SIAM J. DISCRETE MATH.
Vol. 33, No. 2, pp. 795–836

EDITING TO CONNECTED F-DEGREE GRAPH*

FEDOR V. FOMIN†, PETR GOLOVACH†, FAHAD PANOLAN†, AND SAKET SAURABH‡

Abstract. In the EDGE EDITING TO CONNECTED f-DEGREE GRAPH problem we are given a graph G, an integer k, and a function f assigning integers to vertices of G. The task is to decide whether there is a connected graph F on the same vertex set as G, such that for every vertex v, its degree in F is $f(v)$, and the number of edges in $E(G)\triangle E(F)$, the symmetric difference of $E(G)$ and $E(F)$, is at most k. We show that EDGE EDITING TO CONNECTED f-DEGREE GRAPH is fixed-parameter tractable (FPT) by providing an algorithm solving the problem on an n-vertex graph in time $2^{\mathcal{O}(k)}n^{\mathcal{O}(1)}$. We complement this result by showing that the weighted version of the problem with costs 1 and 0 is W[1]-hard when parameterized by k and the maximum value of f even when the input graph is a tree. Our FPT algorithm is based on a nontrivial combination of color-coding and fast computations of representative families over the direct sum matroid of ℓ-elongation of the co-graphic matroid associated with G and a uniform matroid over the set of nonedges of G. We believe that this combination could be useful in designing parameterized algorithms for other edge editing and connectivity problems.

Key words. connected f-factor, graph editing problem, FPT, representative family, color coding, matroids

AMS subject classifications. 68Q25, 68W05, 05C85, 68W40

DOI. 10.1137/17M1129428

1. Introduction.

A subgraph F of a graph G is a factor of G if F is a spanning subgraph of G. When a factor F is described in terms of its degrees, it is called a *degree-factor*. For example, one of the most fundamental notions in Graph Theory is 1-factor (or a perfect matching), the case when a factor F has all of its degrees equal to 1. Another example is r-factor, a regular spanning subgraph of degree r. More generally, for a function $f\colon V(G) \to \mathbb{N}$, subgraph F is an f-factor of G if for every $v \in V(G)$, $d_F(v)$, the degree of v in F is exactly $f(v)$. The study of degree factors is one of the mainstays of combinatorics with a long history dating back to 1847 to the works of Kirkman [13], and Petersen [21]. We refer to surveys [1, 22], as well as the book of Lovász and Plummer [16], for an extensive overview of degree factors.

Another broad set of degree-factor problems is obtained by requesting the factor to be connected. The most famous examples are other classical Graph Theory notions, the Hamiltonian cycle, which is a connected 2-factor, and the Eulerian subgraph, which is a connected even-degree factor. We refer to the survey of Kouider and Vestergaard [14] on connected factors, as well as to the book of Fleischner [7] for a thorough study of Eulerian graphs and related topics.

A natural algorithmic problem concerning (connected) f-factors is for a given graph G and a function f to decide whether G contains a (connected) f-factor. While

*Received by the editors May 9, 2017; accepted for publication (in revised form) February 28, 2019; published electronically May 9, 2019. A preliminary version of this paper appeared in the proceedings of STACS 2016.

http://www.siam.org/journals/sidma/33-2/M112942.html

Funding: This work is supported by the European Research Council (ERC) via grants Rigorous Theory of Preprocessing, reference 267959, PARAPPROX, reference 306992, and by the Research Council of Norway via project MULTIVAL.

†Department of Informatics, University of Bergen, Bergen 5020, Norway (fomin@ii.uib.no, petr.golovach@ii.uib.no, fahad.panolan@ii.uib.no).

‡The Institute of Mathematical Sciences, HBNI, Chennai, Tami Nadu 600 113, India and University of Bergen, Bergen 5020, Norway (saket@imsc.res.in).

deciding whether a given graph contains an f-factor can be done in polynomial time for any function f [3], deciding the existence of even a connected 2-factor (Hamiltonian cycle) is NP-complete. In this work we study the parameterized complexity of the following algorithmic generalization of the problem of finding a connected f-factor.

Edge Editing to Connected f-Degree Graph (EECG)
Input: An undirected graph G, a function $f : V(G) \to \{1, 2, \ldots, d\}$, and $k \in \mathbb{N}$.
Parameter: k
Question: Does there exist a connected graph F such that for every vertex v, $d_F(v) = f(v)$, and the size of the symmetric difference is $|E(G) \triangle E(F)| \leq k$?

Apart from studying a classical problem algorithmically, one of the main motivations for our interest in the generalization of the classical f-factor problem comes from the recent developments in parameterized algorithms for graph modification problems. These recent algorithmic advances were important not only due to the problems they settled but also for the kind of techniques they brought to the area. For example, the work on cut (or edge-deletion) problems of Kawarabayashi and Thorup [12] and Chitnis et al. [4] brought recursive understanding and randomized contraction techniques. The work on Feedback Arc Set in Tournaments [2] led to the chromatic coding. Study of chordal graph completions from [9] triggered the usage of potential maximal cliques in subexponential parameterized algorithms.

The main result of our paper is the following theorem.

Theorem 1.1. EECG *is solvable in time* $2^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$ *deterministically.*

The proof of our theorem is based on (i) color-coding and (ii) fast computation of representative family over a linear matroid. While using graphic matroids to resolve different types of connectivity issues has become a popular theme in algorithms, our proof requires the usage of some nonstandard matroids. In particular, we use fast representative family computations over the direct sum matroid of ℓ-elongation of the co-graphic matroid associated with G and a uniform matroid over $\overline{E(G)}$, the set of non-edges of G. We believe that this combination could be useful for designing parameterized algorithms for other edge editing problems. To the best of our knowledge, this is the first use of elongation of matroids in the designing of parameterized algorithms.

One of the nice properties of using matroids is that often they allow us to lift solutions from unweighted problems to problems with costs. It would be natural to suggest that the nice properties of matroids would help us with the "weighted" version of EECG as well.

Edge Editing to Connected f-Degree Graph with Costs
Input: A graph G, functions $f : V(G) \to \{1, 2, \ldots, d\}$ and $c\colon \binom{V(G)}{2} \to \mathbb{N}$, and $k, C \in \mathbb{N}$.
Parameter: $k + d$
Question: Does there exist a connected graph F such that for every vertex v, $d_F(v) = f(v)$, $|E(G) \triangle E(F)| \leq k$, and $c(E(G) \triangle E(F)) \leq C$?

However, in spite of our attempts, we could not extend the results of Theorem 1.1 to Edge Editing to Connected f-Degree Graph with Costs. The following theorem explains why our initial attempts failed.

Theorem 1.2. Edge Editing to Connected f-Degree Graph with Costs *(parameterized by $k+d$) is W[1]-hard even when the input graph G is a tree and costs are restricted to be* 0 *or* 1*.*

Previous work. It was shown by Mathieson and Szeider that the problem of deleting k vertices to transform an input graph into an r-regular graph, where $r \geq 3$ is W[1]-hard parameterized by k [18]. For edge-modification problems to a graph with certain degrees, Mathieson and Szeider have shown that EDITING TO f-DEGREE GRAPH, the case when the requirement that the resulting graph F is connected is omitted, is solvable in polynomial time. As with the f-factor problem, the situation changes drastically when one adds the requirement that the resulting graph F is connected. A simple reduction from Hamiltonian cycle shows that in this case deciding if a graph can be edited into a connected 2-degree graph, i.e., a cycle, by changing at most k adjacencies, is NP-complete [10].

Golovach in [10] has shown that when parameterized by the maximum vertex degree d in the resulting graph plus the number of editing operations k, the problem EDGE EDITING TO CONNECTED f-DEGREE GRAPH is fixed-parameter tractable (FPT). In the same paper, it was shown that in the variant when the resulting graph F is regular, the problem is FPT, parameterized by k. However, prior to our work the complexity status of EDGE EDITING TO CONNECTED f-DEGREE GRAPH remained open. Thus, Theorem 1.1 resolves the problem in affirmative. However, we still do not know the kernelization status of this problem and leave it as an interesting open problem. A related problem, EULERIAN EDGE DELETION on an n-vertex graph is proved to be solvable in time $2^{\mathcal{O}(k)}n^{\mathcal{O}(1)}$ [11].

Our approach. Each solution to our problem is of the form $D \cup A$ where $D \subseteq E(G)$ and $A \subseteq \overline{E(G)}$. The sets D and A are called a deletion set and an addition set, respectively, corresponding to the solution $D \cup A$. We start by characterizing our solution in terms of special deletion set D, called nice deletion set. Nice deletion sets satisfy certain properties. These properties of nice deletion sets allow us to recover the addition set A in polynomial time. This viewpoint allows us to concentrate on finding a nice deletion set. To get a nice deletion set D, we view the solution $D \cup A$ as a system of "alternating walks and alternating even closed walks." Alternating (closed) walks are essentially normal (closed) walks with edges from $D \cup A$ such that they do not have consecutive edges from either D or A, and no edge from $D \cup A$ repeats in the walk. We take this viewpoint to construct a dynamic programming algorithm as this allows us to proceed between the states of the program by using one edge (either of an addition set or of a deletion set). The number of states can be upper bounded by $2^{\mathcal{O}(k)}$. However, the number of sets $D' \cup A'$ that could satisfy the prerequisite of being in a particular table entry could be as large as $n^{\mathcal{O}(k)}$ and thus this would not lead to an FPT algorithm. However, we follow this template for our algorithm and use some more structural properties to prune the family of partial solutions stored at a Dynamic Programming (DP) table entry. Moreover, our algorithm will output $D^* \cup A^*$, where D^* is a nice deletion set, if the input instance is a YES instance.

Our pruning is based on the proof that $D \cup A$ is an independent set in some linear matroid. The first observation towards an FPT algorithm is that after we delete the edges in D, the number of connected components can at most be $|D| - k + 1$. This allows us to show that, in fact, we can think of D being an independent set in the matroid $M_G(\ell)$. That is, ℓ-elongation of the co-graphic matroid, M_G, associated with G, where $\ell = |E(G)| - |V(G)| + k - |D| + 1$ (we refer to preliminaries for the definition). Next, we show that for the addition set A, all we need to store is some form of disjointness and that can be captured using a uniform matroid over the universe $\overline{E(G)}$. Let $U_{m',k-k'}$ be a uniform matroid with ground set $\overline{E(G)}$, where $m' = |\overline{E(G)}|$ and $k' = |D|$. From the definition of $U_{m',k-k'}$, any set A of size at most $k - k'$ is an independent in $U_{m',k-k'}$. We have already explained that we view the deletion set D as an independent set in $M_G(\ell)$ where $\ell = |E(G)| - |V(G)| + k - k' + 1$. Thus, to see the solution set $D \cup A$ as an independent set in a single matroid, we consider the direct sum of $M_G(\ell)$ and $U_{m',k-k'}$; that is, let $M = M_G(\ell) \oplus U_{m',k-k'}$. In M, a set I is an independent set if and only if $I \cap E(G)$ is an independent set in $M_G(\ell)$ and $I \cap \overline{E(G)}$ is an independent set in $U_{m',k-k'}$. This ensures that any solution $D \cup A$ is an independent set in M. By viewing any solution of the problem as an independent set in a matroid M (which is linear), we can use fast computation of q-representative families to prune the DP table entries. However, we still need to take care of a technical requirement in the definition of a nice deletion set. Towards this, we show that for every deletion set D there exists a set of edges W_D, disjoint from D, of size at most $6k$ such that if these edges are not selected then we can satisfy that technical requirement. To achieve this we apply color coding technique and this can be derandomized using universal sets.

In [11], Eulerian Edge Deletion is solved using representative family on the co-graphic matroid associated with the input graph.

2. Preliminaries.

Throughout the paper we use ω to denote the exponent in the running time of matrix multiplication, the current best-known bound is $\omega < 2.373$ [24]. We use $\mathbb{N}$ and $\mathbb{N}^+$ to denote the set of natural numbers and the set of positive integers, respectively. For any $n \in \mathbb{N}^+$, we use $[n]$ to denote the set $\{1, 2, \ldots, n\}$. We use $\mathbb{Q}$ to denote the field of rational numbers. For any multiset A and an element x, by $A(x)$ we denote the number occurrences of x in A. Let U be a set and $W \subseteq U \times \mathbb{N}^+$. For convenience we denote a pair $(v, i) \in U \times \mathbb{N}^+$ (or in W) by $v(i)$. For any $W' \subseteq W$ and $u \in U$, we define

$$W'(u) = \left|\{u(i) \mid i \in \mathbb{N}^+, u(i) \in W'\}\right|.$$

2.1. Graphs.

We use "graph" to denote simple graphs without self-loops, directions, or labels. We use standard terminology from the book of Diestel [5] for those graph-related terms which we do not explicitly define. In general, we use G to denote a graph. We use $V(G)$ and $E(G)$, respectively, to denote the vertex and edge sets of a graph G. For a graph G, we use $\overline{E(G)}$ to denote the simple non-edge set $\binom{V(G)}{2} \setminus E(G)$. For a vertex $v \in V(G)$, we use $E_G(v)$ to denote the set of edges of $E(G)$ incident with v, $\overline{E}_G(v)$ to denote the set of edges of $\overline{E(G)}$ incident with v, and $d_G(v)$ to denote $|E_G(v)|$, i.e., the degree of vertex v. For an edge set $E' \subseteq E(G)$ and $A \subseteq \overline{E(G)}$, we use (i) $V(E')$ to denote the set of *end vertices* of the edges in E', (ii) $G - E'$ to denote the subgraph $G' = (V(G), E(G) \setminus E')$ of G, (iii) $G + A$ to denote the graph $G' = (V(G), E(G) \cup A)$, and (iv) $G[E']$ to denote the subgraph $(V(G), E')$ of G. We say an edge $e \in E(G)$ is a *bridge* if $G - \{e\}$ has more connected components than G.

2.2. Matroids and representative families.

In the next few subsections we give definitions related to matroids. For a broader overview on matroids, we refer the reader to [20].

Definition 2.1. *A pair $M = (E, \mathcal{I})$, where E is a ground set and $\mathcal{I}$ is a family of subsets (called independent sets) of E, is a* matroid *if it satisfies the following conditions:*

(I1) $\emptyset \in \mathcal{I}$.

(I2) *If* $A' \subseteq A$ *and* $A \in \mathcal{I}$, *then* $A' \in \mathcal{I}$.

(I3) *If* $A, B \in \mathcal{I}$ *and* $|A| < |B|$, *then there is* $e \in (B \setminus A)$ *such that* $A \cup \{e\} \in \mathcal{I}$.

An inclusion-wise maximal set of $\mathcal{I}$ is called a *basis* of the matroid. Using axiom (I3) we can show that all the bases of a matroid have the same size. This size is called the *rank* of the matroid M, and is denoted by $\mathsf{rank}(M)$. The rank function of a matroid $M = (E, \mathcal{I})$ is a function $r_M : 2^E \to \mathbb{N}$ which is defined as follows: $r_M(S)$ for any $S \subseteq E$ is the cardinality of the maximum sized independent set contained in S.

Linear matroids and representable matroids. Let A be a matrix over an arbitrary field $\mathbb{F}$ and let E be the set of columns of A. For A, we define a matroid $M = (E, \mathcal{I})$ as follows. A set $X \subseteq E$ is independent (that is, $X \in \mathcal{I}$) if the corresponding columns are linearly independent over $\mathbb{F}$. The matroids that can be defined by such a construction are called *linear matroids*, and if a matroid can be defined by a matrix A over a field $\mathbb{F}$, then we say that the matroid is representable over $\mathbb{F}$; that is, a matroid $M = (E, \mathcal{I})$ of rank d is representable over a field $\mathbb{F}$ if there exist vectors in $\mathbb{F}^d$ corresponding to the elements such that linearly independent sets of vectors correspond to independent sets of the matroid. A matroid $M = (E, \mathcal{I})$ is called *representable* or *linear* if it is representable over some field $\mathbb{F}$.

Direct sum of matroids. Let $M_1 = (E_1, \mathcal{I}_1)$, $M_2 = (E_2, \mathcal{I}_2)$, ..., $M_t = (E_t, \mathcal{I}_t)$ be t matroids with $E_i \cap E_j = \emptyset$ for all $1 \le i \ne j \le t$. The direct sum $M_1 \oplus \cdots \oplus M_t$ is a matroid $M = (E, \mathcal{I})$ with $E := \bigcup_{i=1}^{t} E_i$ and $X \subseteq E$ is independent if and only if $X \cap E_i \in \mathcal{I}_i$ for all $i \le t$. Let $\mathbb{F}$ be a field. Let A_i be a representation matrix over $\mathbb{F}$ of $M_i = (E_i, \mathcal{I}_i)$. Then,

$$A_M = \begin{pmatrix} A_1 & 0 & 0 & \cdots & 0 \\ 0 & A_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & A_t \end{pmatrix}$$

is a representation matrix of $M_1 \oplus \cdots \oplus M_t$. The correctness of this construction is proved in [17].

Proposition 2.2 (see [17, Proposition 3.4]). *Given representations of matroids* $M_1, \ldots, M_t$ *over the same field* $\mathbb{F}$, *a representation of their direct sum can be found in polynomial time.*

Uniform matroids. A pair $M = (E, \mathcal{I})$ over an n-element ground set E, is called a uniform matroid if the family of independent sets is given by $\mathcal{I} = \{A \subseteq E \mid |A| \le k\}$, where k is some constant. This matroid is also denoted as $U_{n,k}$. Every uniform matroid is linear and can be represented over a finite field of size at least n by a $k \times n$ matrix A_M where $A_M[i,j] = j^{i-1}$ and $1, 2, \ldots, n$ are distinct elements in the field.

$$A_M = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 3 & \cdots & n \\ 1 & 2^2 & 3^2 & \cdots & n^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 2^{k-1} & 3^{k-1} & \cdots & n^{k-1} \end{pmatrix}.$$

Co-graphic matroids. Given a graph G with r connected components, the co-graphic matroid associated with G is defined as (U, I), where $U = E(G)$ and

$$I = \{S \subseteq E(G) : \ G - S \text{ has exactly } r \text{ connected components}\}.$$

We use M_G to denote the co-graphic matroid associated with the graph G.

PROPOSITION 2.3 (see [20]). *Co-graphic matroids are representable over any field of size at least* 2.

Elongation of a matroid. The ℓ-*elongation* of a matroid M, where $\ell > \mathsf{rank}(M)$ is defined as a matroid $M' = (E, \mathcal{I}')$ such that $S \subseteq E$ is a basis in M' if and only if $r_M(S) = r_M(E)$ and $|S| = \ell$. We use $M(\ell)$ to denote the ℓ-elongation of a matroid M.

THEOREM 2.4 (see [15]). *Given a representation of a matroid M and an integer $\ell > \mathsf{rank}(M)$, there is a deterministic polynomial time algorithm to compute a representation of ℓ-elongation of M, $M(\ell)$.*

DEFINITION 2.5 (q-representative family [17]). *Given a matroid $M = (E, \mathcal{I})$ and a family $\mathcal{S}$ of subsets of E, we say that a subfamily $\widehat{\mathcal{S}} \subseteq \mathcal{S}$ is* q-representative *for $\mathcal{S}$ if the following holds: for every set $Y \subseteq E$ of size at most q, if there is a set $X \in \mathcal{S}$ disjoint from Y with $X \cup Y \in \mathcal{I}$, then there is a set $\widehat{X} \in \widehat{\mathcal{S}}$ disjoint from Y with $\widehat{X} \cup Y \in \mathcal{I}$. If $\widehat{\mathcal{S}} \subseteq \mathcal{S}$ is q-representative for $\mathcal{S}$, we write $\widehat{\mathcal{S}} \subseteq^q_{rep} \mathcal{S}$.*

THEOREM 2.6 ([15]). *Let $M = (E, \mathcal{I})$ be a linear matroid of rank n and let $\mathcal{S} = \{S_1, \ldots, S_t\}$ be a family of independent sets, each of size b. Let A be an $n \times |E|$ matrix representing M over a field $\mathbb{F}$, where $\mathbb{F} = \mathbb{F}_{p^\ell}$ or $\mathbb{F}$ is $\mathbb{Q}$ (here p is a prime number and $\ell \in \mathbb{N}^+$). Then there is deterministic algorithm which computes a representative family $\widehat{\mathcal{S}} \subseteq^q_{rep} \mathcal{S}$ of size at most $nb\binom{b+q}{b}$, using $\mathcal{O}\big(\binom{b+q}{b} tb^3 n^2 + t\binom{b+q}{b}^{\omega-1}(bn)^{\omega-1}\big) + (n + |E|)^{\mathcal{O}(1)}$ operations over the field $\mathbb{F}$.*

3. An overview of our algorithm.

Let (G, f, k) be an instance of EECG. Our algorithm finds a solution of size *exactly* k (if one such solution exists) and outputs NO otherwise. Any solution to our problem is of the form $D \cup A$, where $D \subseteq E(G)$ and $A \subseteq \overline{E(G)}$. The sets D and A are called the deletion set and the addition set, respectively, corresponding to the solution $D \cup A$.

3.1. Characterizing the solution.

The starting point of our algorithm is a characterization of its solution in terms of a deletion set D satisfying certain properties (such deletion sets are called *nice deletion sets*). This allows us to focus on finding nice deletion sets. To describe the main steps and ideas involved in our algorithm we first give a semi-informal definition of a nice deletion set. Towards this we need the following definitions. We define the set of deficient vertices as

$$\mathsf{def}(G, f) = \{v \mid v \in V(G), f(v) > d_G(v)\}.$$

We also define a set of all pairs (v, i), where v is a deficient vertex and i is an integer from $\{1, \ldots, f(v) - d_G(v)\}$,

$$\mathsf{S}(G, f) = \{(v, i) \mid v \in \mathsf{def}(G, f), i \in \{1, \ldots, f(v) - d_G(v)\}\}.$$

Thus for every deficient vertex v there are $f(v) - d_G(v)$ pairs in $\mathsf{S}(G, f)$ containing v. The set $\mathsf{S}(G, f)$ specifies how many edges in an addition set must be incident with v. Let $\psi : \mathsf{S}(G, f) \to \mathsf{S}(G, f)$ be an involution. Given ψ, we define a multiset E_ψ as follows. It consists of all possible pairs (u, v) such that $u, v \in \mathsf{def}(G, f)$ and $\psi(u, i) = (v, j)$ for some $j \geq i$. Essentially, map ψ will allow us to obtain the corresponding addition set A. We say that ψ is a *proper deficiency map* if, for every $u \in V(G)$ $(u, u) \notin E_\psi$, E_ψ is a set (not a multiset), and $E_\psi \cap E(G) = \emptyset$. Finally, a set $D \subseteq E(G)$ is called a *nice deletion set* if

(i) For all $v \in V(G)$, $d_{G-D}(v) \leq f(v)$.

(ii) $|\mathsf{S}(G-D, f)| = 2(k - |D|)$.

(iii) Graph $G - D$ has at most $k - |D| + 1$ connected components.

(iv) If $G - D$ is not connected, then each connected component in $G - D$ contains a vertex v deficient in $G - D$, i.e. such that $d_{G-D}(v) < f(v)$.

