algorithm runs in time $O(n^6)$.*

Proof. Let $\Omega^k = (G, \pi^k, \omega)$ be the instance after k many recursive steps. Here $\Omega^0 = \Omega$. Recall that $T_{\Omega^k} = \{v \in V \mid \pi^k(v) \in \mathcal{T}\}$. For an instance Ω^k with $|T_{\Omega^k}| > 0$, the recursive step will then go to the instance $(\Omega^k)_u^0$ or $(\Omega^k)_u^1$ for some $u \in T_{\Omega^k}$. Thus, $\Omega^{k+1} = (\Omega^k)_u^0$ or $(\Omega^k)_u^1$. In both cases, $T_{\Omega^{k+1}} = T_{\Omega^k} \backslash \{u\}$ and hence $|T_{\Omega^{k+1}}| = |T_{\Omega^k}| - 1$. By design, the algorithm will stop the recursion and return `Optimization` (Ω^m) when it reaches an instance Ω^m with $|T_{\Omega^m}| = 0$. Thus, #recursive steps $= m = |T_\Omega| - 0 \leq |V| = n$. To prove the second item, we consider the number of operations inside the recursive step for the instance $\Omega^k = (G, \pi^k, \omega)$. Note that $k \leq n$ and $|T_{\Omega^k}| = |T_\Omega| - k \leq n - k$. If $|T_{\Omega^k}| = 0$, then the algorithm will simply call `Optimization` once. If $|T_{\Omega^k}| > 0$, then inside the recursive step, the algorithm will call `Decision` once, and call `Optimization` once or $|T_{\Omega^k}|$ many times depending on the answer of `Decision`. Moreover, in the later case, the algorithm will also perform $|T_{\Omega^k}|$ many comparisons. Thus, we have #calls of `Decision` $= \sum_{|T_{\Omega^k}|>0} 1 = \sum_{i=1}^{|T_\Omega|} 1 = |T_\Omega| \leq n$, #calls of `Optimization` $\leq 1+\sum_{|T_{\Omega^k}|>0} |T_{\Omega^k}| = 1+\sum_{i=1}^{|T_\Omega|} i \leq \frac{n(n+1)}{2}+1$, and #comparisons $\leq \sum_{|T_{\Omega^k}|>0} |T_{\Omega^k}| \leq \frac{n(n+1)}{2}$. Let $t_{\texttt{Main}}(n)$ denote the running time of Algorithm 1 on an instance of size n, and $t_{\texttt{Dec}}(n)$ and $t_{\texttt{Opt}}(n)$ denote the running time of algorithms for functions `Decision` and `Optimization`, respectively. Then, $t_{\texttt{Dec}}(n) = O(n^4)$ by the algorithm in [7] and $t_{\texttt{Opt}}(n) = O(n^4)$ by the algorithm in [11]. Thus, $t_{\texttt{Main}}(n) \leq nt_{\texttt{Dec}}(n) + \frac{n(n+1)+2}{2} t_{\texttt{Opt}}(n) + \frac{n(n+1)}{2} = O(n^6)$. ◀

4 Proof of Theorem 5

In this section, we give a proof of Theorem 5. The general strategy is that starting with a non-optimal factor F of an instance $\Omega = (G, \omega, \pi)$, we want to find a subgraph H of G such that by taking the symmetric difference $F\Delta H$, we get another factor of Ω with larger weight. The existence of such subgraphs is trivial (Lemma 10). However, the challenge is how to find one efficiently. As an analogy of augmenting paths in the weighted matching problem, we introduce basic augmenting subgraphs (Definition 11) for the weighted graph factor problem, which can be found efficiently. We will show that given a non-optimal factor F, a basic augmenting subgraph always exists (Lemma 12, property 1). Then, we can efficiently improve the factor F to another factor with larger weight. As shown in [12], this already gave a weakly polynomial-time algorithm. However, the existence of basic augmenting subgraphs is not enough to get a strongly polynomial-time algorithm, which requires the number of improvement steps being independent of edge weights. Thus, in order to prove Theorem 5, which leads to a strongly polynomial-time algorithm, we further establish that there exists a basic augmenting subgraph that satisfies certain stronger properties under suitable assumptions (Lemma 12, property 2). This result will imply Theorem 5.

▶ **Definition 9** (F-augmenting subgraphs)**.** *Suppose that F is a factor of an instance $\Omega = (G, \pi, \omega)$. A subgraph H of G is F-augmenting if $F\Delta H \in \Omega$ and $\omega(F\Delta H) - \omega(F) > 0$.*

▶ **Lemma 10.** *Suppose that F is a factor of an instance Ω. If F is not optimal in Ω, then there exists an F-augmenting subgraph.*

Proof. Since F is not optimal, there is some $F' \in \Omega$ such that $\omega(F') > \omega(F)$. Let $H = F\Delta F'$. We have $F\Delta H = F' \in \Omega$ and $\omega(H) = \omega(F') - \omega(F) > 0$. Thus, H is F-augmenting. ◀

Recall that for an instance $\Omega = (G, \pi, \omega)$ of WGFP$(\mathcal{G}, \mathcal{T})$, T_Ω is the set $\{v \in V_G \mid \pi(v) \in \mathcal{T}\}$. For two factors $F, F^* \in \Omega$, we define $T_\Omega^{F\Delta F^*} = \{v \in T_\Omega \mid \deg_{F\Delta F^*}(v) \equiv 1 \mod 2\}$.