(v) There exists a proper deficiency map $\psi : \mathsf{S}(G-D, f) \to \overline{\mathsf{S}(G-D, f)}$.

Let $D \subseteq E(G)$. As we will see in Lemma 4.4, there exists $A \subseteq \overline{E(G)}$, $|A| = k - |D|$ such that $A \cup D$ is a solution to EECG if and only if D is a nice deletion set. Furthermore, given a nice deletion set we can find an addition set, if it exists, in polynomial time. Thus, our problem reduces to finding a nice deletion set $D \subseteq E(G)$ and an accompanying proper deficiency map ψ on $\mathsf{S}(G-D, f)$, if it exists. We use a dynamic programming (DP) algorithm to compute a nice deletion set D.

3.2. Towards the states of the dynamic programming algorithm. So, how can one find a nice deletion set? Throughout this section we will work with a hypothetical deletion set D. We partition the vertex set of G into sets of Green and Red vertices: We color $v \in V(G)$ green if $d_G(v) > f(v)$, and red otherwise. Let $E_r = E(G[\mathsf{Red}])$ and $E_g = E(G) \setminus E_r$. We need a quick sanity check; that is, if $\sum_{\{v : d_G(v) \neq f(v)\}} |d_G(v) - f(v)| > 2k$, then we output No, because in this case any solution to EECG requires more than k edge edits (addition/deletion operations). Now we guess the size $k' \leq k$ of D such that $2k' \geq \sum_{v \in \mathsf{Green}} d_G(v) - f(v)$. Since D is our hypothetical deletion set, we have that for any $v \in \mathsf{Green}$, the number of edges in D which are incident with v is at least $d_G(v) - f(v)$. Now we guess the number k_1 of edges in D which are incident with only green vertices and the number k_2 of edges in D which are incident with at least one vertex in Red. Note that $k_1 + k_2 = k'$. Also note that the number of ways we can guess (k', k_1, k_2) is at most k^2. Now for every $v \in \mathsf{Green}$, we guess the number of edges in D which are incident with v. In particular, we guess a function $\Phi : \mathsf{Green} \to \mathbb{N}$ such that for all $v \in \mathsf{Green}$ we have that $\Phi(v) \geq d_G(v) - f(v)$. The number of possible functions $\Phi(v)$ is upper bounded by $\mathcal{O}(4^k k)$ (a proper analysis of this bound is explained in section 4). From now on we will assume that we are given a function Φ. In other words we have guessed the function Φ corresponding to the hypothetical solution D. We say that a solution $D \cup A$ to EECG satisfies the function Φ if for every vertex $v \in \mathsf{Green}$, the number of edges incident with v in D is exactly $\Phi(v)$.

We start with an intuitive explanation of the structure of the solution that is helpful in designing partial solution for the DP algorithm. Given $D \cup A$, we first define a notion of an alternating walk. An *alternating walk* is a sequence of vertices $u_1 u_2 \cdots u_\ell$ such that consecutive pairs of vertices $((u_i, u_{i+1}), (u_{i+1}, u_{i+2}))$ either belong to $D \times A$ or $A \times D$ and $\{(u_i, u_{i+1}) \mid 1 \leq i < \ell\}$ is a set (not a multiset); that is, an edge from D is followed by an edge from A or vice-versa. In an *alternating even length closed walk*, $u_1 = u_\ell$ and the first and the last edge in the walk are of different type (i.e., one from D and one from A). One might wonder about the definition of alternating odd length closed walk. For our purposes we will think of them as alternating walks that start and end at the same vertex and the first and the last edge either both belong to D or both belong to A. From now on whenever we say an alternating closed walk, we mean an alternating even length closed walk; see Figure 3.1 for an illustration of alternating (closed) walks. For every intermediate, i.e., not the endpoint, vertex in an alternating walk or in an alternating closed walk, one of the edges incident with it belongs to D while the second edge belongs to A. Thus the degree of any vertex is not disturbed

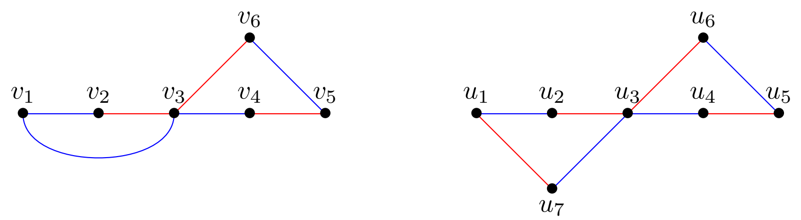

FIG. 3.1. *Illustration of alternating (closed) walks. The blue and red edges belong to D and A, respectively. The sequence $v_1v_2v_3v_4v_5v_6v_3v_1$ is an alternating walk starting and ending at v_1. The sequence $u_1u_2u_3u_4u_5u_6u_3u_7u_1$ is an alternating closed walk. (Figure in color online.)*

by an alternating walk where this vertex is intermediate. We define alternating walks for the following reason. Let $D \cup A$ be a solution of EECG that satisfies Φ. We can think of edges in $D \cup A$ forming *a family $\mathcal{P}$ of edge disjoint alternating walks and alternating closed walks* with the following properties:

- For every vertex $v \in V(G)$ and a set $Z \in \{D, A\}$, we define $\mathsf{apdeg}(\mathcal{P}, Z, v)$ as the number of edges from Z that are incident with v and appear as (i) the first edge in alternating walks from $\mathcal{P}$ that start with v; and (ii) the last edge in alternating walks from $\mathcal{P}$ that end in v. Note that if there is an alternating walk that both starts and ends in v and the start edge as well as the last edge belong to Z, then this walk contributes two to $\mathsf{apdeg}(\mathcal{P}, Z, v)$ (this happens when the alternating walk $u_1 \ldots u_\ell$ is of odd length and $v = u_1 = u_\ell$). For every vertex $v \in \mathsf{Green}$, we require that $\mathsf{apdeg}(\mathcal{P}, D, v) = d_G(v) - f(v)$ and $\mathsf{apdeg}(\mathcal{P}, A, v) = 0$. Furthermore, for every vertex $v \in \mathsf{Red}$, $\mathsf{apdeg}(\mathcal{P}, D, v) = 0$ and $\mathsf{apdeg}(\mathcal{P}, A, v) = f(v) - d_G(v)$. When the number of edges in D which are incident on $v \in \mathsf{Green}$ is greater than $d_G(v) - f(v)$, then the number of times v appears as intermediate vertices in alternating (closed) walks is exactly equal to the number of *excess* edges and these excess edges will not contribute to $\mathsf{apdeg}(\mathcal{P}, D, v)$.
- Every vertex $v \in \mathsf{Green}$ appears as an intermediate vertex in an alternating walk or in an alternating closed walk of $\mathcal{P}$ exactly $\Phi(v) - d_G(v) + f(v)$ times.

This walk system view allows us to make a dynamic programming algorithm where we can move from one state to another using one edge addition or deletion. In particular, the algorithm works by constructing first all alternating walks $P_1, \ldots, P_\eta$ and then alternating closed walks $P_{\eta+1}, \ldots, P_\alpha$. Given a partially constructed walk system, we try to append an edge to the current walk we are constructing by adding an edge from $\binom{V(G)}{2}$ to it; or declaring that we are finished with the current walk and move to obtain a new walk. During this process we also keep a partial proper deficiency map ψ' such that $E_{\psi'}$ are addition-edges in the current partial solution. Thus, a state in the dynamic programming algorithm is given by our current guesses and a subset of domain of partial proper deficiency map. It can be shown that the number of states is upper bounded by $2^{\mathcal{O}(k)}$. However, the number of sets $D' \cup A'$ that could satisfy the prerequisite of being in a particular table entry could be as large as $n^{\mathcal{O}(k)}$, and thus this would not lead to an FPT algorithm. However, this is indeed a template for our algorithm. Next, we show how to prune table entry size to $2^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$, thus obtaining an FPT algorithm.

3.3. Pruning the DP table entry and an FPT algorithm. We need to prune the DP table in a way that we do not change the answer to the given instance

(G, f, k). Towards this we show that if some subset we have stored in a DP table entry could lead to a nice deletion set, then we have at least one such set after the pruning operation. Our guessing of k_1, k_2, and Φ allows us to satisfy the properties (i) and (ii) of nice deletion set. Property (iii) of nice deletion sets implies that D is an independent set in the matroid $M_G(\ell)$—the ℓ-elongation of the co-graphic matroid M_G associated with G, where $\ell = |E(G)| - |V(G)| + k - |D| + 1$ (we refer to preliminaries for the definition). Thus by considering only those D which are independent sets in $M_G(\ell)$, we ensure that property (iii) of nice deletion set is satisfied. Now consider the property (v) of nice deletion set, i.e., there exists a proper deficiency map $\psi : S(G - D, f) \to S(G - D, f)$. Our objective is to get a set $D \cup A$ such that there is a proper deficiency map ψ over $S(G - D, f)$ with $E_\psi = A$, along with other properties as well. Let $D_1 \cup A_1, D_2 \cup A_2$ be two partial solutions belonging to the same equivalence class, where $D_1, D_2 \subseteq E(G)$ and $A_1, A_2 \subseteq \overline{E(G)}$. Suppose that $D' \subseteq E(G)$, $A' \subseteq \overline{E(G)}$, $(D_1 \cup D') \bigcup (A_1 \cup A')$ is a solution and $A_2 \cap A' = \emptyset$. Since $D_1 \cup A_1$, $D_2 \cup A_2$ belongs to the same equivalence class and A_2 and A' are disjoint, there is a proper deficiency map ψ' over $S(G - (D_2 \cup D'), f)$ such that $E_{\psi'} = A_2 \cup A'$. To take care of the disjointness property between the current addition set and the future addition set while doing DP, we view the addition set A of the solution as an independent set in a uniform matroid over the universe $\overline{E(G)}$. Let $U_{m',k-k'}$ be a uniform matroid with ground set $\overline{E(G)}$, where $m' = |\overline{E(G)}|$. From the definition of $U_{m',k-k'}$, every set A of size at most $k - k'$ is independent in $U_{m',k-k'}$. We have mentioned that D is an independent set in $M_G(\ell)$ where $\ell = |E(G)| - |V(G)| + k - k' + 1$. To view the solution set $D \cup A$ as an independent set in a matroid, we consider the direct sum $M = M_G(\ell) \oplus U_{m',k-k'}$ of two matroids. In M, set I is independent if and only if $I \cap E(G)$ is an independent set in $M_G(\ell)$ and $I \cap \overline{E(G)}$ is an independent set in $U_{m',k-k'}$. This ensures that any solution $D \cup A$ is an independent set in M. By viewing any solution of the problem as an independent set in the matroid M (which is linear), we can use the q-representative families to prune the table. However, we still need to ensure that property (iv) of nice deletion set is satisfied. In what follows we explain how we achieve this.

We show that for every deletion set D, there exists a set of edges $W_D \subseteq E(G)$ disjoint from D, of size at most $6k$ such that if these edges are not selected, then we can guarantee that each connected component in $G - D$ contains a vertex v such that $d_{G-D} < f(v)$. To achieve this we apply color coding. We randomly color each edge orange with probability $1/7$ and black with probability $6/7$, then with probability at least $(1/7)^k(6/7)^{6k}$, all the edges of the deletion set D will be colored with orange and all the edges in W_D will be colored with black. If we repeat our algorithm $(7^{7k}/6^{6k})$ times, we can increase the success probability to a constant. Finally, this step is derandomized using universal sets.

4. Algorithm.

For a given graph G, a solution to EECG comprises a *deletion set* $D \subseteq E(G)$ and an *addition set* $A \subseteq \overline{E(G)}$. That is, $F = G - D + A$. In particular, we denote the solution set as a pair (D, A), where $D \subseteq E(G)$ and $A \subseteq \overline{E(G)}$. For an instance (G, f, k) of EECG, our algorithm finds a solution of size *exactly* k (if one such solution exists), otherwise outputs No.

4.1. Structural characterization.

In this subsection we give a structural characterization of a solution that is central to our parameterized algorithm. In particular, we give necessary and sufficient conditions on $D \subseteq E(G)$ for being a deletion set of an optimum solution to EECG. Furthermore, we show that if we have D that satisfies the desired conditions, then we can obtain the corresponding addition set A in polynomial time. We start with a few definitions that will set up a useful language to speak about different aspects of the sets we will be considering throughout the paper.

Now we recall the definitions of $\mathsf{def}(G,f)$ and $\mathsf{S}(G,f)$ for convenience.

DEFINITION 4.1. *Let G be a graph and let $f : V(G) \to \{1,2,\dots,d\}$ be a function. We call a vertex $v \in V(G)$* deficient *with respect to G and f if $f(v) > d_G(v)$. We define the* deficiency vertex set *of G and f, denoted by $\mathsf{def}(G,f)$, as the set of deficient vertices with respect to G and f. We define the* deficiency set *of G and f, denoted by $\mathsf{S}(G,f)$, as a set containing $f(v) - d_G(v)$ many elements corresponding to each vertex $v \in \mathsf{def}(G,f)$. Notice that the cardinality of $\mathsf{S}(G,f)$ is equal to $\sum_{\{v : f(v) > d_G(v)\}} f(v) - d_G(v)$. In particular, these sets are defined as*

$$\mathsf{def}(G,f) = \{v \mid v \in V(G), f(v) > d_G(v)\},$$
$$\mathsf{S}(G,f) = \{(v,i) \mid v \in V(G), f(v) > d_G(v), i \in [f(v) - d_G(v)]\}.$$

Let $W \subseteq V(G) \times \mathbb{N}^+$. Recall that we denote a pair $(v,i) \in V(G) \times \mathbb{N}^+$ (or in W) by $v(i)$. Let $\psi : W \to W$ be an involution. Given ψ, we define a multiset E_ψ as follows. For each $u(i) \in W$ we add (u,v) to E_ψ if $\psi(u(i)) = v(j)$ for some $j \geq i$.

DEFINITION 4.2. *Let G be a graph and let $W \subseteq V(G) \times \mathbb{N}^+$. An involution $\psi : W \to W$ is called a* proper deficiency map *if it satisfies the following properties:*

1. *Non-loop property: For each $u \in V(G)$, $(u,u) \notin E_\psi$.*
2. *Simple edge property: E_ψ is a set, not a multiset; that is, there is no pair $u, v \in V(G)$ such that $\psi(u(i_1)) = v(j_1)$ and $\psi(u(i_2)) = v(j_2)$ for some $i_1 \neq i_2$ and $j_1 \neq j_2$.*
3. *Non-edge property: $E_\psi \cap E(G) = \emptyset$.*

In general, we will have proper deficiency map over the domain $\mathsf{S}(G,f)$ or some set related to this. Finally, we define the notion of *nice deletion set.*

DEFINITION 4.3. *Let (G,f,k) be an instance of* EECG *and let $D \subseteq E(G)$. We say that D is a* nice deletion set *if the following properties are satisfied:*

(i) *For all $v \in V(G)$, $d_{G-D}(v) \leq f(v)$.*

(ii) $|\mathsf{S}(G-D,f)| = 2(k-|D|)$.

(iii) *The graph $G-D$ has at most $k-|D|+1$ connected components.*

(iv) *If $G-D$ is not connected, then each connected component in $G-D$ contains a deficient vertex with respect to $G-D$ and f (i.e., for each connected component F in $G-D$, $V(F) \cap \mathsf{def}(G-D,f) \neq \emptyset$).*

(v) *There exists a proper deficiency map $\psi : \mathsf{S}(G-D,f) \to \mathsf{S}(G-D,f)$.*

LEMMA 4.4. *Let (G,f,k) be an instance of* EECG *and let $D \subseteq E(G)$. Then there exists $A \subseteq \overline{E(G)}$, $|A| = k-|D|$ such that $A \cup D$ is a solution to* EECG *if and only if D is a nice deletion set. Moreover, given a nice deletion set $D \subseteq E(G)$ we can find $A \subseteq \overline{E(G)}$ such that $|A| = k-|D|$ and $D \cup A$ is a solution to* EECG *in polynomial time.*

Proof. ($\Rightarrow$) Let $A \subseteq \overline{E(G)}, |A| = k-|D|$ be such that $A \cup D$ is a solution to EECG. We need to show that $D \subseteq E(G)$ is a nice deletion set. Since $A \cup D$ is a solution to EECG, we have that $d_{G-D}(v) \leq f(v)$ for all $v \in V(G)$, satisfying condition (i). Furthermore, $A \cup D$ being a solution also implies that $\sum_{\{v\ :\ f(v) > d_{G-D}(v)\}} f(v) - d_{G-D}(v) = 2|A| = 2(k-D)$. Hence $|\mathsf{S}(G-D,f)| = 2(k-|D|)$, satisfying condition (ii). Since $G-D+A$ is a connected graph, $G-D$ can have at most $|A|+1 = k-|D|+1$ connected components, satisfying condition (iii) in the definition. The graph $G-D+A$

is connected and thus each connected component F in $G-D$ (if $G-D$ is not connected) contains a vertex $v \in V(F)$ such that $(v,u) \in A$ for some $u \in V(G)$. Since $D \cup A$ is a solution to EECG, $d_{G-D+A}(v) = f(v)$ and hence $d_{G-D}(v) < f(v)$ (because $(v,u) \in A$), satisfying condition (iv). Finally, we show that D satisfies the last property. Let $A = \{e_1, e_2, \ldots, e_r\} \subseteq \overline{E(G)}$, where $r = k-|D|$. Since $D \cup A$ is a solution to EECG, we have that for any vertex v, there are exactly $f(v) - d_{G-D}(v)$ edges in A which are incident to v. Now we define a bijection $\psi : \mathsf{S}(G-D,f) \to \mathsf{S}(G-D,f)$ as follows: $\psi(u(i)) = v(j)$ if $(u,v) = e_\ell$ such that there are exactly $i-1$ edges from $\{e_1, \ldots, e_{\ell-1}\}$ incident on u and there are exactly $j-1$ edges from $\{e_1, \ldots, e_{\ell-1}\}$ incident on v; that is, we traverse the edges $e_1, \ldots, e_r$ from left to right and find the ith edge incident with u, say $e_\ell = (u,v)$, and then we assign it $v(j)$ if there are exactly $j-1$ edges incident with v present among $\{e_1, \ldots, e_{\ell-1}\}$.

CLAIM 1. *$\psi : \mathsf{S}(G-D,f) \to \mathsf{S}(G-D,f)$ is a proper deficiency map.*

Proof. By the definition of ψ, if $\psi(u(i)) = v(j)$, then $\psi(v(j)) = u(i)$, and so ψ is an involution. Since $G-D+A$ is a simple graph, for any $u \in V(G)$, $(u,u) \notin E_\psi$. Now we need to show that E_ψ is not a multiset. Suppose not, then there exists $u,v \in V(G)$ such that $\psi(u(i_1)) = v(j_1)$ and $\psi(u(i_2)) = v(j_2)$ for some $i_1 \neq i_2$ and $j_1 \neq j_2$. This implies that there exist $e_i, e_j \in A$, $i \neq j$ such that $e_i = e_j = (u,v)$. This contradicts the fact that $G-D+A$ is a simple graph. Since A is disjoint from $E(G)$, $E_\psi \cap E(G) = \emptyset$. This completes the proof of the claim. □

($\Leftarrow$) Let $D \subseteq E(G)$ be a nice deletion set. We need to show that we can find $A \subseteq \overline{E(G)}$ such that $|A| = k-|D|$ and $G-D+A$ is a solution to EECG. Properties (i) and (ii) imply that $d_{G-D}(v) \leq f(v)$ for all $v \in V(G)$ and $|\mathsf{S}(G-D,f)| = 2(k-|D|)$. Due to the property (v) in the definition of nice deletion set, we know that there exists a proper deficiency map $\psi : \mathsf{S}(G-D,f) \to \mathsf{S}(G-D,f)$. Define $A_1 = E_\psi$. By the definition of E_ψ, $|A_1| = |E_\psi| = |\mathsf{S}(G-D,f)|/2 = k-|D|$. Also note that $A_1 \cap E(G) = \emptyset$ because ψ is a proper deficiency map. Now consider the graph $G_1 = G-D+A_1$. Note that G_1 is simple graph because ψ is a proper deficiency map. Also by the definition of E_ψ, $d_{G-D+A_1}(v) = f(v)$ for all $v \in V(G)$. So G_1 satisfies the degree constraints. If G_1 is connected, then $D \cup A_1$ is a solution to EECG. Thus, we assume that G_1 is not connected. In what follows we give an iterative procedure that finds the desired A. Suppose we are in ith iteration and we have a set A_i such that $G-D+A_i$ satisfies all degree constraints but $G-D+A_i$ is not connected. Then in the next iteration we find another addition set of size $k-|D|$, say A_{i+1}, such that $G-D+A_{i+1}$ satisfies all degree constraints and $G-D+A_{i+1}$ has strictly less connected components than in $G-D+A_i$. The procedure is started with $A_1 = E_\psi$. Let $i \geq 1$ and we have A_i such that $G-D+A_i$ satisfies all degree constraints but $G-D+A_i$ is not connected. Since $|A_i| = k-|D|$ and $G-D$ has at most $k-|D|+1$ connected components, $G_i = G-D+A_i$ has a component F such that there is an edge $(u_1,v_1) \in A_i$ with the property that $u_1, v_1 \in V(F)$ and (u_1,v_1) is not a bridge in F. Let F' be another connected component in G_i. Since each connected component in $G-D$ contains a vertex $w \in V(G)$ such that $d_{G-D}(w) < f(w)$, there exists an edge $(u_2,v_2) \in A_i$ such that $u_2, v_2 \in V(F')$. Now let $A_{i+1} = (A_i \setminus \{(u_1,v_1),(u_2,v_2)\}) \cup \{(u_1,u_2),(v_1,v_2)\}$. Observe that $G_{i+1} = G-D+A_{i+1}$ is a simple graph with a strictly smaller number of connected components than in G_i and $d_{G_{i+1}}(v) = f(v)$ for all $v \in V(G)$. Observe that when the procedure stops, we find the desired A.

Given a nice deletion set D, we can find the desired A using the iterative procedure described in the reverse direction of the proof. Clearly, this procedure can be implemented in polynomial time. This completes the proof of the lemma. □

4.2. An algorithm with running time $n^{\mathcal{O}(k)}$. In this subsection we design an algorithm for EECG running in time $n^{\mathcal{O}(k)}$—an XP algorithm. Clearly, this is not the algorithm we promised. In fact a simple brute force algorithm for EECG will run in time $n^{\mathcal{O}(k)}$. But this algorithm will provide a skeleton for a parameterized algorithm given in the next subsection. Both the algorithms, XP as well as FPT for EECG, are based on DP. The algorithm given in this subsection allows us to introduce various aspects of the dynamic programming in a gentler manner. Even though the number of states in the dynamic programming algorithm given in this subsection is bounded by $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$, the number of partial solutions stored in a DP table entry could potentially be $n^{\mathcal{O}(k)}$. The FPT algorithm given in the next subsection is based on this algorithm and uses tools from representative family to reduce the cardinality of partial solutions stored in any DP table entry to $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$.

Given an instance (G, f, k) of EECG. The main idea of the algorithm is to find a nice deletion set $D \subseteq E(G)$ and an accompanying proper deficiency map ψ on $\mathsf{S}(G-D, f)$, if it exists, using dynamic programming. *Throughout this section we will work with a hypothetical deletion set D.* We start with coloring the vertices of G, *green* and *red* in the following way. We color $v \in V(G)$ green if $d_G(v) > f(v)$, otherwise we color v red; that is, $\mathsf{Green} = \{v \mid v \in V(G), d_G(v) > f(v)\}$ and $\mathsf{Red} = V(G) \setminus \mathsf{Green}$. Let $E_r = E(G[\mathsf{Red}])$ and $E_g = E(G) \setminus E_r$. We need a quick sanity check; that is, if $\sum_{\{v : d_G(v) \neq f(v)\}} |d_G(v) - f(v)| > 2k$, then we output No, because in this case any solution to EECG requires more than k edge edits (addition/deletion operations). Now we guess the size $k' \leq k$ of D such that $2k' \geq \sum_{v : d_G(v) > f(v)} d_G(v) - f(v)$. Since D is our hypothetical deletion set, we have that for any $v \in \mathsf{Green}$, the number of edges in D which are incident with v is at least $d_G(v) - f(v)$. Now we guess the number k_1 of edges in D which are incident with only green vertices and the number k_2 of edges in D which are incident with at least one vertex in Red. Note that $k_1 + k_2 = k'$. Also note that the number of ways we can guess (k', k_1, k_2) is at most k^2. Now for every $v \in \mathsf{Green}$, we guess the number of edges in D which are incident with v. In particular, we guess a function $\Phi : \mathsf{Green} \to \mathbb{N}$ such that for all $v \in \mathsf{Green}$ we have that $\Phi(v) \geq d_G(v) - f(v)$. We claim that the number of possible functions from Green to $\mathbb{N}$ that represent the number of edges incident with a vertex $v \in \mathsf{Green}$ in the hypothetical solution D is bounded by $\mathcal{O}(4^k k)$. Let k_3 be the number of edges in D which are incident with only red vertices. Observe that we are looking for functions Φ, such that for all $v \in \mathsf{Green}$ we have that $1 \leq d_G(v) - f(v) \leq \Phi(v) \leq k_1 + k_2 - k_3$, and $\sum_{v \in \mathsf{Green}} \Phi(v) \leq 2k_1 + k_2 - k_3$. The number of such functions is upper bounded by

$$\mathcal{O}(2^{|\mathsf{Green}| + 2k_1 + k_2 - k_3 - \sum_{v \in \mathsf{Green}} d_G(v) - f(v)}).$$

The above quantity is upper bounded by $\mathcal{O}(4^k)$ because $|\mathsf{Green}| \leq \sum_{v \in \mathsf{Green}} d_G(v) - f(v)$ and $2k_1 + k_2 - k_3 \leq 2k$. Since there are at most k possible values for k_3, the number of possible functions $\Phi(v)$ is upper bounded by $\mathcal{O}(4^k k)$. From now on we will assume that we are given a function Φ. In other words we have guessed the function Φ corresponding to the hypothetical solution D. We say that a solution $D \cup A$ to EECG satisfies the function Φ if for every vertex $v \in \mathsf{Green}$ the number of edges incident with v in D is exactly equal to $\Phi(v)$.

Intuitive structure of the algorithm. We start with an intuitive explanation of the structure of a solution that helps us in designing a partial solution for the DP algorithm. Any solution to our problem is of the form $D \cup A \in \binom{V(G)}{2}$ where $D \subseteq E(G)$ and $A \subseteq \overline{E(G)}$. Given $D \cup A$, we first define a notion of an alternating walk.

An *alternating walk* is a sequence of vertices $u_1u_2\dots u_\ell$ such that consecutive pairs $((u_i,u_{i+1}),(u_{i+1},u_{i+2}))$ either belong to $D\times A$ or $A\times D$ and no edges from $D\cup A$ repeats in the walk; that is, an edge from D is followed by an edge from A or vice-versa. In an *alternating even length closed walk*, $u_1=u_\ell$ and the starting and ending edges are of different type. One might wonder about the definition of *alternating odd length closed walk*. For our purposes we will think of them as alternating walks that start and end at the same vertex, and the first and the last edge either both belong to D or both belong to A. From now on whenever we say an alternating closed walk, we mean an alternating even length closed walk. For every intermediate vertex in an alternating (closed) walk, one of the edges incident with it belongs to D and the second edge belongs to A. Thus the degree of any vertex is not disturbed by an alternating walk where this vertex is intermediate. We define alternating walks for the following reason. Let $D\cup A$ be a solution of EECG that satisfies Φ. We can think of edges in $D\cup A$ forming *a family $\mathcal{P}$ of edge disjoint alternating (closed) walks* with the following properties:

- For every vertex $v\in V(G)$ and a set $Z\in\{D,A\}$, we define $\mathsf{apdeg}(\mathcal{P},Z,v)$ as the number of edges from Z that are incident with v and appear as (i) the first edge in alternating walks from $\mathcal{P}$ that start with v and (ii) the last edge in alternating walks from $\mathcal{P}$ that end in v. Note that if there is an alternating walk that both starts and ends in v and the start edge as well as the last edge belong to Z, then this path contributes two to $\mathsf{apdeg}(\mathcal{P},Z,v)$. For every vertex $v\in\mathsf{Green}$, $\mathsf{apdeg}(\mathcal{P},D,v)=d_G(v)-f(v)$ and $\mathsf{apdeg}(\mathcal{P},A,v)=0$. Furthermore, for every vertex $v\in\mathsf{Red}$, $\mathsf{apdeg}(\mathcal{P},D,v)=0$ and $\mathsf{apdeg}(\mathcal{P},A,v)=f(v)-d_G(v)$.
- Every vertex $v\in\mathsf{Green}$, appears as an intermediate vertex in an alternating (closed) walk of $\mathcal{P}$ is exactly equal to $\Phi(v)-d_G(v)+f(v)$.

For any solution $\mathcal{P}=\{P_1,P_2,\dots,P_\alpha\}$, without loss of generality we assume that there is η such that $P_1,\dots,P_\eta$ are alternating walks and $P_{\eta+1},\dots,P_\alpha$ are alternating closed walks. *In our solution we will first construct all alternating walks and then construct all alternating closed walks. Also for any alternating closed walk, we always start with a deletion edge.*

Towards an implementation of the intuitive description. Our objective is to design a DP algorithm. Thus, we first need to define a notion of a partial solution which will constitute a basic building block of our algorithm. We first explain partial solutions and its structure which will be utilized to design the algorithm. Any solution to our problem is of the form $D\cup A\in\binom{V(G)}{2}$ where $D\subseteq E(G)$ and $A\subseteq\overline{E(G)}$, thus the partial solution for the problem is also a subset $B\cup A'\in\binom{V(G)}{2}$ where $B\subseteq E(G)$ and $A'\subseteq\overline{E(G)}$. Let $D\cup A$ be a solution of EECG that satisfies Φ. As described before we think of edges in $D\cup A$ forming a family, $\mathcal{P}=\{P_1,\dots,P_s\}$, of alternating walks and alternating closed walks. A partial solution can be thought of as $\{P_1,\dots,P_j^*\}$, where we have already created $P_1,\dots,P_{j-1}$ and P_j^* is some subwalk of P_j that we are creating now. This view could be useful in understanding the algorithm we are going to describe later. *At this point we add a caveat that our algorithm slightly differs from this perspective to make the proof more accessible.*

Let $\mathcal{P}'=\{P_1,\dots,P_j^*\}$ be a partial solution. We first assume that the P_j^* that we are constructing is going to be an alternating walk. For every vertex $v\in V(G)$ and a set $Z\in\{B,A'\}$, we define $\mathsf{apdeg}^*(\mathcal{P}',Z,v)$ as the number of edges from Z that are incident with v and appear as (i) the first edge in alternating walks from $\mathcal{P}'$ that start with v; and (ii) the last edge in alternating walks from $\{P_1,\dots,P_{j-1}\}$ that end in v. As before, if there is an alternating walk that both starts and ends in v and the start edge as well as the last edge belong to Z, then this path contributes two to $\mathsf{apdeg}^*(\mathcal{P}', Z, v)$. If the$P_j^*$ that we are constructing is going to be an alternating closed walk, then $\mathsf{apdeg}^*(\mathcal{P}', Z, v) = \mathsf{apdeg}(\{P_1, \dots, P_{j-1}\}, Z, v)$. Before we go further we would like to add that while making an algorithm we will know whether we are currently constructing an alternating walk or an alternating closed walk.

In our algorithm we form partial solutions of size i from partial solutions of size $i-1$. It is important for designing the FPT algorithm given in the next subsection to partition the set of partial solutions based on some of its structures. Note that we have already guessed some structure of the solution: like the number of deletion edges, the number of deletion edges with both endpoints in Green, the number of addition edges, and the number of deletion edges from $E(v)$ for each vertex $v \in \mathsf{Green}$ (via guessing the function Φ). We use these structures to characterize the equivalence classes of partial solutions. Since we are going to compute a solution which respects Φ, in the description of an equivalence class over partial solution we would like to include for every vertex $v \in \mathsf{Green}$ the number of edges that are incident with v and contribute to $\mathsf{apdeg}^*(\mathcal{P}, D, v)$. This is achieved by creating a multiset set T_m containing $d_G(v) - f(v)$ many copies of v for all $v \in \mathsf{Green}$ and keeping a subset of T'_m for each equivalence class; that is, T'_m tells us how many edges incident with a vertex $v \in \mathsf{Green}$ must be present in the partial solutions that respect $\mathsf{apdeg}^*(\mathcal{P}', B, v)$. In other words, $\mathsf{apdeg}^*(\mathcal{P}', B, v) = T'_m(v)$.

Now we define another set T_g for Green vertices that stores the information about how many times a vertex v appears as an intermediate vertex in the current partial solution $\mathcal{P}' = \{P_1, \dots, P_j^*\}$. In particular, we define

$$T_g = \{v(i) \mid v \in \mathsf{Green}, \Phi(v) > d_G(v) - f(v), i \in [\Phi(v) - (d_G(v) - f(v))]\}.$$

We use $T'_g \subseteq T_g$ to represent an equivalence class with the property that a partial solution $\mathcal{P}' = \{P_1, \dots, P_j^*\}$ satisfies that for every vertex $v \in \mathsf{Green}$ the number of times v appears as an intermediate vertex in $\mathcal{P}'$ is the same as the number of elements corresponding to it in T'_g. Observe that we are differentiating between elements corresponding to a vertex and a copy of a vertex. For example, consider a vertex v. When we say that T_m contains three copies of a vertex v, we mean $\{v, v, v\} \subseteq T_m$ and when we say T_g contains three elements corresponding to v, then we mean that $\{v(1), v(2), v(3)\} \subseteq T_g$.

We have taken care of all alternating walks that start or end with deletion edges as well as alternating closed walk. We also have some alternating walks that start or end with an addition edge. Now we define some sets that will help us to have some control over these alternating walks. Towards this we define a set T_r as follows: T_r is a set such that for every $v \in \mathsf{Red}$ there are exactly $f(v) - d_G(v)$ elements in T_r corresponding to v:

$$T_r = \{v(i) \mid v \in \mathsf{Red}, f(v) > d_G(v), i \in [f(v) - d_G(v)]\}.$$

Ideally, we would like to keep a set $T'_r \subseteq T_r$ to represent an equivalence class with the property that a partial solution $\mathcal{P}' = \{P_1, \dots, P_j^*\}$ satisfies that for every vertex $v \in \mathsf{Red}$, $\mathsf{apdeg}^*(\mathcal{P}', A', v) = T'_r(v)$. However, for technical reasons we represent it as follows. For every vertex $v \in \mathsf{Red}$, $\mathsf{apdeg}^*(\mathcal{P}', A', v) = (T_r \setminus T'_r)(v)$.

Let v be the last vertex of P_j^*. When we are constructing P_j^* it can happen that the edge incident with v could either be a deletion edge or an addition edge. In either case we do not know whether v is the last vertex of P_j^* and thus we cannot store the information on v either using T'_g or T'_r. To overcome this difficulty, we keep a multiset X of size at most two or a set Y of size at most one. If the last edge is an addition edge, then we store v in Y, and if the last edge is a deletion edge, then we store v in X. Now we explain why X is a multiset and $|X| \leq 2$. If P_j^* is going to be an alternating closed walk, then, in fact, we need to store information about both of its end vertices. In other words, if P_j^* is going to be an alternating closed walk, then P_j^* starts with a deletion edge and the information about its starting vertex is stored in X. If P_j^* ends with deletion edge, then the information about it is stored in X and the end vertex very well could be same as the starting vertex of P_j^*. Thus, X could be a multiset of size 2.

Informal description of the algorithm. Our algorithm works as follows. First, we construct all alternating walks $P_1, \ldots, P_\eta$ and then construct alternating closed walks $P_{\eta+1}, \ldots, P_\alpha$. Now we explain how each alternating (closed) walk can be constructed. Suppose we have completed constructing P_{j-1}. We start constructing P_j in the following preference order.

1. If $T_m \neq T'_m$, then we start with a deletion edge incident with $v \in T_m \setminus T'_m$ and add a copy of v to T'_m. Let P_j^* denotes the current partial alternating walk that we have constructed so far. If P_j^* ends in a vertex $u \in T_m \setminus T'_m$ with a deletion edge, then we say $P_j = P_j^*$ and add a copy of u to T'_m. If P_j^* ends in a vertex u with an addition edge and there is an element corresponding to u in T'_r, then we say $P_j = P_j^*$ and we delete an element corresponding to u from T'_r. Else, we continue constructing this alternating walk by either adding or deleting an appropriate edge incident with u. Notice that if u is a vertex in Green, then the last edge of P_j^* will be accounted in the next step using T'_g. However, if $u \in \mathsf{Red}$, then we do not account for these edges using any set.
2. If $T'_r \neq \emptyset$, then we start with an addition edge incident with a vertex v, where $T'_r(v) \neq 0$ and we delete an element corresponding to v from T'_r. Let P_j^* denotes the current partial alternating walk that we have constructed so far. Now if P_j^* ends in a vertex $u \in T_m \setminus T'_m$ with a deletion edge, then we say $P_j = P_j^*$ and add a copy of u to T'_m. If P_j^* ends in a vertex u with an addition edge and there is an element corresponding to u in T'_r, then we say $P_j = P_j^*$ and we delete an element corresponding to u from T'_r. Else, we continue constructing this alternating walk by either adding or deleting an appropriate edge incident with u.
3. If both the above cases are not applicable, then, in fact, $j - 1 \geq \eta$; that is, we have constructed all the alternating walks in the solution. Let $B \subseteq E(G)$ and $A' \subseteq \overline{E(G)}$ be the set of deletion edges and addition edges present in $P_1, \ldots, P_{j-1}$. In fact, at this point $d_{G-B+A'}(v) = f(v)$ for all $v \in V(G)$, $T_m = T'_m$, $T'_r = \emptyset$, $X = \emptyset$, and $Y = \emptyset$. So all the degree constraints are satisfied. But to make the resulting graph *connected* we might need to do more editing. Note that any alternating closed walk will not disturb the degree of any vertex. It is only used for connectivity purpose. We start construction of P_j^* by guessing a deletion edge (u, v) in the closed walk P_j and adding both its end-vertices—u and v to X. After this we continue following this alternating walk until we hit u again with a deletion edge.

In essence, we are constructing our alternating walks greedily (that is, we stop whenever we have a chance to do so).

A formal definition of partial solution. For our partial solution we would like to use Lemma 4.4. That is, we would like to obtain a nice deletion set. However, to completely characterize a nice deletion set, we also need a proper deficiency map. Note that for any $Z \subseteq E(G)$, $T_r \subseteq S(G-Z,f)$. For each partial solution $B \cup A'$, the subset $T_r' \subseteq T_r$ represents the following: the current partial proper deficiency map does not have T_r' as its domain. This is the main reason we defined T_r' differently than T_m' and T_g'. In other words T_r' denotes that we still "need to add certain number of edges" on vertices belonging to Red. However, note that we have a vertex $v \in X$ and the only reason this vertex is in X is that $f(v) > d_{G-B}(v)$. Thus, when we consider $S(G-B,f) \setminus T_r'$, then there is an element corresponding to v present in it. And we cannot take care of this deficiency using the current partial map for which the corresponding edge set is A'. To circumvent this we remove the newly added deficiencies from our domain. Towards this we define $X_{B,f}$ as follows. Let X be a multiset of size 2 such that for all $u \in X$, $f(u) > d_{G-B}(u) + X(u) - 1$. Then,

$$X_{B,f} = \{u(i) \mid u \in X, f(u) - d_{G-B}(u) - X(u) + 1 \leq i \leq f(u) - d_{G-B}(u), i \in \mathbb{N}^+\}.$$

Note that $X_{B,f} \subseteq S(G-B,f)$. Similarly, when $Y \neq \emptyset$ we know that the last operation was an edge addition incident with a vertex $w \in Y$. Thus to have a proper deficiency map ψ' such that $E_{\psi'} = A'$, we need to add Y to the domain of ψ'. For some partial solution $B \cup A'$, it can happen that there exists $v \in \mathsf{Green}$ such that $d_{G-B}(v) \geq f(v)$ and $\overline{E}_G(v) \cap A' \neq \emptyset$. Thus to have a proper deficiency map ψ' such that $E_{\psi'} = A'$, we add the set T_g' corresponding to the partial solution $B \cup A'$ to the domain of ψ'. Thus for any partial solution $B \cup A'$ which are in an equivalence class characterized by $T_g' \subseteq T_g$, $T_r' \subseteq T_r$, X, and Y, we will have a proper deficiency map ψ' over $(S(G-B,f) \cup T_g' \cup Y) \setminus (T_r' \cup X_{B,f})$ such that $E_{\psi'} = A'$.

Now we formally define the notion of partial solutions. Given an instance (G,f,k) we define T_m, T_g, and T_r as described earlier. Also, recall that we have k_1, k_2, and Φ. For any $T_m' \subseteq T_m, T_g' \subseteq T_g, T_r' \subseteq T_r, k_1' \leq k_1, k_2' \leq k_2, i \leq k$, a multiset X containing elements from $V(G)$ and $Y \subseteq V(G)$ such that $|X| \leq 2, |Y| \leq 1, |X \cup Y| \leq 2$, and $X \cap Y = \emptyset$, we define a family $\mathcal{Q}(T_m', T_g', T_r', k_1', k_2', i, X, Y)$ of subsets of $\binom{V(G)}{2}$ as follows. For any $B \subseteq E(G)$ and $A \subseteq \overline{E(G)}$, $B \cup A \in \mathcal{Q}(T_m', T_g', T_r', k_1', k_2', i, X, Y)$ if the following conditions are met:

(a) $|E(G[\mathsf{Green}]) \cap B| = k_1', |B \setminus E(G[\mathsf{Green}])| = k_2'$, and $|B \cup A| = i$.

(b) For every $v \in \mathsf{Green}$, the number of edges in B which are incident with v is exactly equal to $T_m'(v) + T_g'(v) + X(v)$; that is, for all $v \in \mathsf{Green}$, $|B \cap E_G(v)| = T_m'(v) + T_g'(v) + X(v)$.

(c) $|X_{B,f}| = |X|$ and $X_{B,f} \subseteq S(G-B,f) \setminus T_r$.

(d) The graph $G-B$ has at most $k - k' + 1$ connected components.

(e) There is a proper deficiency map $\psi' : (S(G-B,f) \cup T_g' \cup Y) \setminus (T_r' \cup X_{B,f}) \to (S(G-B,f) \cup T_g' \cup Y) \setminus (T_r' \cup X_{B,f})$ such that $A = E_\psi$. Furthermore, for $w \in Y$, if $\psi'(w) = u(j)$ for some j, then $T_m(u) = T_m'(u)$.

In condition (e) we demanded the following: for $w \in Y$, if $\psi'(w) = u(j)$ for some j, then $T_m(u) = T_m'(u)$. We explain the reason for doing this. In our algorithm if $Y \neq \emptyset$, then the last operation is an addition operation with an edge incident with w.

If (u, w) is the edge added in the last operation, then in the proper deficiency map ψ, w maps to $u(j)$ for some j. The only reason we did not stop at u is because either $u \in \mathsf{Red}$ or $u \in \mathsf{Green}$ and $T'_m(u) = T_m(u)$. Thus, in some sense this condition helps us in knowing when the current alternating walk we are constructing will stop.

For $T'_m \subseteq T_m, T'_g \subseteq T_g, T'_r \subseteq T_r, k'_1 \leq k_1, k'_2 \leq k_2, i \leq k$, a multiset X containing elements from $V(G)$ and $Y \subseteq V(G)$, we say that the tuple $(T'_m T'_g, T'_r, k'_1, k'_2, i, X, Y)$ is a *valid* tuple if the following happens:

(1) $|X| \leq 2, |Y| \leq 1, |X \cup Y| \leq 2$, and $X \cap Y = \emptyset$.

(2) For $w \in Y$, $w(j) \notin T'_r$ for all j.

(3) If $u(j) \in T'_g$, then $u(j') \in T'_g$ for all $0 < j' < j$.

(4) For every $v \in X$, $T_m(v) = T'_m(v)$.

For the correctness of the algorithm, it is enough to focus on partial solutions defined over valid tuples. We have already explained that why the cardinality of $|X| \leq 2$ and $|Y| \leq 1$. Also note that if we have a partially constructed alternating (closed) walk, then its current end vertex will be either in X or Y. If we are constructing an alternating closed walk, then its starting vertex will also be in X. Because of this $|X \cup Y| \leq 2$. When both X and Y are nonempty, then we are constructing an alternating closed walk stating at a vertex $x \in X$ and at present it ends at a vertex $w \in Y$. If $x = w$, then we could have greedily completed this closed walk. Hence, $X \cap Y \neq \emptyset$. For any partial solution $B \cup A \in \mathcal{Q}(T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y)$, we have a proper deficiency map ψ over $(S(G-B, f) \cup T'_g \cup Y) \setminus (T'_r \cup X_{B,f})$. If $u(j) \in S(G-B, f)$, then $u(j') \in S(G-B, f)$ for all $j' \leq j$. Since T'_g also accounts for the number of edges deleted from each vertex in Green (along with T'_m and X), the condition (3) of the valid tuple is a sanity check. Conditions (2) and (4) are another set of sanity checks which we have already explained.

Now we prove that, in fact, $\mathcal{Q}(T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y)$ is "a correct notion of partial solutions."