▶ **Definition 11** (Basic augmenting subgraphs)**.** *Suppose that F and F^* are factors of an instance $\Omega = (G, \pi, \omega)$ and $\omega(F) < \omega(F^*)$. An F-augmenting subgraph $H = (V_H, E_H)$ is (F, F^*)-*basic *if $H \subseteq F\Delta F^*$, $|V_H^{\text{odd}}| \leq 2$, and $V_H^{\text{odd}} \cap T_\Omega \subseteq T_\Omega^{F\Delta F^*}$ where $V_H^{\text{odd}} = \{v \in V_H \mid \deg_H(v) \equiv 1 \mod 2\}$.*

▶ **Lemma 12.** *Suppose that F and F^* are two factors of an instance $\Omega = (G, \pi, \omega)$.*

1. *If $\omega(F^*) > \omega(F)$, then there exists an (F, F^*)-basic subgraph.*
2. *If $\omega(F^*) > \text{Opt}(\Omega_W^F)$ for every $W \subseteq T_\Omega^{F\Delta F^*}$ with $|W| \leq 2$, and $T_\Omega^{F\Delta F^*}$ contains a vertex u such that $F \in \Omega_u^0$ (i.e., $\deg_F(u) \in D_u^0$), then there exists an (F, F^*)-basic subgraph H where $\deg_H(u) \equiv 0 \mod 2$.*

▶ Remark 13. The first property of Lemma 12 implies the following: a factor $F \in \Omega$ is optimal if and only if $\omega(F) \geq \mathrm{Opt}(\Omega_W^F)$ for every $W \subseteq T_\Omega$ with $|W| \leq 2$. This is a special case of the main result (Theorem 2) of [12] where the authors consider the WGFP for all constraints with gaps of length at most 1. The second property of Lemma 12 is more refined than the first property and it implies our main result (Theorem 5). In this paper, as a by-product of the proof of property 2, we give a simple proof of Theorem 2 of [12] for the special case WGFP($\mathcal{G} \cup \mathcal{T}$) based on certain properties of subcubic graphs.

Using the second property of Lemma 12, we can prove Theorem 5.

▶ **Theorem** (Theorem 5). *Suppose that F is a factor of an instance $\Omega = (G, \pi, \omega)$, and F is optimal in Ω_u^0 for some $u \in T_\Omega$. Then a factor F' is optimal in Ω if and only if $\omega(F') \geq \omega(F)$ and $\omega(F') \geq \mathrm{Opt}(\Omega_W^F)$ for every W where $u \in W \subseteq T_\Omega$ and $|W| = 1$ or 2.*

Proof. If F' is optimal in Ω, then clearly $\omega(F') \geq \omega(F)$ and $\omega(F') \geq \mathrm{Opt}(\Omega_W^F)$ for every W where $u \in W \subseteq T_\Omega$ and $|W| = 1$ or 2. Thus, to prove the theorem, it suffices to prove the other direction. Since $\omega(F') \geq \omega(F)$ and F is optimal in Ω_u^0, we have $\omega(F') \geq \mathrm{Opt}(\Omega_W^F)$ for every $W \subseteq T_\Omega$ where $u \notin W$ and $|W| \leq 2$. Also, since $\omega(F') \geq \mathrm{Opt}(\Omega_W^F)$ for every W where $u \in W \subseteq T_\Omega$ and $|W| = 1$ or 2, we have $\omega(F') \geq \mathrm{Opt}(\Omega_W^F)$ for every $W \subseteq T_\Omega$ where $|W| \leq 2$. For a contradiction, suppose that F' is not optimal in Ω. Let F^* be an optimal factor of Ω. Then, $\omega(F^*) > \omega(F')$. Thus, $\omega(F^*) > \omega(F') \geq \mathrm{Opt}(\Omega_W^F)$ for every $W \subseteq T_\Omega$ where $|W| \leq 2$. Also, $F^* \notin \Omega_u^0$ since $\omega(F^*) > \omega(F)$ and F is optimal in Ω_u^0. Thus, $\deg_{F^*}(u) \not\equiv \deg_F(u) \mod 2$. Then, $T_\Omega^{F \Delta F^*}$ contains the vertex u such that $F \in \Omega_u^0$. By Lemma 12, there exists an (F, F^*)-basic subgraph H where $\deg_H(u) \equiv 0 \mod 2$. Let $F'' = F \Delta H$. Then $F'' \in \Omega$ and $\omega(F'') > \omega(F)$. Also, $F'' \in \Omega_u^0$ since $\deg_{F''}(u) \equiv \deg_F(u) \mod 2$. This is a contradiction with F being optimal in Ω_u^0. ◀

Now it suffices to prove Lemma 12. By a type of normalization maneuver, we can transfer any instance of WGFP($\mathcal{G}, \mathcal{T}$) to an instance of WGFP($\mathcal{G}, \mathcal{T}$) defined on subcubic graphs, called a key instance (Definition 14). Recall that a subcubic graph is a graph where every vertex has degree 1, 2 or 3. For key instances, there are five possible forms of basic augmenting subgraphs, called basic factors (Definition 15). Then, the crux of the proof of Lemma 12 is to establish the existence of certain basic factors of key instances (Theorem 16). For a proof of Lemma 12 using Theorem 16, please refer to the proof of Lemma 4.4 in the full paper.