LEMMA 4.5. *Let (G, f, k) be a* YES *instance of* EECG *with a solution $D \cup A$ such that $D \subseteq E(G), A \subseteq \overline{E(G)}$, $|D \cap E(G[\mathsf{Green}])| = k_1, |D \setminus E(G[\mathsf{Green}])| = k_2$, $k_1 + k_2 = k'$, and $|D \cap E_G(v)| = \Phi(v)$ for all $v \in \mathsf{Green}$. Let ψ be a proper deficiency map over $S(G-D, f)$ such that $E_\psi = A$. Then, for each $i \leq k$, there exists $D' \cup A' \subseteq D \cup A$ and a valid tuple $(T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y)$ such that $D' \cup A' \in \mathcal{Q}(T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y)$ and there is a proper deficiency map ψ' over $R = (S(G-D', f) \cup T'_g \cup Y) \setminus (T'_r \cup X_{D',f})$ with the property that $E_{\psi'} = A'$*

Proof. We prove the lemma by induction on i. For $i = 0$ we set $D', A' = \emptyset$ and so $D' \cup A' \in \mathcal{Q}(\emptyset, \emptyset, T_r, 0, 0, 0, \emptyset, \emptyset)$. It is obvious to see that $D' \cup A'$ satisfies the conditions (a), (b), (c), and (e) of being in the family $\mathcal{Q}(\emptyset, \emptyset, T_r, 0, 0, 0, \emptyset, \emptyset)$. Since $D \cup A$ is a solution of EECG, G has at most $k - k' + 1$ connected components, and hence $D' \cup A'$ satisfies the condition (d) of being in the family $\mathcal{Q}(\emptyset, \emptyset, T_r, 0, 0, 0, \emptyset, \emptyset)$. Assume that the statement is true for $i - 1$; that is, there exists $D' \cup A' \subseteq D \cup A$ and a valid tuple $(T'_m, T'_g, T'_r, k'_1, k'_2, i-1, X, Y)$ such that $D' \cup A' \in \mathcal{Q}(T'_m, T'_g, T'_r, k'_1, k'_2, i-1, X, Y)$ and there is a proper deficiency map ψ' over $R = (S(G - D', f) \cup T'_g \cup Y) \setminus (T'_r \cup X_{D',f})$ with the property that $E_{\psi'} = A'$. We need to show that the statement holds for i.

Case 1: $\boldsymbol{X, Y = \emptyset}$. Since $i - 1 < k$, we have that $D' \neq D$ or $A' \neq A$.

Subcase (i): $D' \neq D$. Let $e = (x, y) \in D \setminus D'$. Let $D'' = D' \cup \{e\}$ and let $Y' = \emptyset$. Now we explain how to construct X'. The set X' is a subset of $\{x, y\}$. If $x \in \mathsf{Green}$ and $x \notin T_m \setminus T'_m$, then we add x to X'. If $x \in \mathsf{Red}$, then we add x to X'. A similar case holds for y as well. We set $X' = \{z \in \{x, y\} \mid z \notin T_m \setminus T'_m\}$. Note that if $z \in \{x, y\}$ is a Red vertex, then $z \notin T_m \setminus T'_m$. The following claim follows from the definition of X'.

CLAIM 2. *Let $z \in \{x, y\} \cap \mathsf{Green}$. Then $z \notin X'$ if and only if $z \in T_m \setminus T'_m$.*

Let $T''_m = T'_m \cup (\{x, y\} \cap (T_m \setminus T'_m))$; that is, we add those elements from $\{x, y\}$ to T'_m that appear in $T_m \setminus T'_m$. Now we define $k''_1 = k'_1 + 1$ and $k''_2 = k'_2$ if $e \in E(G[\mathsf{Green}])$, otherwise $k''_1 = k'_1$ and $k''_2 = k'_2 + 1$. Now we claim that $D'' \cup A' \in \mathcal{Q}(T''_m, T'_g, T'_r, k''_1, k''_2, i, X', Y')$ and $(T''_m, T'_g, T'_r, k''_1, k''_2, i, X', Y')$ is a valid tuple. Since $|X'| \leq 2$, $Y' = \emptyset$, and $(T'_m, T'_g, T'_r, k'_1, k'_2, i-1, X, Y)$ is a valid tuple, $(T''_m, T'_g, T'_r, k''_1, k''_2, i, X', Y')$ satisfies properties (1), (2), and (3) of a valid tuple. Due to Claim 2, $(T''_m, T'_g, T'_r, k''_1, k''_2, i, X', Y')$ satisfies property (4) of a valid tuple. Since $D' \cup A' \in \mathcal{Q}(T'_m, T'_g, T'_r, k'_1, k'_2, i-1, \emptyset, \emptyset)$ and $D'' = D' \cup \{(x, y)\}$, the subset $D'' \cup A'$, satisfies the following conditions:

(a) $|D'' \cap E(G[\mathsf{Green}])| = k''_1$, $|D'' \setminus E(G[\mathsf{Green}])| = k''_2$, and $|D'' \cup A'| = i$.

(b) Due to Claim 2 and the definition of T''_m, we have that for any $v \in \mathsf{Green}$, $|D'' \cap E_G(v)| = T''_m(v) + X'(v) + T'_g(v)$.

(c) Since for any $v \in X'$, $d_{G-D''}(v) < f(v)$, we have that $|X'| = |X'_{D'',f}|$ and $X'_{D'',f} \subseteq S(G - D'', f) \setminus T_r$.

(d) Since $D \cup A$ is a solution of EECG, by Lemma 4.4, we have that $G - D$ has at most $k - k' + 1$ connected components. This implies that $G - D''$ has at most $k - k' + 1$ connected components.

(e) Since $S(G - D'', f) \cup T'_g = S(G - D', f) \cup T'_g \cup X'_{D'',f}$ and $(S(G - D', f) \cup T'_g) \cap X'_{D'',f} = \emptyset$, we have that $(S(G - D', f) \cup T'_g) \setminus T'_r = (S(G - D'', f) \cup T'_g) \setminus (T'_r \cup X'_{D'',f})$. This implies that ψ' is a proper deficiency map over $(S(G - D'', f) \cup T'_g) \setminus (T'_r \cup X'_{D'',f})$.

Thus we conclude that $D'' \cup A' \in \mathcal{Q}(T''_m, T'_g, T'_r, k''_1, k''_2, i, X', Y')$.

Subcase (ii): $D' = D$. In this subcase we have that $A' \neq A$. Let $(x, y) \in A \setminus A'$. Let $A'' = A' \cup \{(x, y)\}$. Note that $E_\psi = A$ and $E_{\psi'} = A'$. Since $D' = D$, for all $v \in \mathsf{Green}$ the number of edges in D' which are incident with v is equal to $\Phi(v)$. Also, note that $T_m(v) + T_g(v) = \Phi(v)$. This implies that $T'_m = T_m$ and $T'_g = T_g$. Since ψ is a proper deficiency map over $S(G - D, f) = S(G - D, f) \cup T_g$, ψ' is a proper deficiency map over $(S(G - D', f) \cup T'_g) \setminus T'_r$ and $(x, y) \in E_\psi \setminus E_{\psi'}$, there exists j, j' such that $x(j), y(j') \in T'_r$. Now we claim that $D' \cup A'' \in \mathcal{Q}(T'_m, T'_g, T'_r \setminus \{x(j), y(j')\}, k'_1, k'_2, i, \emptyset, \emptyset)$. Since $D' \cup A' \in \mathcal{Q}(T'_m, T'_g, T'_r, k'_1, k'_2, i-1, \emptyset, \emptyset)$, $D' \cup A''$ satisfies conditions (a), (b), (c), and (d) of being in the family $\mathcal{Q}(T'_m, T'_g, T'_r \setminus \{x(j), y(j')\}, k'_1, k'_2, i, \emptyset, \emptyset)$. Now we need to show that $D' \cup A''$ satisfies condition (e). Consider the bijection ψ'' defined over $(S(G - D', f) \cup T'_g) \setminus (T'_r \setminus \{x(j), y(j')\})$ as follows:

$$\psi''(q) = \begin{cases} y(j') & \text{if } q = x(j), \\ x(j) & \text{if } q = y(j'), \\ \psi'(q) & \text{otherwise.} \end{cases}$$

Note that $E_{\psi''} = A'' \subseteq A$. Since ψ' is a proper deficiency map, $\psi''(x(j)) = y(j')$, $\psi''(y(j')) = x(j)$, $E_{\psi''}$ is not a multiset and $E_{\psi''} \cap E(G) = \emptyset$, we have that ψ'' is a proper deficiency map. It is easy to see that $(T'_m, T'_g, T'_r, k'_1, k'_2, i-1, \emptyset, \emptyset)$ is a valid tuple.

Case 2: $\boldsymbol{X \neq \emptyset, Y = \emptyset}$. Let $v \in X$ and let j be the smallest integer such that $v(j) \in X_{D',f}$. Since $X_{D',f} \subseteq S(G-D',f) \setminus T_r$, we have that $v(j) \in S(G-D',f) \subseteq S(G-D,f)$. Also, since ψ' is a proper deficiency map over $(S(G-D',f) \cup T'_g) \setminus (T'_r \cup X_{D',f})$ and $E_{\psi'} \subseteq E_\psi$, there exists $b \in V(G)$ such that $(v,b) \notin E_{\psi'}$ and $(v,b) \in E_\psi$. Let $A'' = A' \cup \{(v,b)\}$.

Subcase (i): $b \in X$. In this subcase the set $X' = \emptyset$. Let $j' = f(b) - d_{G-D'}(b)$. Note that $\{v(j), b(j')\} = X_{D',f} \subseteq S(G-D',f) \setminus T_r$. Let $T''_g = T'_g \cup (\{v(j), b(j')\} \cap T_g)$. If $v \in \mathsf{Green}$, then we know that $|E_G(v) \cap D'| = T'_m(v) + T'_g(v) + X(v)$ and $X(v) = 1$. Since the new set $X' = \emptyset$, to keep track of the cardinality $|E_G(v) \cap D'|$ we include $v(j)$ to T''_g. Claim 3 ensures that $v(j)$ does not belong to T'_g. The similar arguments hold for b as well.

Claim 3. *If $v \in \mathsf{Green}$, then $v(j) \notin T'_g$ and $v(j-1) \in T'_g$.*

Proof. Since $\{v,b\} = X$, $j = f(v) - d_{G-D'}(v)$. This implies that

$$\begin{aligned} |E_G(v) \cap D'| &= j + d_G(v) - f(v) \\ &= j + T_m(v). \end{aligned} \tag{1}$$

We also know that, by the property (b) of partial solutions,

$$\begin{aligned} |E_G(v) \cap D'| &= T'_m(v) + T'_g(v) + X(v) \\ &= T_m(v) + T'_g(v) + X(v) \qquad \text{(because } v \in X). \end{aligned} \tag{2}$$

Equations (1) and (2) imply that $j = T'_g(v) + X(v)$. This implies $T'_g(v) = j-1$ because $|X(v)| = 1$. Thus we can conclude that $v(j) \notin T'_g$ and $v(j-1) \in T'_g$. ∎

Similarly, we can prove the following claim.

Claim 4. *If $b \in \mathsf{Green}$, then $b(j') \notin T'_g$ and $b(j'-1) \in T'_g$.*

Now we claim that $D' \cup A'' \in \mathcal{Q}(T'_m, T''_g, T'_r, k'_1, k'_2, i, \emptyset, \emptyset)$. Since $D' \cup A' \in \mathcal{Q}(T'_m, T'_g, T'_r, k'_1, k'_2, i-1, X, \emptyset)$ and by Claims 3 and 4, we can conclude that $D' \cup A''$ satisfies conditions (a), (b), (c), and (d) of being in the family $\mathcal{Q}(T'_m, T''_g, T'_r, k'_1, k'_2, i, \emptyset, \emptyset)$. Now we need to show that $D' \cup A''$ satisfies condition (e). Consider the bijection ψ'' over $(S(G-D'), f) \cup T''_g) \setminus T'_r$ as follows:

$$\psi''(q) = \begin{cases} b(j') & \text{if } q = v(j), \\ v(j) & \text{if } q = b(j'), \\ \psi'(q) & \text{otherwise,} \end{cases}$$

Note that $E_{\psi''} = A''$. Since ψ' is a proper deficiency map, $\psi''(u(j)) = v(j')$, $\psi''(v(j')) = u(j)$, $E_{\psi''}$ is not a multiset and $E_{\psi''} \cap E(G) = \emptyset$, we have that ψ'' is a proper deficiency map. It is easy to see that $(T'_m, T''_g, T'_r, k'_1, k'_2, i, \emptyset, \emptyset)$ satisfies properties (1), (2) and (4) of a valid tuple. Claims 3 and 4 imply that $(T'_m, T''_g, T'_r, k'_1, k'_2, i, \emptyset, \emptyset)$ satisfies property (3) of a valid tuple.

Subcase (ii): *$b \notin X$ and $b(j') \in T'_r$ for some j'.* Let $T''_g = T'_g \cup (\{v(j)\} \cap T_g)$.

CLAIM 5. *If $v \in \mathsf{Green}$, then $v(j) \notin T'_g$ and $v(j-1) \in T'_g$.*

Proof. If $X(v) = 1$, then the proof is the same as that of Claim 3. Suppose $X(v) = 2$. Since $\{v, v\} = X$, $j = f(v) - d_{G-D'}(v) - 1$. This implies that

$$\begin{aligned} |E_G(v) \cap D'| &= j + 1 + d_G(v) - f(v) \\ &= j + 1 + T_m(v). \end{aligned} \tag{3}$$

We also know that, by the property (b) of partial solutions,

$$\begin{aligned} |E_G(v) \cap D'| &= T'_m(v) + T'_g(v) + X(v) \\ &= T_m(v) + T'_g(v) + 2 \qquad \text{(because } v \in X\text{)}. \end{aligned} \tag{4}$$

Equations (3) and (4) imply that $j - 1 = T'_g(v)$. Thus we can conclude that $v(j) \notin T'_g$ and $v(j-1) \in T'_g$. ∎

Now we claim that $D' \cup A'' \in \mathcal{Q}(T'_m, T''_g, T'_r \setminus \{b(j')\}, k'_1, k'_2, i, X', \emptyset)$ where $X' = X \setminus \{v\}$. Since $D' \cup A' \in \mathcal{Q}(T'_m, T'_g, T'_r, k'_1, k'_2, i-1, X, \emptyset)$ and by Claim 5, we can conclude that $D' \cup A''$ satisfies conditions (a), (b), (c), and (d) of being in the family $\mathcal{Q}(T'_m, T''_g, T'_r \setminus \{b(j')\}, k'_1, k'_2, i, X', \emptyset)$. Now we need to show that $D' \cup A''$ satisfies condition (e). Consider the bijection ψ'' over $(S(G-D', f) \cup T''_g) \setminus ((T'_r \setminus \{b(j')\}) \cup X'_{D',f})$ as follows.

$$\psi''(q) = \begin{cases} b(j') & \text{if } q = v(j), \\ v(j) & \text{if } q = b(j'), \\ \psi'(q) & \text{otherwise,} \end{cases}$$

Note that $E_{\psi''} = A'' \subseteq A$. Since ψ' is a proper deficiency map, $\psi''(v(j)) = b(j')$, $\psi''(b(j')) = v(j)$, $E_{\psi''}$ is not a multiset and $E_{\psi''} \cap E(G) = \emptyset$, we have that ψ'' is a proper deficiency map. Since $X' \subseteq X$ and $(T'_m, T'_g, T'_r, k'_1, k'_2, i-1, X, \emptyset)$ is a valid tuple, we have that $(T'_m, T''_g, T'_r \setminus \{b(j')\}, k'_1, k'_2, i, X', \emptyset)$ satisfies properties (1), (2) and (4) of a valid tuple. Claim 5 implies that $(T'_m, T''_g, T'_r \setminus \{b(j')\}, k'_1, k'_2, i, X', \emptyset)$ satisfies property (3).

Subcase (iii): *$b \notin X$ and $b(j') \notin T'_r$ for all j'.* Let $T''_g = T'_g \cup (\{v(j)\} \cap T_g)$.

CLAIM 6. *If $v \in \mathsf{Green}$, then $v(j) \notin T'_g$ and $v(j-1) \in T'_g$.*

The proof of Claim 6 is the same as that of Claim 5. Now we claim that $D' \cup A'' \in \mathcal{Q}(T'_m, T''_g, T'_r, k'_1, k'_2, i, X', Y')$, where $X' = X \setminus \{v\}$ and $Y' = \{b\}$. Since $D' \cup A' \in \mathcal{Q}(T'_m, T'_g, T'_r, k'_1, k'_2, i-1, X, \emptyset)$ and $X' \subseteq X$, and by Claim 6, we can conclude that $D' \cup A''$ satisfies conditions (a), (b), (c), and (d) of being in the family $\mathcal{Q}(T'_m, T''_g, T'_r, k'_1, k'_2, i, X', Y')$. Now we need to show that $D' \cup A''$ satisfies condition

(e). Consider the bijection ψ'' over $(S(G-D',f)\cup T''_g\cup\{b\})\setminus(T'_r\cup X'_{D',f})$ defined as follows:

$$\psi''(q)=\begin{cases} b & \text{if } q=v(j),\\ v(j) & \text{if } q=b,\\ \psi'(q) & \text{otherwise.}\end{cases}$$

Note that $E_{\psi''}=E_{\psi'}\cup\{e\}=A''$. Since ψ' is a proper deficiency map, $\psi''(v(j))=b$, $\psi''(b)=v(j)$, $E_{\psi''}$ is not a multiset, and $E_{\psi''}\cap E(G)=\emptyset$, we have that ψ'' is a proper deficiency map. Also note that $\psi''(b)=v(j)$ and $T_m(v)=T'_m(v)$, because $v\in X$ and $(T'_m,T'_g,T'_r,k'_1,k'_2,i-1,X,\emptyset)$ is a valid tuple. Since $X'\subset X$, $Y'=\{b\}$, $b(j')\notin T'_r$ for all j', and $(T'_m,T'_g,T'_r,k'_1,k'_2,i-1,X,\emptyset)$ is a valid tuple, we have that $(T'_m,T''_g,T'_r,k'_1,k'_2,i,X',Y')$ satisfies properties (1), (2), and (4) of a valid tuple. Claim 5 implies that $(T'_m,T''_g,T'_r\setminus\{b(j')\},k'_1,k'_2,i,X',\emptyset)$ satisfies property (3).

Case 3: $Y\neq\emptyset$. Let $Y=\{w\}$. Since $(T'_m,T'_g,T'_r,k'_1,k'_2,i-1,X,Y)$ is a valid tuple and $|Y|=1$, we have that $|X|\le 1$. We claim that there exists $x\in V(G)$ such that $(w,x)\in D\setminus D'$. Suppose not, then $d_{G-D'}(w)=d_{G-D}(w)$. This implies that $d_{G-D'+A'}(w)=d_{G-D+A'}(w)$. We know that ψ' is a proper deficiency map over $(S(G-D',f)\cup T'_g\cup\{w\})\setminus(T'_r\cup X_{D',f})$, $E_{\psi'}=A'$, and $w(i')\notin T'_r\cup X_{D',f}$ for all i'. This implies that $d_{G-D'+A'}(w)\ge f(w)+1$ and hence $d_{G-D+A}(w)>f(w)$, contradicting the fact that $D\cup A$ is a solution to EECG. Thus we know that there exists x such that $(w,x)\in D\setminus D'$. Let $D''=D'\cup\{(w,x)\}$ and let $X'=X\cup\{z|z=x,x\notin T_m\setminus T'_m\}$. Note that $|X'|\le 2$, because $|X|\le 1$. Let $T''_m=T'_m\cup(\{x\}\cap(T_m\setminus T'_m))$. The following claim follows from the definition of X'.

CLAIM 7. *Let $x\in$ Green. Then $x\notin X'\setminus X$ if and only if $x\in T_m\setminus T'_m$.*

Subcase (i): $w\in$ Green. Let $k''_1=k'_1+1, k''_2=k'_2$ if $(w,x)\in E(G[\text{Green}])$, otherwise $k''_1=k'_1, k''_2=k'_2+1$. We claim that there exists $j\in\mathbb{N}^+$ such that $w(j)\in T_g\setminus T'_g$. Suppose not. Since ψ' is a proper deficiency map over $(S(G-D'))\cup T'_g\cup\{w\}\setminus(T'_r\cup X_{D',f})$ such that $E'_\psi=A'\subseteq A$ and $w(j)\notin T_g\setminus T'_g$ for all j, we can conclude that the number of edges in A' which are incident with w is at least $1+|\{w(j')\mid j'\in\mathbb{N}^+, w(j')\in T_g\}|=1+\Phi(w)-(d_G(w)-f(w))$. This implies that $d_{G-D+A}(w)\ge d_{G-D+A'}(w)\ge d_G(w)-|E_G(v)\cap D|+1+\Phi(w)-(d_G(w)-f(w))=d_G(w)-\Phi(w)+1+\Phi(w)-(d_G(w)-f(w))>f(w)$, which is a contradiction to the fact that $D\cup A$ is a solution of EECG. Without loss of generality let j be the smallest integer such that $w(j)\in T_g\setminus T'_g$. Now we show that $D''\cup A'\in\mathcal{Q}(T''_m,T''_g,T'_r,k''_1,k''_2,i,X',\emptyset)$, where $T''_g=T'_g\cup\{w(j)\}$. Since $D'\cup A'\in\mathcal{Q}(T'_m,T'_g,T'_r,k'_1,k'_2,i-1,X,Y)$ and $D''=D'\cup\{(w,x)\}$, the subset $D''\cup A'$ satisfies the following conditions:

(a) $|D''\cap E(G[\text{Green}])|=k''_1$, $|D''\setminus E(G[\text{Green}])|=k''_2$, and $|D''\cup A'|=i$.

(b) For any $v\in$ Green,

$$\begin{aligned}|D''\cap E_G(v)|&=\begin{cases} T'_m(v)+T'_g(v)+X(v)+1 & \text{if } v\in\{w,x\},\\ T'_m(v)+T'_g(v)+X(v) & \text{if } v\notin\{w,x\}\end{cases}\\ &=\begin{cases} T'_m(v)+T''_g(v)+X(v) & \text{if } v=w \text{ (by definition of } T''_g),\\ T''_m(v)+T'_g(v)+X'(v) & \text{if } v=x \text{ (due to Claim 7)},\\ T'_m(v)+T'_g(v)+X(v) & \text{if } v\notin\{w,x\}\end{cases}\\ &=T''_m(v)+T''_g(v)+X'(v).\end{aligned}$$

(c) $|X'| = 1+|\{z|z = x, x \notin T_m \setminus T'_m\}| = |X'_{D'',f}|$ and $X'_{D'',f} \subseteq S(G-D'', f)\setminus T_r$.

(d) Since $D \cup A$ is a solution of EECG, by Lemma 4.4 we have that $G - D$ has at most $k - k' + 1$ connected components. This implies that $G - D''$ has at most $k - k' + 1$ connected components.

Now we need to show that $D'' \cup A'$ satisfies the condition (e). Consider the bijection ψ'' over $(S(G - D'') \cup T''_g) \setminus (T'_r \cup X'_{D'',f})$ as follows:

$$\psi''(q) = \begin{cases} \psi'(w) & \text{if } q = w(j), \\ w(j) & \text{if } q = \psi'(w), \\ \psi'(q) & \text{otherwise,} \end{cases}$$

The function ψ'' is a proper deficiency map such that $E_{\psi''} = A'$. Since the tuple $(T'_m, T'_g, T'_r, k'_1, k'_2, i-1, X, Y)$ is a valid tuple, $|X'| \leq 2$, and j is the smallest integer such that $w(j) \in T_g \setminus T'_g$, we have that $(T''_m, T''_g, T'_r, k''_1, k''_2, i, X', \emptyset)$ satisfies properties (1), (2), and (3) of a valid tuple. Due to Claim 7, $(T''_m, T''_g, T'_r, k''_1, k''_2, i, X', \emptyset)$ satisfies property (4) of a valid tuple.

Subcase (ii): $w \in \mathsf{Red}$. Here, we claim that $D'' \cup A' \in \mathcal{Q}(T''_m, T'_g, T'_r, k'_1, k'_2 + 1, i, X', \emptyset)$. Let $j = f(w) - d_{G-D''}(w)$. Since $D' \cup A' \in \mathcal{Q}(T'_m, T'_g, T'_r, k'_1, k'_2, i-1, X, Y)$ and $D'' = D' \cup \{(w, x)\}$, the subset $D'' \cup A'$ satisfies the following conditions:

(a) $|D'' \cap E(G[\mathsf{Green}])| = k'_1$, $|D'' \setminus E(G[\mathsf{Green}])| = k'_2 + 1$, and $|D'' \cup A'| = i$.

(b) For any $v \in \mathsf{Green}$,

$$\begin{aligned} |D'' \cap E_G(v)| &= \begin{cases} T'_m(v) + T'_g(v) + X(v) + 1 & \text{if } v = x, \\ T'_m(v) + T'_g(v) + X(v) & \text{if } v \neq x \end{cases} \\ &= T''_m(v) + T'_g(v) + X'(v) \qquad \text{(due to Claim 7).} \end{aligned}$$

(c) $|X'| = 1+|\{z|z = x, x \notin T_m \setminus T'_m\}| = |X'_{D'',f}|$ and $X'_{D'',f} \subseteq S(G-D'', f)\setminus T_r$.

(d) $G - D''$ has at most $k - k' + 1$ connected components.

Now we need to show that $D'' \cup A'$ satisfies the condition (e). Consider the bijection ψ'' over $(S(G - D'') \cup T'_g) \setminus (T'_r \cup X'_{D'',f})$ as follows:

$$\psi''(q) = \begin{cases} \psi'(w) & \text{if } q = w(j), \\ w(j) & \text{if } q = \psi'(w), \\ \psi'(q) & \text{otherwise.} \end{cases}$$

The function ψ'' is a proper deficiency map such that $E_{\psi''} = A'$. Since the tuple $(T'_m, T'_g, T'_r, k'_1, k'_2, i-1, X, Y)$ is a valid tuple, $|X'| \leq 2$, we have that $(T''_m, T'_g, T'_r, k'_1, k'_2 + 1, i, X', \emptyset)$ satisfies properties (1), (2) and (3) of a valid tuple. Due to Claim 7, $(T''_m, T'_g, T'_r, k'_1, k'_2 + 1, i, X', \emptyset)$ satisfies property (4) of a valid tuple.

Our algorithm is based on DP. It keeps a table entry $\mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y]$ for each valid tuple $(T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y)$. The idea is to store a subset of $\mathcal{Q}(T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y)$ in the DP table entry $\mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y]$ which is sufficient to maintain the correctness of the algorithm. Next, we write the recurrence relation for $\mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y]$ and prove its correctness. Towards that consider the following $\bullet$ and $\circ$ operations defined as follows. For any family $\mathcal{S}$ of subsets of $\binom{V(G)}{2}$, $e \in E(G)$, and $e' \in \overline{E(G)}$,

$$\begin{aligned}\mathcal{S} \bullet e &= \{B \cup A \cup \{e\} \mid A \cup B \in \mathcal{S}, B \subseteq E(G) \setminus \{e\}, A \subseteq \overline{E(G)},\\ &\qquad G - (B \cup \{e\}) \text{ has at most } k - k' + 1 \text{ connected components}\},\\ \mathcal{S} \circ e' &= \{B \cup A \cup \{e'\} \mid A \cup B \in \mathcal{S}, B \subseteq E(G), A \subseteq \overline{E(G)} \setminus \{e'\}\}.\end{aligned}$$

Now we write the recurrence relation. For $i = 0$, we have the following base cases:

$$\mathcal{D}[T'_m, T'_g, T'_r, 0, 0, 0, X, Y] := \begin{cases} \{\emptyset\} & \text{if } T'_m, T'_g, X, Y = \emptyset, \text{ and } T'_r = T_r, \\ \emptyset & \text{otherwise.} \end{cases} \tag{5}$$

For any tuple $(T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y)$ which is not a valid tuple, we set the entry $D[T'_g, T'_r, k'_1, k'_2, i, X, Y] = \emptyset$. Now we describe how to compute DP table entry for $D[T'_m, T'_g, T'_r, k'_1, k'_2, i{+}1, X, Y]$ for a valid tuple $(T'_m, T'_g, T'_r, k'_1, k'_2, i{+}1, X, Y)$ using the previously calculated table entries. We write the following recurrence by following Lemma 4.5; that is, we see which all cases in Lemma 4.5 will lead to the current table entry and for the current table entry we just take the union of previously calculated table entries corresponding to these cases.

Case 1: $X = \emptyset$ and $Y = \emptyset$.

$$\begin{aligned}&\mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i+1, X, Y]\\
&:= \left(\bigcup_{\substack{(x,y) \in E(G)\\ x,y \in T'_m}} \mathcal{D}[T'_m \setminus \{x,y\}, T'_g, T'_r, k'_1 - 1, k'_2, i, \emptyset, \emptyset] \bullet \{(x,y)\} \right)\\
&\bigcup \left(\bigcup_{\substack{(x,y) \in \overline{E(G)}\\ \exists j,j' x(j), y(j') \in T_r \setminus T'_r}} \mathcal{D}[T'_m, T'_g, T'_r \cup \{x(j), y(j')\}, k'_1, k'_2, i, \emptyset, \emptyset] \circ \{(x,y)\} \right)\\
&\bigcup \left(\bigcup_{\substack{(x,y) \in \overline{E(G)}\\ j = T'_g(x), j' = T'_g(y)}} \mathcal{D}[T'_m, T'_g \setminus \{x(j), y(j')\}, T'_r, k'_1, k'_2, i, \{x,y\}, \emptyset] \circ \{(x,y)\} \right)\\
&\bigcup \left(\bigcup_{\substack{(y,x) \in E(G)\\ j = T'_g(y), x \in T'_m}} \mathcal{D}[T'_m \setminus \{x\}, T'_g \setminus \{y(j)\}, T'_r, k'_1 - 1, k'_2, i, \emptyset, \{y\}] \bullet \{(y,x)\} \right)\\
&\bigcup \left(\bigcup_{\substack{(y,x) \in E(G)\\ x \in T'_m, y \in \mathsf{Red}}} \mathcal{D}[T'_m \setminus \{x\}, T'_g, T'_r, k'_1, k'_2 - 1, i, \emptyset, \{y\}] \bullet \{(y,x)\} \right)\\
&\bigcup \left(\bigcup_{\substack{(x,y) \in \overline{E(G)}\\ \exists j, y(j) \in T_r \setminus T'_r}} \mathcal{D}[T'_m, T'_g, T'_r \cup \{y(j)\}, k'_1, k'_2, i, \{x\}, \emptyset] \circ \{(x,y)\} \right).\end{aligned} \tag{6}$$

Case 2: $x \in X \neq \emptyset$ and $Y = \emptyset$.

$$
\begin{aligned}
&\mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i+1, X, Y] \qquad (7)\\
&\quad := \Bigg(\bigcup_{\substack{(x,y)\in E(G[\mathsf{Green}])\\ y\in X, T'_m(x)=T_m(x)\\ T'_m(y)=T_m(y)}} \mathcal{D}[T'_m, T'_g, T'_r, k'_1-1, k'_2, i, \emptyset, \emptyset] \bullet \{(x,y)\} \Bigg)\\
&\bigcup \Bigg(\bigcup_{\substack{(x,y)\in E(G)\\ y\in X, x\in\mathsf{Green}, y\in\mathsf{Red}\\ T'_m(x)=T_m(x)}} \mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2-1, i, \emptyset, \emptyset] \bullet \{(x,y)\} \Bigg)\\
&\bigcup \Bigg(\bigcup_{\substack{(x,y)\in E(G[\mathsf{Green}])\\ y\notin X, y\in T'_m\\ T'_m(x)=T_m(x)}} \mathcal{D}[T'_m \setminus \{y\}, T'_g, T'_r, k'_1-1, k'_2, i, X\setminus\{x\}, \emptyset] \bullet \{(x,y)\} \Bigg)\\
&\bigcup \Bigg(\bigcup_{\substack{(x,y)\in E(G)\\ x\in\mathsf{Red}, y\notin X, y\in T'_m}} \mathcal{D}[T'_m \setminus \{y\}, T'_g, T'_r, k'_1, k'_2-1, i, X\setminus\{x\}, \emptyset] \bullet \{(x,y)\} \Bigg)\\
&\bigcup \Bigg(\bigcup_{\substack{(y,z)\in \overline{E(G)}, j=T'_g(z)\\ \exists j': y(j')\in T_r\setminus T'_r}} \mathcal{D}[T'_m, T'_g \setminus \{z(j)\}, T'_r \cup \{y(j')\}, k'_1, k'_2, i, X\setminus\{x\}, \emptyset] \circ \{(y,z)\} \Bigg)\\
&\bigcup \Bigg(\bigcup_{\substack{(y,z)\in E(G)\\ j=T'_g(y), z\in T'_m}} \mathcal{D}[T'_m \setminus \{z\}, T'_g \setminus \{y(j)\}, T'_r, k'_1-1, k'_2, i, X, \{y\}] \bullet \{(y,x)\} \Bigg)\\
&\bigcup \Bigg(\bigcup_{\substack{(y,x)\in E(G[\mathsf{Green}])\\ x\notin T'_m, j=T'_g(y)}} \mathcal{D}[T'_m, T'_g \setminus \{y(j)\}, T'_r, k'_1-1, k'_2, i, X\setminus\{x\}, \{y\}] \bullet \{(y,x)\} \Bigg)\\
&\bigcup \Bigg(\bigcup_{\substack{(y,x)\in E(G)\\ x\in\mathsf{Red}, j=T'_g(y)}} \mathcal{D}[T'_m, T'_g \setminus \{y(j)\}, T'_r, k'_1-1, k'_2, i, X\setminus\{x\}, \{y\}] \bullet \{(y,x)\} \Bigg)\\
&\bigcup \Bigg(\bigcup_{\substack{(y,z)\in E(G), z\notin X\\ y\in\mathsf{Red}, z\in T'_m}} \mathcal{D}[T'_m \setminus \{z\}, T'_g, T'_r, k'_1, k'_2-1, i, X, \{y\}] \bullet \{(y,x)\} \Bigg)\\
&\bigcup \Bigg(\bigcup_{\substack{(y,x)\in E(G)\\ y\in\mathsf{Red}, x\notin T'_m}} \mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2-1, i, X\setminus\{x\}, \{y\}] \bullet \{(y,x)\} \Bigg).
\end{aligned}
$$

Case 3: $y \in Y \neq \emptyset$**.**

$$\mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i+1, X, Y] \tag{8}$$
$$:= \left(\bigcup_{\substack{(x,y)\in\overline{E(G)}, j=T'_g(x) \\ X'=X\cup\{x\}, T'_r(y)=0}} \mathcal{D}[T'_m, T'_g \setminus \{x(j)\}, T'_r, k'_1, k'_2, i, X', \emptyset] \circ \{(x,y)\} \right).$$

The algorithm computes DP table entries $\mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y]$ for all valid tuples $(T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y)$, and if there exists $D\cup A \in \mathcal{D}[T_m, T_g, \emptyset, k_1, k_2, k, \emptyset, \emptyset]$ such that $D \cup A$ is a solution to EECG, then outputs Yes, otherwise outputs No. Since the size of $\mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y]$ can potentially be $n^{\mathcal{O}(i)}$, this algorithm takes time $n^{\mathcal{O}(k)}$. Now we prove the correctness of the algorithm.

Correctness. If the algorithm outputs Yes, then there exists $D \cup A$ which is a solution to EECG. Now we need to show that if the input instance is a Yes instance, then the algorithm will always output Yes. Lemma 4.7 achieves this. The following lemma is useful for Lemma 4.7.

Lemma 4.6. *$\mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y] \subseteq \mathcal{Q}(T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y)$, for any valid tuple $(T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y)$.*

Proof. This lemma can easily be proved using induction on i. ∎

The next lemma is very similar to Lemma 4.5 and we use some of the arguments used there (for example, in showing a particular tuple to be valid) directly in our proof.

Lemma 4.7. *Let (G, f, k) be a* Yes *instance of* EECG *with a solution $D\cup A$ such that $D \subseteq E(G), A \subseteq \overline{E(G)}$, $|D \cap E(G[\mathsf{Green}])| = k_1, |D \setminus E(G[\mathsf{Green}])| = k_2$, $k_1 + k_2 = k'$ and $|D \cap E_G(v)| = \Phi(v)$ for all $v \in \mathsf{Green}$. Let ψ be a proper deficiency map over $S(G-D, f)$ such that $E_\psi = A$. Then for each $i \leq k$, there exists $D'\cup A' \subseteq D\cup A$ and a valid tuple $(T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y)$ such that $D' \cup A' \in \mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y]$ and there is a proper deficiency map ψ' over $R = (S(G-D', f)\cup T'_g \cup Y) \setminus (T'_r \cup X_{D',f})$ with the property that $E_{\psi'} = A'$. Moreover, $D \cup A \in \mathcal{D}[T_m, T_g, \emptyset, k_1, k_2, k, \emptyset, \emptyset]$.*

Proof. We prove the lemma by induction on i and its proof is very much similar to the proof of Lemma 4.5. For $i = 0$ we set $D', A' = \emptyset$ and by definition, $\emptyset \in \mathcal{D}[\emptyset, \emptyset, T_r, 0, 0, 0, \emptyset, \emptyset]$. We assume that the statement is true for $i - 1$; that is, there exists $D' \cup A' \subseteq D \cup A$ and a valid tuple $(T'_m, T'_g, T'_r, k'_1, k'_2, i - 1, X, Y)$ such that $D' \cup A' \in \mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i-1, X, Y]$ and there is a proper deficiency map ψ' over $R = (S(G - D', f) \cup T'_g \cup Y) \setminus (T'_r \cup X_{D',f})$ with the property that $E_{\psi'} = A'$. Due to Lemma 4.6, $D' \cup A' \in \mathcal{Q}(T'_m, T'_g, T'_r, k'_1, k'_2, i - 1, X, Y)$. We need to show that the statement holds for i.

Case 1: $\boldsymbol{X, Y = \emptyset}$**.** Since $i - 1 < k$, we have that $D' \neq D$ or $A' \neq A$.

Subcase (i): $D' \neq D$. Let $e = (x, y) \in D \setminus D'$. Let $D'' = D' \cup \{e\}$, $X' = \{z \in \{x, y\} \mid z \notin T_m \setminus T'_m\}$, and $Y' = \emptyset$. Let $T''_m = T'_m \cup (\{x, y\} \cap (T_m \setminus T'_m))$. We have several cases based on vertices belong to X'.

Suppose $X' = \{x, y\}$ and $x, y \in \mathsf{Green}$. By the definition of X', since $x, y \in X'$, we have that $x, y \notin T_m \setminus T'_m$. This implies that $T''_m = T'_m$, $T_m(x) = T'_m(x)$ and $T_m(y) = T'_m(y)$. Thus, by (7), $D'' \cup A' \in \mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i - 1, \emptyset, \emptyset] \bullet \{(x, y)\} \subseteq \mathcal{D}[T'_m, T'_g, T'_r, k'_1 + 1, k'_2, i, X', \emptyset]$.

Suppose $X' = \{x, y\}$, $x \in \mathsf{Green}$ and $y \in \mathsf{Red}$. By the definition of X', since $x \in X'$, we have that $x \notin T_m \setminus T'_m$. This implies that $T''_m = T'_m$ and $T_m(x) = T'_m(x)$.

Thus by (7), $D''\cup A' \in \mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i-1, \emptyset, \emptyset]\bullet\{(x,y)\} \subseteq \mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2+1, i, X', \emptyset]$.

Suppose $X' = \{x\}$ and $x \in \mathsf{Green}$. Then by the definition of X', since $x \in X'$ and $y \notin X'$, we have that $x \notin T_m \setminus T'_m$ and $y \in T_m \setminus T'_m$. This implies that $T''_m = T'_m \cup \{y\}$ and $T_m(x) = T'_m(x)$. Thus by (7), $D'' \cup A' \in \mathcal{D}[T''_m \setminus \{y\}, T'_g, T'_r, k'_1, k'_2, i-1, \{x\} \setminus \{x\}, \emptyset] \bullet \{(x,y)\} \subseteq \mathcal{D}[T''_m, T'_g, T'_r, k'_1+1, k'_2, i, X', \emptyset]$.

Suppose $X' = \{x\}$ and $x \in \mathsf{Red}$. By the definition of X', since $y \notin X'$, we have that $y \in T_m \setminus T'_m$. This implies that $T''_m = T'_m \cup \{y\}$. Thus by (7), $D'' \cup A' \in \mathcal{D}[T''_m \setminus \{y\}, T'_g, T'_r, k'_1, k'_2, i-1, \{x\} \setminus \{x\}, \emptyset] \bullet \{(x,y)\} \subseteq \mathcal{D}[T''_m, T'_g, T'_r, k'_1, k'_2+1, i, X', \emptyset]$.

Suppose $X' = \emptyset$. Then by the definition of X', since $x, y \notin X'$, we have that $x, y \in T_m \setminus T'_m$ and $T''_m = T'_m \cup \{x,y\}$. Thus by (6), $D'' \cup A' \in \mathcal{D}[T''_m \setminus \{x,y\}, T'_g, T'_r, k'_1, k'_2, i-1, \emptyset, \emptyset] \bullet \{(x,y)\} \subseteq \mathcal{D}[T''_m, T'_g, T'_r, k'_1+1, k'_2, i, \emptyset, \emptyset]$.

In all the above cases, we can show that ψ' is a proper deficiency map over $(S(G-D'', f) \cup T'_g) \setminus (T'_r \cup X'_{D'',f})$ and its proof is the same as the corresponding proof in Lemma 4.5. By an argument similar to the one in the proof of Lemma 4.5, we can show that $(T''_m, T'_g, T'_r, k'_1+1, k'_2, i, \emptyset, \emptyset)$ is a valid tuple.

Subcase (ii): $D' = D$. In this subcase we have that $A' \neq A$. Let $(x,y) \in A \setminus A'$. Let $A'' = A' \cup \{(x,y)\}$. Note that $E_\psi = A$ and $E_{\psi'} = A'$. Since $D' = D$, for all $v \in \mathsf{Green}$ the number of edges in D' which are incident with v is equal to $\Phi(v)$. Also, note that $T_m(v) + T_g(v) = \Phi(v)$. This implies that $T'_m = T_m$ and $T'_g = T_g$. Since ψ is a proper deficiency map over $S(G-D,f) = S(G-D,f) \cup T_g$, ψ' is a proper deficiency map over $(S(G-D',f) \cup T'_g) \setminus T'_r$, and $(x,y) \in E_\psi \setminus E_{\psi'}$, there exists j, j' such that $x(j), y(j') \in T'_r$. Thus, by (6), $D' \cup A'' \in \mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i-1, \emptyset, \emptyset] \circ \{(x,y)\} \subseteq \mathcal{D}(T'_m, T'_g, T'_r \setminus \{x(j), y(j')\}, k'_1, k'_2, i, \emptyset, \emptyset]$.

We can show that there is a proper deficiency map ψ'' over $(S(G-D',f) \cup T'_g) \setminus (T'_r \setminus \{x(j), y(j')\})$ such that $A'' = E_{\psi''}$ and its proof is the same as the corresponding proof in Lemma 4.5. Since $D' \cup A' \in \mathcal{Q}(T'_m, T'_g, T'_r, k'_1, k'_2, i-1, X, Y)$, by an argument similar to the one in the proof of Lemma 4.5, we can show that $(T'_m, T'_g, T'_r \setminus \{x(j), y(j')\}, k'_1, k'_2, i, \emptyset, \emptyset)$ is a valid tuple.

Case 2: $X \neq \emptyset, Y = \emptyset$. Let $x \in X$ and let j be the smallest integer such that $x(j) \in X_{D',f}$. Since $X_{D',f} \subseteq S(G-D',f) \setminus T_r$, we have that $x(j) \in S(G-D',f) \subseteq S(G-D,f)$. Also since ψ' is a proper deficiency map over $(S(G-D',f) \cup T'_g) \setminus (T'_r \cup X_{D',f})$ and $E_{\psi'} \subseteq E_\psi$, there exists $y \in V(G)$ such that $(x,y) \notin E_{\psi'}$ and $(x,y) \in E_\psi$. Let $A'' = A' \cup \{(x,y)\}$.

Subcase (i): $y \in X$. Let $j' = f(y) - d_{G-D'}(y)$. Note that $\{x(j), y(j')\} = X_{D',f} \subseteq S(G-D',f) \setminus T_r$. Let $T''_g = T'_g \cup (\{x(j), y(j')\} \cap T_g)$.

CLAIM 8. *If $x \in \mathsf{Green}$, then $x(j) \notin T'_g$ and $x(j-1) \in T'_g$. If $y \in \mathsf{Green}$, then $y(j') \notin T'_g$ and $y(j-1) \in T'_g$.*

Claim 8 is identical to Claim 3. Thus, by (6), $D' \cup A'' \in \mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i-1, X, \emptyset] \circ \{(x,y)\} \subseteq \mathcal{D}[T'_m, T''_g, T'_r, k'_1, k'_2, i, \emptyset, \emptyset]$. We can show that there is a proper deficiency map ψ'' over $(S(G-D',f) \cup T''_g) \setminus T'_r$ such that $A'' = E_{\psi''}$ and its proof is the same as the corresponding proof in Lemma 4.5. Since $D' \cup A' \in \mathcal{Q}(T'_m, T'_g, T'_r, k'_1, k'_2, i-1, X, Y)$, by argument similar to the one in the proof of Lemma 4.5, we can show that $(T'_m, T''_g, T'_r, k'_1, k'_2, i, \emptyset, \emptyset)$ is a valid tuple.

Subcase (ii): *$y(j') \in T'_r$ for some j'.* Let $T''_g = T'_g \cup (\{x(j)\} \cap T_g)$.