▶ **Definition 14** (Key instance). *A* key instance $\Omega = (G, \pi, \omega)$ *is an instance of* WGFP($\mathcal{G}, \mathcal{T}$) *where G is a subcubic graph, and for every $v \in V_G$, $\pi(v) = \{0, 1\}$ if $\deg_G(v) = 1$, $\pi(v) = \{0, 2\}$ if $\deg_G(v) = 2$, and $\pi(v) = \{0, 1, 3\}$ (i.e., type-1) or $\{0, 2, 3\}$ (i.e., type-2) if $\deg_G(v) = 3$. We say a vertex v of degree 3 is of type-1 or type-2 if $\pi(v)$ is type-1 or type-2 respectively. We say a vertex v of any degree is 1-feasible or 2-feasible if $1 \in \pi(v)$ or $2 \in \pi(v)$ respectively.*

▶ **Definition 15** (Basic factor). *Let Ω be a key instance. A factor of Ω is a* basic factor *if it is in one of the following five forms: a* path, *a* cycle, *a* tadpole graph *(i.e., a graph consisting of a cycle and a path such that they intersect at one endpoint of the path), a* dumbbell graph *(i.e., a graph consisting of two vertex disjoint cycles and a path such that the path intersects with each cycle at one of its endpoints), and a* theta graph *(i.e., a graph consisting of three vertex disjoint paths with the same two endpoints).*

▶ **Theorem 16.** *Suppose that $\Omega = (G, \pi, \omega)$ is a key instance.*
1. *If $\omega(G) > 0$, then there is a basic factor F of Ω such that $\omega(F) > 0$.*
2. *If $\omega(G) > 0$, $\omega(G) > \omega(F)$ for every basic factor F of Ω, and G contains a vertex u with $\deg_G(u) = 1$ or $\deg_G(u) = 3$ and $\pi(u) = \{0, 2, 3\}$, then there is a basic factor F^* of Ω such that $\omega(F^*) > 0$ and $\deg_{F^*}(u) \equiv 0 \mod 2$. (Recall that $\deg_{F^*}(u) = 0$ if $u \notin V_{F^*}$.)*

▶ Remark 17. For the second property of Theorem 16, the requirement of $\pi(u) = \{0, 2, 3\}$ when $\deg_G(u) = 3$ is crucial. Consider the instance $\Omega = (G, \pi, \omega)$ as shown in Figure 1. Note that Ω is a key instance. and $\pi(u) = \{0, 1, 3\}$. In this case where $\pi(u) = \{0, 1, 3\}$, it can be checked that the second property does *not* hold.

5 Proof Sketch of Theorem 16

In this section, we give a proof sketch of Theorem 16 and we focus on the proof of the second property using the first property. Omitted proofs can be found in Section 5 of the full paper.

Proof sketch. By property 1 of Theorem 16, there exists at least one basic factor of Ω such that its weight is positive. Among all such basic factors, we pick an F such that $\omega(F)$ is the largest. Consider the graph $G' = G \backslash F$, i.e., the subgraph of G induced by the edge set $E_G \backslash E_F$. We consider the instance $\Omega' = (G', \pi', \omega')$ where for every $x \in V_{G'}$, $\pi'(x) = \{0, 1\}$ if $\deg_{G'}(x) = 1$, $\pi'(x) = \{0, 2\}$ if $\deg_{G'}(x) = 2$ and $\pi'(x) = \pi(x)$ if $\deg_{G'}(x) = 3$, and ω' is the weight function ω restricted to G'. Note that Ω' is also a key instance, but it is not necessarily a sub-instance of Ω. Since $\omega(G) > \omega(F)$, we have $\omega'(G') = \omega(G') = \omega(G) - \omega(F) > 0$. Without causing ambiguity, we may simply write ω' as ω in the instance Ω'. By property 1 of Theorem 16, there exists a basic factor F' of Ω' such that $\omega(F') > 0$. Since $E_{F'} \subseteq E_G \backslash E_F$, F and F' are edge-disjoint. Let $H = F \cup F'$, which is the subgraph of G induced by the edge set $E_F \cup E_{F'}$. We will show that we can find a subgraph F^* of H such that F^* is the desired basic factor of Ω satisfying $\omega(F^*) > 0$ and $\deg_{F^*}(u) \equiv 0 \mod 2$.