CLAIM 9. *If $x \in \mathsf{Green}$, then $x(j) \notin T'_g$ and $x(j-1) \in T'_g$.*

Claim 9 is identical to Claim 5. Thus, by (6) (if $X = \{x\}$) and by (7) (if $X \neq \{x\}$),

$$D' \cup A'' \in \mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i-1, X, \emptyset] \circ (x, y) \subseteq \mathcal{D}[T'_m, T''_g, T'_r \backslash y(j'), k'_1, k'_2, i, X \backslash \{x\}, \emptyset].$$

We can show that there is a proper deficiency map ψ'' over $(S(G-D', f) \cup T''_g) \setminus ((T'_r \setminus \{y(j)\}) \cup X'_{D',f})$, where $X' = X \setminus \{x\}$, such that $A'' = E_{\psi''}$ and its proof is the same as the corresponding proof in Lemma 4.5. Since $D' \cup A' \in \mathcal{Q}(T'_m, T'_g, T'_r, k'_1, k'_2, i-1, X, Y)$, by an argument similar to the one in the proof of Lemma 4.5, we can show that $(T'_m, T''_g, T'_r \setminus y(j'), k'_1, k'_2, i, X \setminus \{x\}, \emptyset)$ is a valid tuple.

Subcase (iii): *$y \notin X$ and $y(j') \notin T'_r$ for all j'.* Let $T''_g = T'_g \cup (\{x(j)\} \cap T_g)$.

CLAIM 10. *If $x \in \mathsf{Green}$, then $x(j) \notin T'_g$ and $x(j-1) \in T'_g$.*

Proof of the above claim is the same as that of Claim 5 as both are identical. Thus, by (8),

$$D' \cup A'' \in \mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i-1, X, \emptyset] \circ (x, y) \subseteq \mathcal{D}[T'_m, T''_g, T'_r, k'_1, k'_2, i, X \setminus \{x\}, \{y\}].$$

We can show that there is a proper deficiency map ψ'' over $(S(G-D', f) \cup T''_g \cup Y') \setminus (T'_r \cup X'_{D',f})$, where $X' = X \setminus \{x\}$ and $Y' = \{y\}$, such that $A'' = E_{\psi''}$ and its proof is the same as the corresponding proof in Lemma 4.5. Since $D' \cup A' \in \mathcal{Q}(T'_m, T'_g, T'_r, k'_1, k'_2, i-1, X, Y)$, by an argument similar to the one in the proof of Lemma 4.5, we can show that $(T'_m, T''_g, T'_r, k'_1, k'_2, i, X \setminus \{x\}, \{y\})$ is a valid tuple.

Case 3: $\boldsymbol{Y \neq \emptyset}$. Let $Y = \{y\}$. Since $(T'_m, T'_g, T'_r, k'_1, k'_2, i-1, X, Y)$ is a valid tuple and $|Y| = 1$, we have that $|X| \leq 1$. There exists x such that $(y, x) \in D \setminus D'$ and the proof of this statement can be found in the proof of Lemma 4.5. Let $D'' = D' \cup \{(y, x)\}$ and $X' = X \cup \{z | z = x, z \notin T_m \setminus T'_m\}$. Note that $|X'| \leq 2$, because $|X| \leq 1$. Let $T''_m = T'_m \cup (\{x\} \cap (T_m \setminus T'_m))$.

Subcase (i): $y \in \mathsf{Green}$. Then there exists $j \in \mathbb{N}^+$ such that $y(j) \in T_g \setminus T'_g$ and its proof can be found in Lemma 4.5. Without loss of generality, let j be the smallest integer such that $y(j) \in T_g \setminus T'_g$. Let $T''_g = T'_g \cup \{y(j)\}$.

Subsubcase (ia): $(y, x) \in E(G[\mathsf{Green}])$. If $X = X' = \emptyset$, then by the definition of X', $T''_m = T'_m \cup \{x\}$. Then, by (6),

$$D'' \cup A' \in \mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i-1, \emptyset, Y] \bullet (y, x) \subseteq \mathcal{D}[T''_m, T''_g, T'_r, k'_1 + 1, k'_2, i, X', \emptyset].$$

If $X = X' \neq \emptyset$, then by the definition of X', $T''_m = T'_m \cup \{x\}$. Then, by (7), $D'' \cup A' \in \mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i-1, X, Y] \bullet (y, x) \subseteq \mathcal{D}[T''_m, T''_g, T'_r, k'_1 + 1, k'_2, i, X', \emptyset]$.

If $X \neq X'$, then by by the definition of X', $T''_m = T'_m$. Then, by (7), $D'' \cup A' \in \mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i-1, X, Y] \bullet (y, x) \subseteq \mathcal{D}[T'_m, T''_g, T'_r, k'_1 + 1, k'_2, i, X', \emptyset]$.

We can show that there is a proper deficiency map ψ'' over $(S(G-D', f) \cup T''_g) \setminus (T'_r \cup X'_{D',f})$, such that $A'' = E_{\psi''}$ and its proof is the same as the corresponding proof in Lemma 4.5. Since $D' \cup A' \in \mathcal{Q}(T'_m, T'_g, T'_r, k'_1, k'_2, i-1, X, Y)$, by an argument similar to the one in the proof of Lemma 4.5, we can show that $(T''_m, T''_g, T'_r, k'_1 + 1, k'_2, i, X', \emptyset)$ is a valid tuple.

Subsubcase (ib): $(y, x) \notin E(G[\mathsf{Green}])$. In this subcase $X' = X \cup \{x\}$. Then by (7), $D'' \cup A' \in \mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i-1, X, Y] \bullet (y, x) \subseteq \mathcal{D}[T'_m, T''_g, T'_r, k_1 1, k'_2 + 1, i, X', \emptyset]$.

We can show that there is a proper deficiency map ψ'' over $(S(G-D', f) \cup T''_g) \setminus (T'_r \cup X'_{D',f})$, such that $A'' = E_{\psi''}$ and its proof is the same as the corresponding proof in Lemma 4.5. Since $D' \cup A' \in \mathcal{Q}(T'_m, T'_g, T'_r, k'_1, k'_2, i-1, X, Y)$, by an argument similar to the one in the proof of Lemma 4.5, we can show that $(T''_m, T''_g, T'_r, k'_1, k'_2 + 1, i, X', \emptyset)$ is a valid tuple.

Subcase (ii): $y \in \mathsf{Red}$. If $X = X' = \emptyset$, then by the definition of X', $T''_m = T'_m \cup \{x\}$. Then, by (6), $D'' \cup A' \in \mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i-1, \emptyset, Y] \bullet (y,x) \subseteq \mathcal{D}[T''_m, T'_g, T'_r, k'_1, k'_2 + 1, i, \emptyset, \emptyset]$. If $X = X' \neq \emptyset$, then by the definition of X', $T''_m = T'_m \cup \{x\}$. Then, by (7), $D'' \cup A' \in \mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i-1, X, Y] \bullet (y,x) \subseteq \mathcal{D}[T''_m, T'_g, T'_r, k'_1, k'_2 + 1, i, X', \emptyset]$.

If $X \neq X'$, then by the definition of X', $T''_m = T'_m$. Then, by (7), $D'' \cup A' \in \mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i-1, X, Y] \bullet (y,x) \subseteq \mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2 + 1, i, X', \emptyset]$.

We can show that there is a proper deficiency map ψ'' over $(S(G - D', f) \cup T'_g) \setminus (T'_r \cup X'_{D',f})$, such that $A'' = E_{\psi''}$ and its proof is the same as the corresponding proof in Lemma 4.5. Since $D' \cup A' \in \mathcal{Q}(T'_m, T'_g, T'_r, k'_1, k'_2, i-1, X, Y)$, by an argument similar to the one in the proof of Lemma 4.5, we can show that $(T''_m, T'_g, T'_r, k'_1, k'_2 + 1, i, X', \emptyset)$ is a valid tuple.

Now we need to show that $D \cup A \in \mathcal{D}[T_m, T_g, \emptyset, k_1, k_2, k, \emptyset, \emptyset]$. We have already shown that there is a valid tuple $(T'_m, T'_g, T'_r, k'_1, k'_2, k, X, Y)$ such that $D \cup A \in \mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, k, X, Y]$. Due to Lemma 4.6, $D \cup A \in \mathcal{Q}(T'_m, T'_g, T'_r, k'_1, k'_2, k, X, Y)$. This implies that there is a proper deficiency map ψ' over $(S(G - D, f) \cup T'_g \cup Y) \setminus (T'_r \cup X_{D,f})$ such that $E_{\psi'} = A$. Since $D \cup A$ is a solution to EECG, there is a proper deficiency map ψ over $S(G - D, f)$ such that $E_\psi = A$. This implies that

$$S(G - D, f) = (S(G - D, f) \cup T'_g \cup Y) \setminus (T'_r \cup X_{D,f}). \tag{9}$$

Equation (9) implies that $Y = \emptyset$. Since $T'_r \subseteq S(G - D, f)$ and $X_{D,f} \subseteq S(G - D, f) \setminus T_r$, by (9), we get that $T'_r = \emptyset$ and $X = \emptyset$. Since $D \cup A \in \mathcal{Q}(T'_m, T'_g, T'_r, k'_1, k'_2, k, X, Y)$, we have that for any $v \in \mathsf{Green}$, $|E_G(v) \cap D| = T'_m(v) + T'_g(v)$. By assumption we know that $|E_G(v) \cap D| = \Phi(v) = T_m(v) + T_g(v)$. This implies that $T'_m = T_m$ and $T'_g = T_g$. We also know, by assumption, that $D \cap E(G[\mathsf{Green}]) = k_1$ and $D \setminus E(G[\mathsf{Green}]) = k_2$. This implies that $k'_1 = k_1$ and $k'_2 = k_2$. Hence $D \cup A \in \mathcal{D}[T_m, T_g, \emptyset, k_1, k_2, k, \emptyset, \emptyset]$. This completes the proof. ∎

4.3. Pruning the table—FPT algorithm. Now we explain how to prune the family of partial solutions stored at each DP table entry such that its size is at most $2^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$ and thereby get an FPT algorithm. The objective is to find a nice deletion set $D \subseteq E(G)$. In fact, if the input instance is a Yes instance, we will find a set $D \subseteq E(G)$, $A \subseteq \overline{E(G)}$ such that D is a nice deletion set with the property that $A = E_\psi$, where ψ is a proper deficiency map over $S(G - D, f)$.

Recall that for the algorithm we have guessed k'—the size of proposed deletion set D, k_1—the number edges in $D \cap E(G[\mathsf{Green}])$, k_2—the number of edges in $D \setminus E(G[\mathsf{Green}])$ and for all $v \in \mathsf{Green}$, $\Phi(v)$ ($\geq d_G(v) - f(v)$)—the number of edges in D which are incident with v. Consider the property (i) of nice deletion set, i.e., $d_{G-D}(v) \leq f(v)$ for all v. By guessing $\Phi(v) \geq d_G(v) - f(v)$ for all $v \in \mathsf{Green}$, we know that any solution we compute will satisfy property (i).

Consider property (ii) of a nice deletion set, i.e., $|S(G - D, f)| = 2(k - k')$. Since the total number of edges in D which has one endpoint in Green and other in Red is $(\sum_{v \in \mathsf{Green}} \Phi(v)) - 2k_1$, we have that $\sum_{v \in \mathsf{Red}} |D \cap E_G(v)| = 2k_2 - ((\sum_{v \in \mathsf{Green}} \Phi(v)) - 2k_1)$,

$$|S(G - D, f)| = \left(\sum_{v \in \mathsf{Green}} \Phi(v) - (d_G(v) - f(v)) \right) + \sum_{v \in \mathsf{Red}} (f(v) - d_{G-D}(v))$$

$$= \left(\sum_{v \in \mathsf{Green}} \Phi(v) - (d_G(v) - f(v)) \right) + \sum_{v \in \mathsf{Red}} (f(v) - d_G(v)) + \sum_{v \in \mathsf{Red}} |D \cap E_G(v)|$$

$$= \left(\sum_{v \in \mathsf{Green}} \Phi(v) - (d_G(v) - f(v)) \right) + \sum_{v \in \mathsf{Red}} (f(v) - d_G(v)) + \left(2k_2 + 2k_1 - \sum_{v \in \mathsf{Green}} \Phi(v) \right)$$

$$= 2k_1 + 2k_2 + \sum_{v \in V(G)} (f(v) - d_G(v)).$$

So after guessing k', k_1, k_2 and $\Phi(v)$ for all $v \in \mathsf{Green}$, we check whether $2k_1 + 2k_2 + \sum_{v \in V(G)} (f(v) - d_G(v)) = 2(k - k')$, and if they are not equal, we consider it as an invalid guess. Thus our guesses take care of property (ii).

The property (iii) of a nice deletion set and Lemma 4.8 below imply that D is an independent set in the matroid $M_G(\ell)$, the ℓ-elongation of the co-graphic matroid M_G associated with G, where $\ell = |E(G)| - |V(G)| + k - |D| + 1$.

LEMMA 4.8. *Let G be a graph and $D \subseteq E(G)$. Then D is an independent set in $M_G(\ell)$ where $\ell = |E(G)| - |V(G)| + k - |D| + 1$ if and only if $G - D$ has at most $k - |D| + 1$ connected components.*

Proof. Let r be the number of connected components in G. Suppose D is an independent set in $M_G(\ell)$. Then there exists $S \subseteq E(G) \setminus D$ such that $S \cup D$ is a basis of $M_G(\ell)$. This implies that there exists $S' \subseteq S \cup D$ such that S' is a basis of M_G, and hence $G - S'$ is a forest with r connected components and $|S'| = E(G) - V(G) + r$. Since $|S \cup D| - |S'| = (k - |D| + 1) - r$ and $G - S'$ is a forest with exactly r connected components, we have that $G - (S \cup D)$ has exactly $k - |D| + 1$ connected components. This implies that $G - D$ has at most $k - |D| + 1$ connected components.

Suppose $G - D$ has at most $k - |D| + 1$ connected components. Let $S \subseteq E(G)$ be a maximal subset such that $G - (S \cup D)$ is a forest with exactly $k - |D| + 1$ connected components. This implies that $|S \cup D| = E(G) - V(G) + k - |D| + 1$. Since G has r connected components, there exists $S' \subseteq (S \cup D)$ of size $(k - |D| + 1) - r$ such that $G - ((S \cup D) \setminus S')$ is a forest with exactly r connected components. Since $|(S \cup D) \setminus S'| = E(G) - V(G) + r$ and $G - ((S \cup D) \setminus S')$ is a forest with exactly r connected components, $(S \cup D) \setminus S'$ is a basis in M_G. This implies that $S \cup D$ is a basis in $M_G(\ell)$ and hence D is an independent set in $M_G(\ell)$. ◻

Thus by only considering those D which are independent sets in $M_G(\ell)$ we ensure that property (iii) of the nice deletion set is satisfied.

Now consider the property (v) of a nice deletion set, i.e., there exists a proper deficiency map $\psi : S(G - D, f) \to S(G - D, f)$. Our objective is to get a set $D \cup A$ such that there is a proper deficiency map ψ over $S(G - D, f)$ such that $E_\psi = A$, along with other properties as well. We have already defined equivalence classes for the partial solutions in the previous section. Let $D_1 \cup A_1, D_2 \cup A_2 \in Q(T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y)$ be two partial solutions where $D_1, D_2 \subseteq E(G)$ and $A_1, A_2 \subseteq \overline{E(G)}$. Suppose $D' \subseteq E(G)$, $A' \subseteq \overline{E(G)}$, $(D_1 \cup D') \bigcup (A_1 \cup A')$ is a solution, and $A_2 \cap A' = \emptyset$. Since $D_1 \cup A_1$ and $D_2 \cup A_2$ belong to the same equivalence class and $A_2 \cap A'$ is disjoint, there is a proper deficiency map ψ' over $S(G - (D_2 \cup D'), f)$ such that $E_{\psi'} = A_2 \cup A'$.

To take care of the disjointness property between the current addition set and the future addition set while doing the DP, we view the addition set A of a solution as an independent set in a uniform matroid over the universe $\overline{E(G)}$. Let $U_{m',k-k'}$ be the uniform matroid with ground set $\overline{E(G)}$, where $m' = |\overline{E(G)}|$. From the definition of $U_{m',k-k'}$, any set A of size at most $k - k'$ is independent in $U_{m',k-k'}$. We have already explained that we view the deletion set D as an independent set in $M_G(\ell)$ where $\ell = |E(G)| - |V(G)| + k - k' + 1$. Thus, to see the solution set $D \cup A$ as an independent set in a single matroid, we consider the direct sum of $M_G(\ell)$ and $U_{m',k-k'}$; that is, let $M = M_G(\ell) \oplus U_{m',k-k'}$. In M, a set I is an independent set if and only if $I \cap E(G)$ is an independent set in $M_G(\ell)$ and $I \cap \overline{E(G)}$ is an independent set in $U_{m',k-k'}$. This ensures that any solution $D \cup A$ is an independent set in M. By viewing any solution of the problem as an independent set in the matroid M (which is linear), we can use the representative families to prune DP table entries. However, we still need to ensure that property (iv) of a nice deletion set is satisfied. In what follows we explain how we achieve this.

Consider the property (iv) mentioned in the definition of a nice deletion set; that is, for any connected component F in $G - D$, $V(F) \cap \mathsf{def}(G - D, f) \neq \emptyset$. The following lemma helps us to satisfy property (iv) partially.

LEMMA 4.9. *Let F be a connected component in the graph $G[\mathsf{Red}]$ and let $D' \subseteq E(G)$. If at least one edge in D' is incident with a vertex in $V(F)$, then for any connected component C in $G - D'$ such that $V(C) \cap V(F) \neq \emptyset$, there is a vertex $v \in V(C) \cap \mathsf{def}(G - D', f)$.*

Proof. Let $u \in V(F)$ be a vertex such that an edge in D' is incident with u. Consider a connected component C in $G - D'$ such that $V(C) \cap V(F) \neq \emptyset$. We need to show that $V(C) \cap \mathsf{def}(G - D', f) \neq \emptyset$. Suppose $u \in V(C)$. Since $u \in \mathsf{Red}$, $d_G(u) \leq f(v)$. However, u is incident with an edge in D', and thus we have that $d_{G-D'}(u) < f(u)$. This implies that $u \in V(C) \cap \mathsf{def}(G - D', f)$. Now we are in a case where $u \notin V(C)$. Pick an arbitrary vertex $w \in V(C) \cap V(F)$. Since $w, u \in V(F)$, there exists a path P from w to u using only vertices from Red. Since w and u are in different connected components in $G - D'$, $D' \cap E(P) \neq \emptyset$. Pick the first edge (v, v') in the path P which are also in D'. Note that there exists a path from w to v in $G - D'$ and $v \in V(C)$. Since $v \in \mathsf{Red}$ and $(v, v') \in D'$, we have that $v \in V(C) \cap \mathsf{def}(G - D', f)$. This completes the proof. ∎

Now we explain how Lemma 4.9 is useful in satisfying property (iv) partially. Let $\mathcal{C}$ be the set of connected components in G such that for each vertex v in the component, $d_G(v) = f(v)$,

$$\mathcal{C} = \{C \mid C \text{ is a connected component in } G \wedge \forall v \in V(C), d_G(v) = f(v)\}.$$

Let D_1 and D_2 be deletion sets corresponding to two partial solutions such that for all $C \in \mathcal{C}$, $D_1 \cap E(C) \neq \emptyset$ if and only if $D_2 \cap E(C) \neq \emptyset$. Suppose there is a set $D' \subseteq E(G)$ such that $D_1 \cup D'$ is a nice deletion set. Now we claim that any connected component F in $G - (D_2 \cup D')$ containing only red vertices will have a deficient vertex. Let $v \in V(F)$ and $v \notin V(\mathcal{C})$. We also know that $v \in \mathsf{Red}$. Since $v \notin V(\mathcal{C}) (= \bigcup_{C \in \mathcal{C}} V(C))$ one of the following conditions hold:

1. There is a path from v to a vertex in Green in the graph G.
   Since $v \in V(F)$ and F is a fully red connected component in $G - (D_2 \cup D')$, there is a vertex w in $V(F)$ such that $D_2 \cup D'$ contains an edge incident with w. Since $w \in \mathsf{Red}$ as well, $w \in \mathsf{def}(G - (D_2 \cup D'), f)$.

2. Else, v is in a connected component C_1 of G such that $V(C_1) \subseteq \mathsf{Red}$ and there is a vertex $u \in V(C_1)$ such that $f(u) > d_G(v)$.
   If $F = C_1$, then u is the required deficient vertex. If $F \neq C_1$, then by Lemma 4.9, $V(F) \cap \mathsf{def}(G - (D_2 \cup D'), f) \neq \emptyset$.

Let $v \in V(F)$ and $v \in V(C)$ where $C \in \mathcal{C}$. Since $D_1 \cup D'$ is a solution, either $D_1 \cap E(C) \neq \emptyset$ or $D' \cap E(C) \neq \emptyset$. If $D_1 \cap E(C) \neq \emptyset$, then by our assumption, $D_2 \cap E(C) \neq \emptyset$. Thus by Lemma 4.9, $V(F) \cap \mathsf{def}(G - (D_2 \cup D'), f) \neq \emptyset$.

Essentially due to Lemma 4.9, if we partition our partial solutions based on how these partial solutions hit the edges from $\mathcal{C}$ and keep at least one from each equivalence class, the property (iv) of a nice deletion set will be satisfied partially. But this only allows us to take care of connected components containing only red vertices. Now we explain how we can ensure property (iv) for the connected components containing vertices from Green as well.

To achieve this we will prove that corresponding to every deletion set D of a solution, there is a "witness" of $\mathcal{O}(k)$ sized subset of edges whose disjointness from D will ensure property (iv) of nice deletion sets; that is, these witnesses are dependent on solutions; the witness for solution D will be different from the witness for solution D^*. Even then, these witnesses allow us to satisfy property (iv). In order to avoid this witness being picked in a deletion set D, that is to keep this witness nondeletable, we use color coding in our algorithm on top of representative family based pruning of table entries. Towards that we define a weight function w on $E(G)$ as follows:

$$\mathrm{w}((u,v)) = \begin{cases} 0 & \text{if } u, v \in \mathsf{Red}, \\ 1 & \text{otherwise.} \end{cases}$$

For any subset $S \subseteq E(G)$, $\mathrm{w}(S) = \sum_{e \in S} \mathrm{w}(e)$. The next lemma is crucial for our approach as this not only defines the witness but also gives an upper bound on its size.

LEMMA 4.10. *Let* Green $= \{v_1, v_2, \ldots, v_\eta\}, \eta \leq 2k$. *Let* $D \subseteq E(G)$ *such that for any connected component* F *in* $G - D$, $V(F) \cap \mathsf{def}(G - D, f) \neq \emptyset$. *Then there exist paths* $P_1, \ldots, P_\eta$ *such that for all* i, P_i *is a path in* $G - D$ *from* v_i *to a vertex in* $\mathsf{def}(G - D, f)$, *and* $\mathrm{w}(\bigcup_i E(P_i)) \leq 6k$ *where* $\bigcup_i E(P_i)$ *is the set of edges in the paths* $P_1, \ldots P_\eta$.