First, we show that H is a factor of Ω. Let $V_\cap = V_F \cap V_{F'}$. We show that for every $x \in V_H \backslash V_\cap$, $\deg_H(x) \in \pi(x)$. If $x \in V_F \backslash V_\cap$, then $\deg_H(x) = \deg_F(x)$. Since $F \in \Omega$, $\deg_F(x) \in \pi(x)$. Then, $\deg_H(x) \in \pi(x)$. If $x \in V_{F'} \backslash V_\cap$, then $\deg_H(x) = \deg_{F'}(x)$. Since $x \notin V_F$ and $G' = G \backslash F$, $\deg_{G'}(x) = \deg_G(x)$. Then, by the definition of Ω', we have $\pi'(x) = \pi(x)$. Since F' is a factor of Ω', $\deg_{F'}(x) \in \pi'(x)$. Thus, $\deg_H(x) \in \pi(x)$. Now, we consider vertices in $V_\cap$. Since F and F' are edge disjoint, for every $x \in V_\cap$ we have $\deg_H(x) = \deg_F(x) + \deg_{F'}(x) \leq \deg_G(x) \leq 3$. Also, $\deg_F(x), \deg_{F'}(x) \geq 1$ since F and F' are subcubic graphs which have no isolated vertices.

- If $\deg_F(x) = 1$, then $1 \in \pi(x)$. The vertex x is 1-feasible. Thus, $\deg_G(x) \neq 2$. Since $\deg_G(x) > \deg_F(x) = 1$, $\deg_G(x) = 3$. Then, $\deg_{G'}(x) = \deg_G(x) - \deg_F(x) = 2$, $\pi'(x) = \{0, 2\}$ and $\deg_{F'}(x) = 2$.
- If $\deg_F(x) = 2$, then $\deg_G(x) = 3$ since $\deg_G(x) > \deg_F(x)$. Then, $\deg_{G'}(x) = \deg_G(x) - \deg_F(x) = 1$, $\pi'(x) = \{0, 1\}$ and $\deg_{F'}(x) = 1$.

Thus, for every $x \in V_\cap$, $\deg_H(x) = \deg_F(x) + \deg_{F'}(x) = 3 \in \pi(x)$. Thus, H is a factor of Ω.

Then, we finish the proof by a careful analysis of possible forms of F and F', and possible intersection vertices in $V_\cap$. Here, we give an example where F is a tadpole graph with a vertex u of degree 3 and a vertex v of degree 1 to illustrate this. Since $\deg_F(u) = 3$, by assumption, $\pi(u) = \{0, 2, 3\}$. Also, since $\deg_F(v) = 1 \in \pi(v)$, v is 1-feasible.

Consider possible vertices in $V_\cap$. Recall that for every $x \in V_\cap$, $\deg_F(x) = 1$ and $\deg_{F'}(x) = 2$, or $\deg_F(x) = 2$ and $\deg_{F'}(x) = 1$. Since $\deg_F(u) = 3 = \deg_G(u)$, we have $u \notin V_\cap$. Also, consider the possible forms of F'. We show that F' is not a cycle. For a contradiction, suppose that F' is a cycle. Then, all vertices of F' have degree 2. Thus, the only possible vertex in $V_\cap$ is v. If $V_\cap = \emptyset$, then for every $x \in V_{F'}$, $\deg_{F'}(x) = \deg_H(x) \in \pi(x)$. Thus, F' is a basic factor of Ω where $\omega(F') > 0$ and $\deg_{F'}(u) = 0$. We are done. Otherwise, $V_\cap = \{v\}$. Then, $\deg_F(v) = 1$ and $\deg_{F'}(v) = 2$. Since F is a tadpole graph, the graph H is a dumbbell graph where u and v are the two vertices of degree 3. Thus, H is a basic factor of Ω. Since $\omega(F') > 0$, we have $\omega(H) = \omega(F) + \omega(F') > \omega(F)$ which leads to a contraction with F being a basic factor with the largest weight. Thus, F' is a basic factor which is not a cycle. By Definition 15, F' contains exactly two vertices of odd degree, denoted by s and t. Then, we have $V_\cap \subseteq \{v, s, t\}$.

Recall that F is a tadpole graph consisting of a path and a cycle. We use C to denote the cycle part of F, and V_C denotes its vertex set. Consider $\{s,t\} \cap V_C$. Now, we handle possible subcases according to intersection vertices appearing in V_C. There are three subcases. Below, for two points x and y, we use p_{xy} or p'_{xy} to denote a path with endpoints x and y.

1. $\{s,t\} \subseteq V_C$. Then, $\deg_F(s) = \deg_F(t) = \deg_C(s) = \deg_C(t) = 2$. In this case, $\deg_H(u) = \deg_H(s) = \deg_H(t) = 3$ and $\pi(u) = \pi(s) = \pi(t) = \{0,2,3\}$. Also, $\deg_{F'}(s) = \deg_{F'}(t) = 1$. Thus, F' is a path with endpoints s and t. Note that in this case, it is possible that $v \in V_{F'}$. If $v \in V_{F'}$, then $\deg_H(v) = 3$ and $\pi(v) = \{0,1,3\}$; otherwise, $\deg_H(v) = 1$ and $\pi(v) = \{0,1\}$ or $\{0,1,3\}$. The points u, s, and t split C into three paths, p_{us}, p_{st}, p_{tu}. Then, $C = p_{us} \cup p_{st} \cup p_{tu}$. (See Figure 2.) If $\omega(C) > 0$, then we are done since C is a basic factor of Ω and $\deg_C(u) = 2$. Thus, we may assume that $\omega(C) \leq 0$.

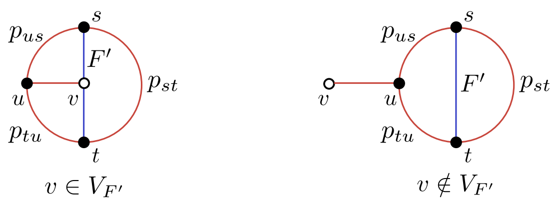

Figure 2 The two possible forms of graph H when $\{s,t\} \in V_C$. Hollow nodes denote 1-feasible vertices, and solid nodes denote 2-feasible vertices; red-colored lines denote paths in F, and blue-colored lines denote paths in F'.

Consider the graph $H_1 = H \backslash p_{st} = (F \backslash p_{st}) \cup F'$. Note that $V_{H_1} = (V_H \backslash V_{p_{st}}) \cup \{s,t\}$. For every $x \in V_{H_1} \backslash \{s,t\}$, we have $\deg_{H_1}(x) = \deg_H(x) \in \pi(x)$ since H is a factor of Ω. Also, $\deg_{H_1}(s) = 2 \in \pi(s)$ and $\deg_{H_1}(t) = 2 \in \pi(t)$. Thus, H_1 is a factor of Ω. Also, H_1 is a tadpole graph if $\deg_H(v) = 1$ or a theta graph if $\deg_H(v) = 3$. Thus, in both cases, H_1 is a *basic* factor of Ω. Since F is a basic factor of Ω with the largest weight, we have $\omega(F) \geq \omega(H_1) = \omega(F) - \omega(p_{st}) + \omega(F')$. Thus, $\omega(p_{st}) \geq \omega(F') > 0$. Since $\omega(C) = \omega(p_{st}) + \omega(p_{us}) + \omega(p_{tu}) \leq 0$, $\omega(p_{us}) + \omega(p_{tu}) < 0$. Without loss of generality, we may assume that $\omega(p_{us}) < 0$. Then, consider the graph $H_2 = H \backslash p_{us}$. Similarly, one can check that H_2 is a factor of Ω, and $\deg_{H_2}(u) = 2$. Also, H_2 is a tadpole graph if $\deg_H(v) = 1$, or a theta graph if $\deg_H(v) = 3$. Thus, H_2 is a basic factor of Ω. Moreover, $\omega(H_2) = \omega(H) - \omega(p_{us}) > 0$. We are done.

2. $\{s,t\} \cap V_C = \{s\}$ or $\{t\}$. Without loss of generality, we may assume that $s \in V_C$. Then, $\deg_H(u) = \deg_H(s) = 3$ and $\pi(u) = \pi(s) = \{0,2,3\}$. If $\omega(C) > 0$, then we are done since C is a basic factor of Ω and $\deg_C(u) = 2$. Thus, we may assume that $\omega(C) \leq 0$. Vertices s and u split C into two paths p_{us} and p'_{us}. Since $\omega(C) = \omega(p_{us}) + \omega(p'_{us}) \leq 0$, among them at least one is non-positive. Without loss of generality, we assume that $\omega(p_{us}) \leq 0$. Consider the graph $H' = H \backslash p_{us}$. We have $\omega(H') = \omega(H) - \omega(p_{us}) > 0$, and $\deg_{H'}(u) = 2$. Similar to the above case, one can check that H' is a factor of Ω. However, it is not clear whether H' is a *basic* factor of Ω. Consider the sub-instance $\Omega'_H = (H', \pi_{H'}, \omega_{H'})$ of Ω defined on the subgraph H' of G where $\pi_{H'}(x) = \pi(x) \cap [\deg_{H'}(x)] \subseteq \pi(x)$ for every $x \in V_{H'}$ and $\omega_{H'}$ is the restriction of ω on $E_{H'}$ (we may write $\omega_{H'}$ as ω for simplicity). Since $\omega(H') > 0$, by property 1 of Theorem 16, there is a basic factor $F^* \in \Omega_{H'}$ such that $\omega(F^*) > 0$. Then, $\deg_{F^*}(u) \in \pi_{H'}(u) = \{0,2\}$. Now, F^* is a basic factor of Ω.

3. $\{s,t\} \cap V_C = \emptyset$. In this case, the cycle C does not intersect with F'. Then, by viewing the cycle C as an enlargement of the vertex u, this case is similar to the case that F is a path with endpoints u and v, which is proved separately. Please refer to the full paper for its proof. ◀

References

1 Jin Akiyama and Mikio Kano. *Factors and factorizations of graphs: Proof techniques in factor theory*, volume 2031. Springer, 2011.

2 Elliot Anshelevich and Adriana Karagiozova. Terminal backup, 3d matching, and covering cubic graphs. *SIAM Journal on Computing*, 40(3):678–708, 2011.

3 Richard P. Anstee. A polynomial algorithm for b-matchings: an alternative approach. *Information Processing Letters*, 24(3):153–157, 1987.

4 André Bouchet. Greedy algorithm and symmetric matroids. *Mathematical Programming*, 38(2):147–159, 1987. doi:10.1007/BF02604639.

5 André Bouchet. Matchings and Δ-matroids. *Discrete Applied Mathematics*, 24(1-3):55–62, 1989.

6 David A Cohen, Martin C Cooper, Peter G Jeavons, and Andrei A Krokhin. The complexity of soft constraint satisfaction. *Artificial Intelligence*, 170(11):983–1016, 2006.