Proof. We construct $P_1, \ldots, P_\eta$ with the required property. Pick an arbitrary vertex $u_1 \in \mathsf{def}(G - D, f)$ such that v_1 and u_1 are in the same connected component in $G - D$. Let P_1 be a smallest weight path according to weight function w, from v_1 to u_1 in $G - D$. Now we explain how to construct P_i, given that we have already constructed paths $P_1, \ldots, P_{i-1}$. Pick an arbitrary vertex $u_i \in \mathsf{def}(G - D, f)$ such that v_i and u_i are in the same connected component in $G - D$. Let P be a smallest weight path from v_i to u_i in $G - D$. If P is vertex disjoint from $P_1, \ldots, P_{i-1}$, then we set $P_i = P$. Otherwise, let x be the first vertex in P such that $x \in V(P_1) \cup \ldots \cup V(P_{i-1})$. Let $x \in P_j$ where $j < i$. Let $P = P'P''$ such that P' ends in x and P'' starts at x. Let $P_j = P'_j P''_j$ such that P'_j ends in x and P''_j starts at x. Now we set $P_i = P' P''_j$. Note that P_i is a path in $G - D$ from v_i to a vertex in $\mathsf{def}(G - D, f)$.

Now we claim that $\mathrm{w}(\bigcup_{i=1}^{\eta} E(P_i)) \leq 6k$. Towards the proof, we need to count that $|(\bigcup_{i=1}^{\eta} E(P_i)) \cap \mathrm{w}^{-1}(1)| \leq 6k$. We assign each vertex v in $\bigcup_{i=1}^{\eta} V(P_i)$ to the smallest indexed path P_j such that $v \in V(P_j)$; that is, v is assigned to P_j if $v \in V(P_j)$ and $v \notin (\bigcup_{i=1}^{j-1} V(P_i)$. Note that each vertex in $\bigcup_{i=1}^{\eta} V(P_i)$ is assigned to a unique path. Consider the edge set $A^* \subseteq (\bigcup_{i=1}^{\eta} E(P_i)) \cap \mathrm{w}^{-1}(1)$ as follows. An edge $e = (u, v)$ belongs to A^* if $\mathrm{w}(e) = 1, e \in E(P_j)$, and vertices u and v are assigned to path P_j

for some j. Observe that each edge $e \in A^*$ has at least one endpoint in Green. Since each vertex is assigned to exactly one path, each vertex in a path has degree at most 2 and $|\mathsf{Green}| \leq 2k$, we have that $|A^*| \leq 4k$.

Now we show that there exist sets $\emptyset = B_1 \subseteq B_2 \subseteq \ldots B_\eta$ such that $(\bigcup_{i=1}^{j} E(P_i)) \cap \mathrm{w}^{-1}(1) \subseteq A^* \cup B_j$ and $|B_j| \leq j$. We prove the statement using induction on j. For $j = 1$, we know that $(\bigcup_{i=1}^{j} E(P_i)) \cap \mathrm{w}^{-1}(1) \subseteq A^*$. Thus, the statement is true. Now suppose the statement is true for $j-1$. Consider any path P_j. If the vertices in P_j are disjoint from $\bigcup_{i=1}^{j-1} V(P_i)$, then all the weight one edges in $E(P_j)$ are counted in A^*. So we can set $B_j = B_{j-1}$ and the statement is true. Otherwise, by the construction of P_j, we have that $P_j = P'_j P''_j$ and there exists $r < j$ such that $P_r = P'_r P''_j$. Let (u_1, u_2) be the last edge in P'_j. Note that all the weight one edges in $E(P''_j)$ are counted in $A^* \cup B_{j-1}$ and all the weight one edges in $E(P'_j) \setminus \{(u_1, u_2)\}$ are counted in A^*. In this case we set $B_j = B_{j-1}$ if $\mathrm{w}((u_1, u_2) = 0)$ and $B_j = B_{j-1} \cup \{(u_1, u_2)\}$ otherwise. This implies that $|(\bigcup_{i=1}^{\eta} E(P_i)) \cap \mathrm{w}^{-1}(1)| \leq 6k$. This concludes the proof. ∎

Recall that $E_r = E(G[\mathsf{Red}])$ and $E_g = E(G) \setminus E_r$. Note that in Lemma 4.10, the weight of each edge in E_g is 1 and the weight of each edge in E_r is 0. By Lemma 4.10, we have that if D is a nice deletion set, then there exists $E' \subseteq E_g$ of cardinality at most $6k$ such that E' witnesses that each connected component of $G - D$ containing at least one vertex from Green, will also contain a vertex from $\mathsf{def}(G - D, f)$. We call such an edge set E' as *certificate* of D. Now we explain how Lemma 4.10 helps us to satisfy property (iv) of nice deletion sets for components containing at least one vertex from Green. Let $\mathsf{Green} = \{v_1, \ldots, v_\eta\}$ and $D_1 \cup D'$ be a deletion set corresponding to a solution. By Lemma 4.10 we know that there are paths $P_1, \ldots, P_\eta$ such that the total number of edges from E_g among these paths is bounded by $6k$, and each path P_i is from v_i to a vertex in $\mathsf{def}(G - (D_1 \cup D'), f)$. Suppose we color the edges in E_g with black and orange such that the coloring guarantees that all the edges in $E_g \cap (\bigcup_{i=1}^{\eta} E(P_i))$ are colored black and all the edges in $E_g \cap (D_1 \cup D')$ are colored orange. Assume that we are going to find a nice deletion set which does not contain black color edges. Let D_2 be a deletion set corresponding to a partial solution. Also, for a vertex $v_i \in \mathsf{Green}$, there is a path from v_i to a vertex in $\mathsf{def}(G - D_1, f)$ in the graph $G - D_1$ which does not contain any orange colored edge if and only if there is path from v_i to a vertex in $\mathsf{def}(G - D_2, f)$ in the graph $G - D_2$ which does not contain any orange colored edge. Like in the case of red components, we can show that any connected component in $G - (D_2 \cup D')$ containing a vertex from Green will contain a vertex from $\mathsf{def}(G - (D_2 \cup D'), f)$. The formal proof of this statement will be given in Lemma 4.12. Essentially, by Lemma 4.10 we get the following. Suppose we take all partial solutions corresponding to a DP table entry (or a subset of it) and now we partition these partial solutions based on which all green vertices have found their deficient vertex currently (there are $2^{|\mathsf{Green}|}$ such partitions), then it is enough to keep a partial solution from each class. Furthermore, suppose $\mathcal{A}$ corresponds to partial solutions with respect to one particular subset of Green and we have kept a set D_1 in $\mathcal{A}$ and deleted the rest of the partial solutions from $\mathcal{A}$ (say one of the partial solutions we threw out was D_2). Then, if there is D' such that $D_2 \cup D'$ is a solution, then all the connected components in $G - (D_1 \cup D')$ containing at least one green vertex will have a deficient vertex. Just a word of caution that in our actual algorithm, in fact, we keep a subset of $\mathcal{A}$ of size $2^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$ so that we can also take care of all other properties of a nice deletion set. Even though we explained that the property (iv) can be achieved by imposing more structure to the equivalence class we defined in the last section, we will not include these structures in the index of the DP table entries.

Rather, for each table entry indexed by an equivalence class, we keep at least one partial solutions for each refinement of this equivalence class based on which green vertices have found their partner deficient vertex. This will ensure that property (iv) is satisfied.

We have explained how we will ensure each of the individual properties of a nice deletion set. Now we design a randomized FPT algorithm for the problem. Later, we derandomize the algorithm. The algorithm is a DP algorithm in which we have DP table entries indexed exactly in the same way as in the case of the XP algorithm. But instead of keeping $\mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y]$, we store a small representative family of $\mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y]$ which is enough to maintain the correctness of the algorithm. The algorithm uses both color coding and representative family techniques. We have explained that we use color coding to separate the proposed deletion set from its certificate mentioned in Lemma 4.10. We color each edge $e \in E_g$ black with probability 6/7 and orange with probability 1/7. Let E_b be the set of edges colored black and let E_o be the set of edges colored orange. Let D be a deletion set of size k' for the problem and let paths $P_1, \ldots, P_\eta$ be its witnesses mentioned in Lemma 4.10. Then the number of edges in paths $P_1, \ldots, P_\eta$, which are from E_g, is bounded by $6k$. We say that a random coloring is *good* if each edge in $D \cap E_g$ is colored orange and each edge in $E_g \cap (\bigcup_{i=1}^{\eta} P_i)$ is colored black. The random coloring of edges in E_g is *good*, with probability $\left(\frac{6^6}{7^7}\right)^k$. Now our algorithm works with the edge colored graph and output a nice deletion set D, with the property that $D \cap E_g \subseteq E_o$, if there exists such a deletion set. We know that if the input instance is a Yes instance, then with probability at least $\left(\frac{6^6}{7^7}\right)^k$ our algorithm will output a solution. Thus, we can increase the success probability to at least $(1 - 1/e)$ by running the entire algorithm $\left(\frac{7^7}{6^6}\right)^k$ times. From now on we assume that the edges in E_g of the input graph are colored with black or orange, and our objective is to find out a nice deletion set D such that all edges in $D \cap E_g$ are colored orange. Note that the edges in E_r are uncolored.

Recall that $\mathcal{C}$ is the set of connected components in G such that for each vertex v in the component, $d_G(v) = f(v)$. Now we define a family $\mathcal{J}$ of functions as

$$\mathcal{J} = \{g : \mathsf{Green} \cup \mathcal{C} \to \{0, 1\}\}.$$

Now we explain how to reduce the size of $\mathcal{D}[T'_g, T'_r, k'_1, k'_2, i, X, Y]$ which is computed using the recurrence relations (see (6), (7), and (8)). We say a partial solution $B \in \mathcal{D}[T'_g, T'_r, k'_1, k'_2, i, X, Y]$ is *properly colored* if $B \cap E_b = \emptyset$. Since our objective is to find out a nice deletion set disjoint from E_b, we delete all partial solutions which contain an edge from E_b; that is, if $B \in \mathcal{D}[T'_g, T'_r, k'_1, k'_2, i, X, Y]$ and $B \cap E_b \neq \emptyset$, then we delete B from $\mathcal{D}[T'_g, T'_r, k'_1, k'_2, i, X, Y]$. From now on we assume that for each $B \in \mathcal{D}[T'_g, T'_r, k'_1, k'_2, i, X, Y]$, $B \cap E_b = \emptyset$. Further pruning of the DP table entry $\mathcal{D}[T'_g, T'_r, k'_1, k'_2, i, X, Y]$ is discussed below.

Definition 4.11. *A subset $\mathcal{R}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y] \subseteq \mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y]$ is called a representative set of partial solutions for $\mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y]$, denoted by*

$$\mathcal{R}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y] \sqsubseteq_{rep}^{k-i} \mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y]$$

if the following holds. If there exist two sets $B, Z \subseteq \binom{V(G)}{2}$ such that B belongs to $\mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y]$, $B \cap Z = \emptyset$, $B \cap E_b = \emptyset$, and $(B \cup Z) \cap E(G)$ satisfies five properties of a nice deletion set with the property that there exists a proper deficiency *map ψ with $E_\psi = (B \cup Z) \cap \overline{E(G)}$, then there exists $\widehat{B} \subseteq \binom{V(G)}{2}$ such that $\widehat{B} \cap Z = \emptyset$, $\widehat{B} \cap E_b = \emptyset$, $\widehat{B} \in \mathcal{R}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y]$, and $(\widehat{B} \cup Z) \cap E(G)$ satisfies five properties of a nice deletion set with the property that there exists a proper deficiency map ψ' with $E_{\psi'} = (\widehat{B} \cup Z) \cap \overline{E(G)}$.*

For each valid tuple $(T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y)$ we compute a representative set of partial solutions for $\mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y]$ in the increasing order of i and store it instead of $\mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y]$. Now we explain how to compute it and prove its correctness. First, we compute a subfamily $\mathcal{S}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y]$ of $\mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y]$ using the recurrence relation—see (6), (7), and (8)—on the DP table entries computed for value $i-1$ and deleting all partial solutions which contain edges from E_b. Now we partition $\mathcal{S}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y]$ according to the refinement of each function in $\mathcal{J}$; that is , $\mathcal{S}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y] = \bigcup_{g \in \mathcal{J}} \mathcal{A}_g$, where $\mathcal{A}_g$ is defined as follows. For each $g \in \mathcal{J}$ and $S \cup R \in \mathcal{S}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y]$ where $S \in E(G)$ and $R \in \overline{E(G)}$, $S \cup R \in \mathcal{A}_g$ if the following happens:

(i) For any $v \in \mathsf{Green}$, $g(v) = 1$ if and only if there exists a path from v to a vertex in $\mathsf{def}(G-S, f)$ in $G[E_b \cup (E_r \setminus S)]$ (checking whether there is a witness path that does not use edges in E_o).

(ii) For any $C \in \mathcal{C}$, $g(C) = 1$ if and only if $S \cap E(C) \neq \emptyset$.

Recall that any set $S \cup R \in \mathcal{S}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y]$ is an independent set of size i in M. Now we compute $\widehat{\mathcal{A}}_g \subseteq^{k-i}_{rep} \mathcal{A}_g$ using Theorem 2.6. Then we set

$$\mathcal{D}[T'_m, T'_g, \widehat{T'_r, k'_1}, k'_2, i, X, Y] = \bigcup_{g \in \mathcal{J}} \widehat{\mathcal{A}}_g$$

and store it instead of $\mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y]$. The next lemma proves the correctness of this step.

LEMMA 4.12. $\mathcal{D}[T'_m, T'_g, \widehat{T'_r, k'_1}, k'_2, i, X, Y] \subseteq^{k-i}_{rep} \mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y]$.

Proof. We prove the lemma by induction on i. Suppose there exists $B, Z \subseteq \binom{V(G)}{2}$ such that $B \in \mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y]$, $B \cap Z = \emptyset$, $B \cap E_b = \emptyset$, $(B \cup Z) \cap E(G)$ satisfies the five properties of a nice deletion set and there exists a proper deficiency map ψ over $S(G - ((B \cup Z) \cap E(G)), f)$ with the property that $E_\psi = (B \cup Z) \cap \overline{E(G)}$. By the recurrence relations given by Equations 6,7 and 8, there exists $e \in B$ and a valid tuple $(T''_m, T''_g, T''_r, k''_1, k''_2, i-1, X', Y')$ such that

$$B \setminus \{e\} \in \mathcal{D}[T''_m, T''_g, T''_r, \widehat{k''_1, k''_2}, i-1, X', Y'].$$

Let $B' = B \setminus \{e\}$ and $Z' = Z \cup \{e\}$. We know that $(B' \cup Z') \cap E(G) = (B \cup Z) \cap E(G)$ satisfies five properties of a nice deletion set and ψ is a proper deficiency map over $S(G - ((B' \cup Z') \cap E(G)), f)$ with the property that $E_\psi = (B' \cup Z') \cap \overline{E(G)}$. Thus, by induction hypothesis we have that there exists $\widehat{B'} \subseteq \binom{V(G)}{2}$ such that

- $\widehat{B'} \in \mathcal{D}[T''_m, T''_g, T''_r, \widehat{k''_1, k''_2}, i-1, X', Y']$,
- $\widehat{B'} \cap Z' = \emptyset$, $\widehat{B'} \cap E_b = \emptyset$,
- $(\widehat{B'} \cup Z') \cap E(G)$ satisfies five properties of a nice deletion set, and
- there exists a proper deficiency map ψ' over $S(G - ((\widehat{B'} \cup Z') \cap E(G)), f)$ with the property that $E_{\psi'} = (\widehat{B'} \cup Z') \cap \overline{E(G)}$.

Since $(\widehat{B'} \cup Z') \cap E(G)$ is a nice deletion set, $G - ((\widehat{B'} \cup \{e\}) \cap E(G))$ has at most $k - k' + 1$ connected components. The definitions of $\bullet, \circ$ and recurrence relation of $\mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y]$ imply that

- if $e \in E(G)$, then $\widehat{B'} \cup \{e\} \in \mathcal{D}[T''_m, T''_g, T''_r, \widehat{k''_1, k''_2}, i-1, X', Y'] \bullet e$ and
- if $e \in \overline{E(G)}$, then $\widehat{B'} \cup \{e\} \in \mathcal{D}[T''_m, T''_g, T''_r, \widehat{k''_1, k''_2}, i-1, X', Y'] \circ e$.

Also by our assumption, if $e \in E(G)$, then $e \in E(G) \setminus E_b$. For ease of presentation, let us call $B = \widehat{B'} \cup \{e\}$. Furthermore, let $D = (B \cup Z) \cap E(G)$ and $A = (B \cup Z) \cap \overline{E(G)}$. Now we have that

- $B \in \mathcal{S}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y]$ and
- ψ' is a proper deficiency map over $S(G - ((B \cup Z) \cap E(G)), f)$ such that $E_{\psi'} = (B \cup Z) \cap \overline{E(G)}$.

Let $g : \mathsf{Green} \cup \mathcal{C} \to \{0, 1\}$ be defined as follows:

1. For any $v \in \mathsf{Green}$, $g(v) = 1$ if and only if there exists a path from v to a vertex in $\mathsf{def}(G - (B \cap E(G)), f)$ in $G[E_b \cup (E_r \setminus B)]$.
2. For any $C \in \mathcal{C}$, $g(C) = 1$ if and only if $B \cap E(C) \neq \emptyset$.

From the definition of $\mathcal{A}_g$, we have that $B \in \mathcal{A}_g$. Since $B \cup Z$ is an independent set in the matroid M, by the definition of representative families, there exists $\widehat{B} \in \widehat{\mathcal{A}}_g$ such that $\widehat{B} \cap Z = \emptyset$ and $\widehat{B} \cup Z$ is an independent set in M. Note that $\widehat{B} \in \mathcal{D}[T'_m, T'_g, \widehat{T'_r, k'_1}, k'_2, i, X, Y]$ and $\widehat{B} \cap E_b = \emptyset$. To conclude the proof of the lemma, the only thing that remains to show is that $(\widehat{B} \cup Z) \cap E(G)$ satisfies all of the five properties of a nice deletion set. The next claim does this job.

CLAIM 11. *$(\widehat{B} \cup Z) \cap E(G)$ satisfies five properties of a nice deletion set, and there exists a proper deficiency map $\widehat{\psi}$ over $S(G - ((\widehat{B} \cup Z) \cap E(G)), f)$ such that $E_{\widehat{\psi}} = (\widehat{B} \cup Z) \cap \overline{E(G)}$.*

Proof. Let $\widehat{D} = (\widehat{B} \cup Z) \cap E(G)$ and $\widehat{A} = (\widehat{B} \cup Z) \cap \overline{E(G)}$. We know that D satisfies five properties of a nice deletion set, and there exists a proper deficiency map ψ' over $S(G - D, f)$ with the property that $E_{\psi'} = A$. We need to show that $\widehat{D}$ satisfies five properties of a nice deletion set and there exists a proper deficiency map $\widehat{\psi}$ over $S(G - \widehat{D}, f)$ with the property that $E_{\widehat{\psi}} = \widehat{A}$.

Property (i). We know that $B, \widehat{B} \in \mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y]$ because

$$\widehat{\mathcal{A}}_g \subseteq \mathcal{A}_g \subseteq \mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y].$$

Hence, for all $v \in \mathsf{Green}$, $d_{G-(\widehat{B} \cap E(G))}(v) = d_{G-(B \cap E(G))}(v)$. This implies that for all $v \in \mathsf{Green}$, $d_{G-\widehat{D}}(v) = d_{G-D}(v)$. Since for all $v \in \mathsf{Green}$, $d_{G-D}(v) \leq f(v)$, we have that for all $v \in \mathsf{Green}$, $d_{G-\widehat{D}}(v) \leq f(v)$. For any $v \in \mathsf{Red}$, $d_{G-\widehat{D}} \leq f(v)$. Hence $\widehat{D}$ satisfies property (i) of a nice deletion set.

Property (ii). Since $B, \widehat{B} \in \mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y]$, we have that $|S(G - D, f)| = |S(G - \widehat{D}, f)|$ and $|D| = |\widehat{D}| = k'$. Thus $\widehat{D}$ satisfies property (ii) of a nice deletion set.

Property (iii). We know that $\widehat{B} \cup Z$ is an independent set in the matroid M, and hence $\widehat{D}$ is an independent set in $M_G(\ell)$, where $\ell = E(G) - V(G) + k - k' + 1$. Thus, by Lemma 4.8, $G - \widehat{D}$ has at most $k - k' + 1 = k - |\widehat{D}| + 1$ connected components, and so $\widehat{D}$ satisfies property (iii) of a nice deletion set.

Property (iv). Now we consider property (iv) of a nice deletion set. We need to show that for any connected component C in $G - \widehat{D}$, $V(C) \cap \mathsf{def}(G - \widehat{D}, f) \neq \emptyset$.

Case 1. Suppose $V(C) \cap \mathsf{Green} \neq \emptyset$. Let $v \in V(C) \cap \mathsf{Green}$. Suppose $g(v) = 1$. Since $\widehat{B} \in \mathcal{A}_g$, there exists a path P in $G[(E_b \cup (E_r \setminus \widehat{B})]$ from v to a vertex u in $\mathsf{def}(G - (\widehat{B} \cap E(G)), f)$. This implies that $u \in \mathsf{def}(G - \widehat{D}, f)$. If $E(P) \subseteq E(G - \widehat{D})$, then $u \in V(C)$. This implies that $V(C) \cap \mathsf{def}(G - \widehat{D}) \neq \emptyset$. Suppose $E(P) \nsubseteq E(G - \widehat{D})$. This implies that some of the edges in $E(P)$ are present in $\widehat{D}$. Since for all $e \in \widehat{D}$, $e \in E_o$ and for all $e' \in E_g \cap E(P)$, $e' \in E_b$, we have that any edge $e \in E(P) \cap \widehat{D}$ also belongs to E_r. Let $e_1 = (u_1, v_1)$ be the first edge in the path P, such that $e_1 \in \widehat{D}$. Note that $u_1 \in V(C)$ and $d_{G-\widehat{D}}(u_1) < f(u_1)$, because $u_1 \in \mathsf{Red}$. Hence $V(C) \cap \mathsf{def}(G - \widehat{D}) \neq \emptyset$.

Now we consider the case $g(v) = 0$. We know that there is a path P in $G - D$ from the vertex v to a vertex u such that $u \in \mathsf{def}(G - D, f)$ and $E(P) \cap E_g \subseteq E_b$. This implies that P is a path in $G[E_b \cup (E_r \setminus D)]$. Since $g(v) = 0$ and P is a path in $G[E_b \cup (E_r \setminus D)]$, we have that $d_{G-(B\cap E(G))}(u) \geq f(u)$. If $u \in \mathsf{Green}$, then $d_{G-\widehat{D}}(u) = d_{G-D}(u) < f(u)$. If $u \in \mathsf{Red}$, then there is $e = (u, w) \in Z \cap E(G)$. This implies that $d_{G-\widehat{D}}(u) < f(u)$ because $d_G(u) \leq f(u)$ and $(u, w) \in \widehat{D}$. In either case $u \in \mathsf{def}(G - \widehat{D}, f)$. If $E(P) \subseteq E(G - \widehat{D})$, then $u \in V(C)$. This implies that $V(C) \cap \mathsf{def}(G - \widehat{D}) \neq \emptyset$. Suppose $E(P) \nsubseteq E(G - \widehat{D})$. This implies that some of the edges in $E(P)$ are present in $\widehat{D}$. Since for all $e \in \widehat{D}$, $e \in E_o$ and for all $e' \in E_g \cap E(P_i)$, $e' \in E_b$, any edge $e \in E(P) \cap \widehat{D}$ also belongs to E_r. Let $e_1 = (x, y)$ be the first edge in the path P, such that $e_1 \in \widehat{D}$. Note that $x \in V(C)$ and $d_{G-\widehat{D}}(x) < f(x)$, because $x \in \mathsf{Red}$. Hence $V(C) \cap \mathsf{def}(G - \widehat{D}) \neq \emptyset$.