7 Gérard Cornuéjols. General factors of graphs. *Journal of Combinatorial Theory, Series B*, 45(2):185–198, 1988. doi:10.1016/0095-8956(88)90068-8.

8 William H. Cunningham and Alfred B. Marsh. A primal algorithm for optimum matching. In *Polyhedral Combinatorics*, pages 50–72. Springer, 1978.

9 Víctor Dalmau and Daniel K. Ford. Generalized satisfability with limited occurrences per variable: A study through delta-matroid parity. In *Proceedings of the 28th International Symposium on Mathematical Foundations of Computer Science (MFCS'03)*, volume 2747, pages 358–367. Springer, 2003. doi:10.1007/978-3-540-45138-9_30.

10 Ran Duan and Seth Pettie. Linear-time approximation for maximum weight matching. *Journal of the ACM*, 61(1):1–23, 2014. doi:10.1145/2529989.

11 Szymon Dudycz and Katarzyna Paluch. Optimal general matchings. *Lecture Notes in Computer Science*, 11159 LNCS:176–189, 2018. doi:10.1007/978-3-030-00256-5_15.

12 Szymon Dudycz and Katarzyna Paluch. Optimal general matchings. *arXiv, version 3*, 2021. arXiv:1706.07418v3.

13 Zdeněk Dvořák and Martin Kupec. On Planar Boolean CSP. In *Proceedings of the 42nd International Colloquium on Automata, Languages and Programming (ICALP'15)*, volume 9134, pages 432–443. Springer, 2015. doi:10.1007/978-3-662-47672-735.

14 Jack Edmonds. Maximum matching and a polyhedron with 0, 1-vertices. *Journal of research of the National Bureau of Standards B*, 69(125-130):55–56, 1965.

15 Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of mathematics*, 17:449–467, 1965. doi:10.4153/CJM-1965-045-4.

16 Jack Edmonds and Ellis L Johnson. Matching: A well-solved class of integer linear programs. In *Combinatorial Structures and Their Applications*, pages 89–92. Gordon & Breach, New York, 1970.

17 Tomás Feder. Fanout limitations on constraint systems. *Theoretical Computer Science*, 255(1-2):281–293, 2001. doi:10.1016/S0304-3975(99)00288-1.

18 Tomás Feder and Daniel K. Ford. Classification of bipartite Boolean constraint satisfaction through Delta-matroid intersection. *SIAM J. Discrete Math.*, 20(2):372–394, 2006. doi:10.1137/S0895480104445009.

19 Tomás Feder and Moshe Y. Vardi. The Computational Structure of Monotone Monadic SNP and Constraint Satisfaction: A Study through Datalog and Group Theory. *SIAM Journal on Computing*, 28(1):57–104, 1998. doi:10.1137/S0097539794266766.

20 Harold N. Gabow. *Implementation of algorithms for maximum matching on nonbipartite graphs*. PhD thesis, Stanford University, 1974.

21 Harold N. Gabow. An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems. In *Proceedings of the fifteenth annual ACM symposium on Theory of computing*, pages 448–456, 1983.

22 Harold N. Gabow. A scaling algorithm for weighted matching on general graphs. In *Proceedings of the 26th Annual IEEE Symposium on Foundations of Computer Science (FOCS'85)*, pages 90–100, 1985.

23 Harold N. Gabow. Data structures for weighted matching and nearest common ancestors with linking. In *Proceedings of the first annual ACM-SIAM symposium on Discrete algorithms*, pages 434–443, 1990.

24 Harold N. Gabow, Zvi Galil, and Thomas H. Spencer. Efficient implementation of graph algorithms using contraction. *Journal of the ACM (JACM)*, 36(3):540–572, 1989.

25 Harold N. Gabow and Piotr Sankowski. Algebraic algorithms for b-matching, shortest undirected paths, and f-factors. In *Proceedings of the 54th IEEE Annual Symposium on Foundations of Computer Science (FOCS'13)*, pages 137–146, 2013.

26 Harold N. Gabow and Robert E. Tarjan. Faster scaling algorithms for general graph matching problems. *Journal of the ACM (JACM)*, 38(4):815–853, 1991.

27 Zvi Galil, Silvio Micali, and Harold N. Gabow. An $O(EV \setminus \log V)$ algorithm for finding a maximal weighted matching in general graphs. *SIAM Journal on Computing*, 15(1):120–130, 1986.

28 James F. Geelen, Satoru Iwata, and Kazuo Murota. The linear delta-matroid parity problem. *Journal of Combinatorial Theory, Series B*, 88(2):377–398, 2003. doi:10.1016/S0095-8956(03)00039-X.

29 Chien-Chung Huang and Telikepalli Kavitha. Efficient algorithms for maximum weight matchings in general graphs with small edge weights. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1400–1412. SIAM, 2012.

30 Gabriel Istrate. Looking for a version of Schaefer's dichotomy theorem when each variable occurs at most twice. Technical report, University of Rochester, 1997. UR CSD/TR652.

31 Aleksandr V. Karzanov. Efficient implementations of Edmonds' algorithms for finding matchings with maximum cardinality and maximum weight. *Studies in Discrete Optimization*, pages 306–327, 1976.

32 Alexandr Kazda, Vladimir Kolmogorov, and Michal Rolínek. Even delta-matroids and the complexity of planar Boolean CSPs. *ACM Transactions on Algorithms (TALG)*, 15(2):1–33, 2018.

33 Yusuke Kobayashi. Optimal general factor problem and jump system intersection. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 291–305. Springer, 2023.

34 Bernhard Korte and Jens Vygen. *Combinatorial optimization: Theory and Algorithms*, volume 21. Springer, 2018.

35 Eugene L. Lawler. *Combinatorial optimization: networks and matroids*. Holt, Reinhart and Winston, New York., 1976.

36 László Lovász. The factorization of graphs. In *Combinatorial Structures and Their Applications*, pages 243–246. Gordon & Breach, New York, 1970.

37 László Lovász. The factorization of graphs. II. *Acta Mathematica Academiae Scientiarum Hungarica*, 23(1-2):223–246, 1972. doi:10.1007/BF01889919.

38 László Lovász and Michael D. Plummer. *Matching theory*, volume 367. American Mathematical Soc., 2009.

39 Alfred B. Marsh. *Matching algorithms*. PhD thesis, The Johns Hopkins University, 1979.

40 Michael D. Plummer. Graph factors and factorization: 1985–2003: a survey. *Discrete Mathematics*, 307(7-8):791–821, 2007.

41 William R. Pulleyblank. *Faces of Matching Polyhedra*. PhD thesis, University of Waterloo, 1973.

42 Thomas J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the tenth annual ACM symposium on Theory of computing*, pages 216–226, 1978.

43 Alexander Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer, 2003.

44 Shuai Shao and Stanislav Živný. A Strongly Polynomial-Time Algorithm for Weighted General Factors with Three Feasible Degrees. *arXiv*, 2023. `arXiv:2301.11761`.

45 Jácint Szabó. Good characterizations for some degree constrained subgraphs. *Journal of Combinatorial Theory, Series B*, 99(2):436–446, 2009. `doi:10.1016/J.JCTB.2008.08.009`.

46 William Thomas Tutte. The factorization of locally finite graphs. *Canadian Journal of Mathematics*, 2:44–49, 1950.

47 William Thomas Tutte. The factors of graphs. *Canadian Journal of Mathematics*, 4:314–328, 1952.

48 William Thomas Tutte. A short proof of the factor theorem for finite graphs. *Canadian Journal of mathematics*, 6:347–352, 1954.

49 Dahai Xu, Elliot Anshelevich, and Mung Chiang. On survivable access network design: Complexity and algorithms. In *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*, pages 186–190. IEEE, 2008.

A Δ-Matroids and Matching Realizability

A Δ-matroid is a family of sets obeying an axiom generalizing the matroid exchange axiom. Formally, a pair $M = (U, \mathcal{F})$ is a Δ-matroid if U is a finite set and $\mathcal{F}$ is a collection of subsets of U satisfying the following: for any $X, Y \in \mathcal{F}$ and any $u \in X\Delta Y$ in the symmetric difference of X and Y, there exits a $v \in X\Delta Y$ such that $X\Delta\{u, v\}$ belongs to $\mathcal{F}$ [4]. A Δ-matroid is *symmetric* if, for every pair of $X, Y \subseteq U$ with $|X| = |Y|$, we have $X \in \mathcal{F}$ if and only if $Y \in \mathcal{F}$. A Δ-matroid is *even* if for every pair of $X, Y \subseteq U$, $|X| \equiv |Y| \mod 2$.

Suppose that $U = \{u_1, u_2, \ldots, u_n\}$. A subset $V \subseteq U$ can be encoded by a binary string α_V of n-bits where the i-th bit of α_V is 1 if $u_i \in V$ and 0 if $u_i \notin V$. Then, a Δ-matroid $M = (U, \mathcal{F})$ can be represented by a relation R_M of arity $|U|$ which consists of binary strings that encode all subsets in $\mathcal{F}$. Such a representation is unique up to a permutation of variables of the relation. A degree constraint D of arity n can be viewed as an n-ary symmetric relation which consists of binary strings with the Hamming weight d for every $d \in D$. By the definition of Δ-matroids, it is easy to check that a degree constraint D (as a symmetric relation) represents a Δ-matroid if and only if D has all gaps of length at most 1.

▶ **Definition 18** (Matching Gadget). *A* gadget *using a set $\mathcal{D}$ of degree constraints consists of a graph $G = (U \cup V, E)$ where $\deg_G(u) = 1$ for every $u \in U$ and there are no edges between vertices in U, and a mapping $\pi : V \to \mathcal{D}$. A* matching gadget *is a gadget where $\mathcal{D} = \{\{0, 1\}, \{1\}\}$. A degree constraint D of arity n is* matching realizable *if there exists a matching gadget $(G = (U \cup V, E), \pi : V \to \{\{0, 1\}, \{1\}\})$ such that $|U| = n$ and for every $k \in [n]$, $k \in D$ if and only if for every $W \subseteq U$ with $|W| = k$, there exists a matching $F = (V_F, E_F)$ of G such that $V_F \cap U = W$ and for every $v \in V$ where $\pi(v) = \{1\}$, $v \in V_F$.*

The definition of matching realizability can be extended to a relation R of arity n by requiring the set U of n vertices in a matching gadget to represent the n variables of R. If R is realizable by a matching gadget $G = (U \cup V, E)$, then for every $\alpha \in \{0, 1\}^n$, $\alpha \in R$ if and only if there is a matching $F = (V_F, E_F)$ of G such that $V_F \cap U$ is exactly the subset of U encoded by α (i.e., for every $u_i \in U$, $u_i \in V_F$ if and only if $\alpha_i = 1$), and for every $v \in V$

where $\pi(v) = \{1\}$, $v \in V_F$. Note that the matching realizability of a relation is invariant under a permutation of its variables. We say that a Δ-matroid is matching realizable if the relation representing it is matching realizable.[6]

The following result generalizes Lemma A.1 of [32].

▶ **Lemma 19.** *Suppose that $M = (U, \mathcal{F})$ is a matching realizable Δ-matroid, and $V_1, V_2 \in \mathcal{F}$. Then, $V_1 \Delta V_2$ can be partitioned into single variables $S_1, \ldots, S_k$ and pairs of variables $P_1, \ldots, P_\ell$ such that for every $P = S_{i_1} \cup \cdots \cup S_{i_r} \cup P_{j_1} \cup \cdots \cup P_{j_t}$ ($\{i_1, \ldots, i_r\} \subseteq [k], \{j_1, \ldots, j_t\} \subseteq [\ell]$), $V_1 \Delta P \in \mathcal{F}$ and $V_2 \Delta P \in \mathcal{F}$.*

▶ **Theorem 20.** *A degree constraint D of gaps of length at most 1 is matching realizable if and only if all its gaps are of the same length 0 or 1.*

Proof. By the gadget constructed in the proof of [7, Theorem 2], if a degree constraint has all gaps of length 1 then it is matching realizable.[7] We give the following gadget (Figure 3) to realize a degree constraint D with all gaps of length 0, which generalizes the gadget in [48]. Suppose that $D = \{p, p+1, \ldots, p+r\}$ of arity n where $n \geq p+r \geq p \geq 0$. Consider the following graph $G = (U \cup V, E)$: U consists of n vertices of degree 1, and V consists of two parts V_1 with $|V_1| = n$ and V_2 with $|V_2| = n - p$; the induced subgraph $G(V)$ of G induced by V is a complete bipartite graph between V_1 and V_2, and the induced subgraph $G(U \cup V_1)$ of G induced by $U \cup V_1$ is a bipartite perfect matching between U and V_1. Every vertex in V_1 is labeled by the constraint $\{1\}$. There are r vertices in V_2 labeled by $\{0, 1\}$ and the other $n - p - r$ vertices in V_2 labeled by $\{1\}$. One can check that this gadget realizes D.

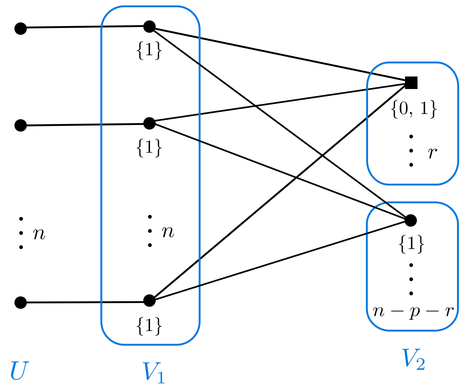

Figure 3 A matching gadget realizing $D = \{p, p+1, \ldots, p+r\}$ of arity n.

For the other direction, without loss of generality, we may assume that $\{p, p+1, p+3\} \subseteq D$ and $p+2 \notin D$. Since D has gaps of length at most 1, it can be associated with a symmetric Δ-matroid $M = (U, \mathcal{F})$. Then, there is $V_1 \in \mathcal{F}$ with $|V_1| = p$ and $V_2 \in \mathcal{F}$ with $|V_2| = p+3$. Since M is symmetric, we may pick $V_2 = V_1 \cup \{v_1, v_2, v_3\}$ for some $\{v_1, v_2, v_3\} \cap V_1 = \emptyset$. Let $S = V_1 \Delta V_2 = \{v_1, v_2, v_3\}$. By Lemma 19, S can be partitioned into single variables and/or pairs of variables such that for any union P of them, $V_2 \backslash P \in \mathcal{F}$. Since $|S| = 3$, there exists at least a single variable x_i in the partition of S such that $V_2 \backslash \{v_i\} \in \mathcal{F}$. Note that $|V_2 \backslash \{v_i\}| = p+2$. Thus, $p+2 \in D$. A contradiction. ◀

[6] This definition of matching realizability for Δ-matroids is different from the one that is usually used for even Δ-matroids [5, 13, 32], in which the gadget is only allowed to use the constraint $\{1\}$ for perfect matchings, and hence the resulting Δ-matroid must be even.

[7] We remark that [7] includes gadgets for other types of degree constraints, including type-1 and type-2, but only under a more general notion of gadget constructions that involve edges and triangles. The gadget that only involves edges is a matching gadget defined in this paper.