Case 2. Suppose $V(C) \subseteq \mathsf{Red}$ and there exists a connected component F in $\mathcal{C}$ such that $V(F) \cap V(C) \neq \emptyset$. If $g(F) = 1$, then we know that $\widehat{B} \cap E(G)$ contains an edge which is also present in $E(F)$, because $\widehat{B} \in \mathcal{A}_g$. Then, by Lemma 4.9 we have that $V(C) \cap \mathsf{def}(G - \widehat{D}) \neq \emptyset$. If $g(F) = 0$, then there exists an edge e in $Z \cap E(G)$ such that $e \in E(F)$, because D satisfies properties of Lemma 4.4 and $B \in \mathcal{A}_g$. This implies that, by Lemma 4.9, we have that $V(C) \cap \mathsf{def}(G - \widehat{D}) \neq \emptyset$.

Case 3. Suppose $V(C) \subseteq \mathsf{Red}$ and for all connected component F in $\mathcal{C}$, $V(F) \cap V(C) = \emptyset$. Then there exists a path P from a vertex v in $V(C)$ to a vertex u in Green in graph G. Let $e = (x, y)$ be the first edge in the path P such that $e \in \widehat{D}$. Note that such an edge e exists because $V(C) \subseteq \mathsf{Red}$ and $x \in \mathsf{Red}$. Since $x \in \mathsf{Red}$ and $(x, y) \in \widehat{D}$, we have that $d_{G-\widehat{D}}(x) < f(x)$. This implies that $V(C) \cap \mathsf{def}(G - \widehat{D}) \neq \emptyset$.

Property (v). Now consider property (v) of a nice deletion set. We need to show that there exists a proper deficiency map $\widehat{\psi}$ over $S(G - \widehat{D})$. We know that ψ' is a proper deficiency map over $S(G - D)$. Let

$$P = B \cap E(G), \quad \widehat{P} = \widehat{B} \cap E(G), \quad Q = B \cap \overline{E(G)} \quad \text{and} \quad \widehat{Q} = \widehat{B} \cap \overline{E(G)}.$$

We claim that for all $v \in V(G), d_{G-\widehat{P}+\widehat{Q}}(v) = d_{G-P+Q}(v)$. Since $\widehat{P} \cup \widehat{Q}, P \cup Q \in \mathcal{D}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y] \subseteq \mathcal{Q}(T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y)$, there exist proper deficiency maps ψ_1 over $(S(G - \widehat{P}, f) \cup T'_g \cup Y) \setminus (T'_r \cup X_{\widehat{P},f})$ and ψ_2 over $(S(G - P, f) \cup T'_g \cup Y) \setminus (T'_r \cup X_{P,f})$ such that $E_{\psi_1} = \widehat{Q}$ and $E_{\psi_2} = Q$. This implies that for any $v \in \mathsf{Red}$,

$$d_{G-\widehat{P}+\widehat{Q}}(v) = f(v) + Y(v) - T'_r(v) - X(v) = d_{G-P+Q}(v).$$

For any $v \in \mathsf{Green}$, since $d_{G-P}(v) = d_{G-\widehat{P}}(v)$ and ψ_1, ψ_2 are proper deficiency maps over $(S(G - \widehat{P}, f) \cup T'_g \cup Y) \setminus (T'_r \cup X_{\widehat{P},f})$, $(S(G - P, f) \cup T'_g \cup Y) \setminus (T'_r \cup$

$X_{P,f}$), respectively, we have that $d_{G-P+Q}(v) = d_{G-\widehat{P}+\widehat{Q}}(v)$. Hence, for all $v \in V(G), d_{G-\widehat{P}+\widehat{Q}}(v) = d_{G-P+Q}(v)$.

Now we claim that for all $v \in V(G), d_{G-\widehat{D}+\widehat{A}}(v) = d_{G-D+A}(v)$,

$$\begin{aligned} d_{G-\widehat{D}+\widehat{A}}(v) &= d_{G-\widehat{P}+\widehat{Q}}(v) - |E_G(v) \cap Z| + |\overline{E}_G(v) \cap Z| \\ &= d_{G-P+Q}(v) - |E_G(v) \cap Z| + |\overline{E}_G(v) \cap Z| \\ &= d_{G-D+A}(v). \end{aligned}$$

We have that ψ' is a proper deficiency map over $S(G-D), f)$ such that $E_{\psi'} = A$. This implies that $d_{G-D+A}(v) = f(v)$ for all $v \in V(G)$. Since $d_{G-\widehat{D}+\widehat{A}}(v) = d_{G-D+A}(v)$, we have that $d_{G-\widehat{D}+\widehat{A}}(v) = f(v)$ for all $v \in V(G)$. Let $\widehat{A} = \{e_1, e_2, \dots, e_r\}$, where $r = k - |D|$. Since for all $v \in V(G)$, $d_{G-\widehat{D}+\widehat{A}}(v) = f(v)$, we have that there are exactly $f(v) - d_{G-D}(v)$ edges in $\widehat{A}$ which are adjacent to v. Now we define a function $\widehat{\psi} : S(G - \widehat{D}, f) \to S(G - \widehat{D}, f)$ as follows. $\widehat{\psi}(u(i)) = v(j)$ if $(u, v) = e_\ell$ such that there are exactly $i - 1$ edges from $\{e_1, \dots, e_{\ell-1}\}$ are incident with u and there are exactly $j - 1$ edges from $\{e_1, \dots, e_{\ell-1}\}$ are incident with v. By Claim 1 of Lemma 4.4 we have that $\widehat{\psi}$ is a proper deficiency map. Since we constructed $\widehat{\psi}$ from $\widehat{A}$, $E_{\widehat{\psi}} = \widehat{A}$. ∎

The proof of the above claim completes the proof of the lemma. ∎

So our algorithm computes $\mathcal{S}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y]$ using (6), (7), and (8) from DP table entries computed for $i-1$ and then computes $\mathcal{D}[T'_m, T'_g, \widehat{T'_r, k'_1}, k'_2, i, X, Y]$ as explained above. If there exists $B \in \mathcal{D}[T_m, T_g, \widehat{\emptyset, k_1}, k_2, k, \emptyset, \emptyset]$ such that $B \cap E(G)$ is a nice deletion set, then the algorithm outputs YES. Otherwise, the algorithm outputs NO. The correctness of the algorithm follows from Lemmas 4.4 and 4.12.

Running time. Let $|V(G)| = n$ and let $E(G) = m$. Then the rank of the matroid M is bounded by $m + k$. Consider the construction of $\mathcal{D}[T'_m, T'_g, \widehat{T'_r, k'_1}, k'_2, i, X, Y]$. First, we constructed $\mathcal{S}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y]$ using (6), (7), or (8) from DP table entries $\mathcal{D}[T''_m, T''_g, T''_r, \widehat{k''_1, k''_2}, i-1, X', Y']$. Thus the size of $\mathcal{S}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y]$ is

$$\mathcal{O}\left(\max_{T''_m, T''_g, T''_r, k''_1, k''_2, X', Y'} |\mathcal{D}[T''_m, T''_g, T''_r, \widehat{k''_1, k''_2}, i-1, X', Y']| \cdot \binom{n}{2} \right).$$

We know that $\mathcal{D}[T'_m, T''_g, T''_r, \widehat{k''_1, k''_2}, i-1, X', Y'] = \bigcup_{g \in \mathcal{J}} \widehat{\mathcal{A}}_g$, where $\widehat{\mathcal{A}}_g$ is a $(k - (i-1))$-representative family computed using Theorem 2.6. Thus, by Theorem 2.6, $|\widehat{\mathcal{A}}_g|$ is bounded by $(m+k)k\binom{k}{i-1}$. The cardinality of $\mathsf{Green} \cup \mathcal{C}$ is bounded by $2k$, because any solution should contain at least one edge incident with each green vertex and one edge from each component in $\mathcal{C}$. Hence $|\mathcal{J}| = 4^k$. Thus, the size of $\mathcal{S}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y]$ is upper bounded by $4^k\binom{k}{i-1}(m+k)k \log n = 4^k\binom{k}{i-1}n^{\mathcal{O}(1)}$.

Then, we have partitioned $\mathcal{S}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y]$ based on $g \in \mathcal{J}$; that is, $\mathcal{S}[T'_m, T'_g, T'_r, k'_1, k'_2, i, X, Y] = \bigcup_{g \in \mathcal{J}} \mathcal{A}$ and then computed a $k{-}i$-representative family $\widehat{\mathcal{A}}_g$ of $\mathcal{A}_g$ for each $g \in \mathcal{J}$. By Theorem 2.6, the running time of this computation is upper bounded by

$$4^k \binom{k}{i-1} \binom{k}{i}^{\omega-1} n^{\mathcal{O}(1)}.$$

Thus, the running time to compute $\mathcal{D}[T'_m, T'_g, \widehat{T'_r, k'_1}, k'_2, i, X, Y]$ is bounded by

$$4^k \binom{k}{i-1} \binom{k}{i}^{\omega-1} n^{\mathcal{O}(1)}.$$

The cardinality of $T_m \cup T_g \cup T_r$ is at most $2k$; otherwise, we need more than k edges in the solution. Thus, the running time of the algorithm, once we guessed k', k_1, k_2 and $\Phi(v)$ for all v, is upper bounded by

$$\sum_{i=1}^{k} 2^{2k} 4^k \binom{k}{i-1} \binom{k}{i}^{\omega-1} n^{\mathcal{O}(1)} = 2^{(4+\omega)k+o(k)} n^{\mathcal{O}(1)}.$$

Since the number of possible guesses for k', k_1, k_2, and Φ is at most $4^k k^{\mathcal{O}(1)}$, the total running time of the algorithm is $2^{(6+\omega)k} n^{\mathcal{O}(1)}$. Also note that we run the entire algorithm $\left(\frac{7^7}{6^6}\right)^k$ time to improve the success probability to at least $(1 - 1/e)$.

THEOREM 4.13. *There is a randomized algorithm for* EECG *running in time* $\left(\frac{7^7}{6^6}\right)^k 2^{(6+\omega)k} n^{\mathcal{O}(1)}$ *with success probability to at least* $(1 - 1/e)$.

4.4. Derandomization. In this subsection we explain how to derandomize the above algorithm. Our algorithm can be derandomized using *n-p-q-lopsided-universal* family introduced in [8]. A family $\mathcal{F}$ of sets over a universe U of size n is an *n-p-q-lopsided-universal* family if for every $A \in \binom{U}{p}$ and $B \in \binom{U \setminus A}{q}$ there is an $F \in \mathcal{F}$ such that $A \subseteq F$ and $B \cap F = \emptyset$. It turns out that by slightly changing the construction of Naor, Schulman, and Srinivasan [19], one can prove the following lemma and it can also be derived as a corollary of a result from [8].

LEMMA 4.14 (see [19, 8]). *There is an algorithm that, given n, p, and q, constructs an n-p-q-lopsided-universal family* $\mathcal{F}$ *of size* $\binom{p+q}{p} \cdot 2^{o(p+q)} \cdot \log n$ *in time* $\mathcal{O}(\binom{p+q}{p} \cdot 2^{o(p+q)} \cdot n \log n)$.

To derandomize our algorithm, instead of randomly coloring the edges in E_g, we use $(|E_g|, 7k, k)$-lopsided-universal family $\mathcal{F}$. We run our algorithm $|\mathcal{F}|$ many times as follows. For each $F \in \mathcal{F}$, we color F with orange and $E_g \setminus F$ with black and run our algorithm. The correctness of derandomization follows from the definition of n-$7k$-k-lopsided-universal family.

FACT 1. *By Stirling's approximation,* $\binom{k}{\alpha k} \le \left(\alpha^{-\alpha}(1-\alpha)^{(\alpha-1)}\right)^k$ [23].

Thus, by using Lemma 4.14, we can derandomize our algorithm and we get the following theorem where its running time follows from Fact 1.

THEOREM 4.15. *There is a deterministic algorithm for* EECG *running in time* $\left(\frac{7^7}{6^6}\right)^k 2^{(6+\omega)k} n^{\mathcal{O}(1)}$.

5. Hardness. In this section we prove hardness of EDGE EDITING TO CONNECTED f-DEGREE GRAPH WITH COSTS.

THEOREM 5.1. EDGE EDITING TO CONNECTED f-DEGREE GRAPH WITH COSTS *is W[1]-hard for trees when parameterized by* $k + d$ *even if costs are restricted to* 0 *or* 1.

Proof. We reduce the CLIQUE problem that is well known to be W[1]-complete [6]. In this problem we are given an undirected graph G and a positive integer k as an input and the objective is to check whether G has a clique of size at least k. It is straightforward to observe that Clique is W[1]-complete for the instances where k is restricted to be odd. To see this, it is sufficient to notice that if G' is the graph obtained from a graph G by adding a vertex that is adjacent to every other vertex of G, then G has a clique of size k if and only if G' has a clique of size $k+1$.

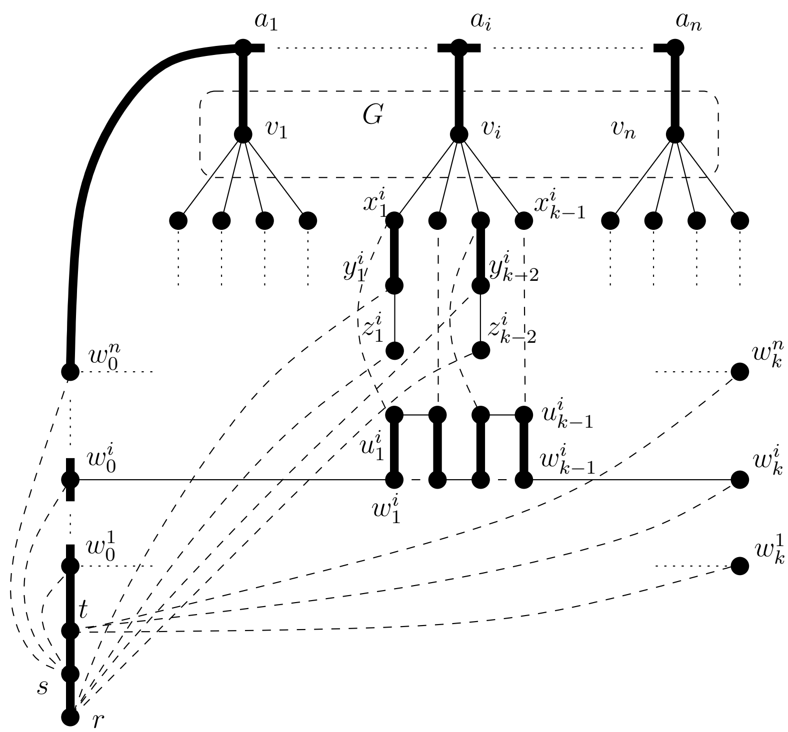

Fig. 5.1. *Construction of T. The edges of cost 0 are shown by thin lines, the edges of cost 1 are shown by thick lines, and the non-edges of cost 0 are shown by dashed lines. Notice that the graph G is encoded by assigning the cost 0 to every non-edge of T corresponding to an edge of G.*

Let (G,k) be an instance of Clique and $k \geq 3$ is odd. Let $V(G) = \{v_1, \ldots, v_n\}$. We construct a tree T and define the function c as follows (see Figure 5.1):

(i) Construct vertices $v_1, \ldots, v_n$ and set $c((v_i, v_j)) = 0$ if $(v_i, v_j) \in E(G)$ for $i, j \in [n]$.

(ii) For each $i \in [n]$, construct vertices a_i, $x_1^i, \ldots, x_{k-1}^i$, $y_1^i, y_3^i, y_5^i, \ldots, y_{k-2}^i$, and $z_1^i, z_3^i, z_5^i \ldots, z_{k-2}^i$ and edges (a_i, v_i), $(v_i, x_1^i), \ldots, (v_i, x_{k-1}^i)$, $(x_1^i, y_1^i), (x_3^i, y_3^i)$, $(x_5^i, y_5^i), \ldots, (x_{k-2}^i, y_{k-2}^i)$ and $(y_1^i, z_1^i), (y_3^i, z_3^i), (y_5^i, z_5^i), \ldots, (y_{k-2}^i, z_{k-2}^i)$. We set

$$\begin{aligned} c((a_i, v_i)) &= 1, \\ c((v_i, x_1^i)) = \cdots = c((v_i, x_{k-1}^i)) &= 0, \\ c((x_1^i, y_1^i)) = c((x_3^i, y_3^i)) = \cdots = c((x_{k-2}^i, y_{k-2}^i)) &= 1, \text{ and} \\ c((y_1^i, z_1^i)) = c((y_3^i, z_3^i)) = \cdots = c((y_{k-2}^i, z_{k-2}^i)) &= 0. \end{aligned}$$

(iii) For each $i \in [n]$, construct vertices $u_1^i, \ldots, u_{k-1}^i$ and $w_0^i, \ldots, w_k^i$ and edges $(u_1^i, w_1^i), \ldots, (u_{k-1}^i, w_{k-1}^i)$, $(w_0^i, w_1^i), (w_2^i, w_3^i), \ldots, (w_{k-1}^i, w_k^i)$, and (u_1^i, u_2^i),

$(u_3^i, u_4^i), \ldots, (u_{k-2}^i, u_{k-1}^i)$. We set

$$c((u_1^i, w_1^i)) = \cdots = c((u_{k-1}^i, w_{k-1}^i)) = 1,$$
$$c((w_0^i, w_1^i)) = c((w_2^i, w_3^i)) = \cdots = c((w_{k-1}^i, w_k^i)) = 0,$$
$$c((u_1^i, u_2^i)) = c((u_3^i, u_4^i)) = \cdots = c((u_{k-2}^i, u_{k-1}^i)) = 0, \text{ and}$$
$$c((w_1^i, w_2^i)) = c((w_3^i, w_4^i)) = \cdots = c((w_{k-2}^i, w_{k-1}^i)) = 0.$$

(iv) For each $i \in [n]$, set $c((u_1^i, x_1^i)) = \cdots = c((u_{k-1}^i, x_{k-1}^i)) = 0$.

(v) Construct vertices s, t, r and edges $(r, s), (s, t), (t, w_0^1)$, $(w_0^1, w_0^2), (w_0^2, w_0^3), \ldots,$ (w_0^{n-1}, w_0^n), (w_0^n, a_1) and $(a_1, a_2), (a_2, a_3), \ldots, (a_{n-1}, a_n)$. We set

$$c((r, s)) = c((s, t)) = c((t, w_0^1)) = 1,$$
$$c((w_0^1, w_0^2)) = c((w_0^2, w_0^3)) = \ldots = c((w_0^{n-1}, w_0^n)) = 1,$$
$$c((w_0^n, a_1)) = 1, \text{ and}$$
$$c((a_1, a_2)) = c((a_2, a_3)) = \ldots = c((a_{n-1}, a_n)) = 1.$$

(vi) Set $c((s, w_0^1)) = \cdots = c((s, w_0^n)) = 0$, $c((t, w_k^1)) = \cdots = c((t, w_k^n)) = 0$.

(vii) For each $i \in [n]$, set $c((r, y_1^i)) = c((r, y_3^i)) = \cdots = c((r, y_{k-2}^i)) = 0$ and $c((r, z_1^i)) = c((r, z_3^i)) \cdots = c((r, z_{k-2}^i)) = 0$.

(viii) For any $(p, q) \in \binom{V(T)}{2} \setminus E(T)$, set $c((p, q)) = 1$ if $c((p, q))$ was not set to be 0 in (i)–(vii).

We define $f(s) = k+2$, $f(t) = k+2$, $f(r) = 1+k(k-1)$, and set $f(p) = d_T(p)$ for $p \in V(G)\setminus\{s, t, r\}$. Finally, we set $C = 0$, $d = 2+k(k-1)$, and $k' = 5k^2-2k+k(k-1)/2$, and obtain an instance (T, d, k', C, f, c) of Edge Editing to Connected f-Degree Graph with Costs. Clearly, T is a tree. We show that (T, d, k', C, f, c) is a Yes-instance of Edge Editing to Connected f-Degree Graph with Costs if and only if G has a clique of size k.

For $i \in \{1, \ldots, n\}$, let

$$D_i = \{(w_0^i, w_1^i), (w_2^i, w_3^i), \ldots, (w_{k-1}^i, w_k^i)\} \cup \{(u_1^i, u_2^i), (u_3^i, u_4^i), \ldots, (u_{k-2}^i, u_{k-1}^i)\}$$
$$\cup \{(v_i, x_1^i), \ldots, (v_i, x_{k-1}^i)\} \cup \{(y_1^i, z_1^i), (y_3^i, z_3^i), \ldots, (y_{k-2}^i, z_{k-2}^i)\}$$

and

$$A_i = \{(s, w_0^i), (t, w_k^i)\} \cup \{(w_1^i, w_2^i), (w_3^i, w_4^i), \ldots, (w_{k-2}^i, w_{k-1}^i)\}$$
$$\cup \{(x_1^i, u_1^i), \ldots, (x_{k-1}^i, u_{k-1}^i)\}$$
$$\cup \{(r, y_1^i), (r, y_3^i), \ldots, (r, y_{k-2}^i)\} \cup \{(r, z_1^i), (r, z_3^i), \ldots, (r, z_{k-2}^i)\}.$$

Note that $D_i \subseteq E(T)$, $c(D_i) = 0$, $|D_i| = 2k-1+(k-1)/2$, and $A_i \subseteq \binom{V(T)}{2} \setminus E(T)$, $c(A_i) = 0$, $|A_i| = 2k+(k-1)/2$ for $i \in \{1, \ldots, n\}$.

Suppose that G has a clique $K = \{v_{i_1}, \ldots, v_{i_k}\}$. Let $A' = \{(v_{i_j}, v_{i_h}) | 1 \leq j < h \leq k\}$. Because K is a clique in G, $c(A') = 0$. Clearly, $A' \subseteq \binom{V(T)}{2} \setminus E(T)$ and $|A'| = k(k-1)/2$. We let $D = \bigcup_{j=1}^k D_{i_j}$ and $A = A' \cup (\bigcup_{j=1}^k A_{i_j})$. It is straightforward to verify that $c(D \cup A) = 0$, $|D| + |A| = k'$, $G' = T - D + A$ is a connected graph, and for every $p \in V(G')$, $d_{G'}(p) = f(p)$.

Assume now that (T, d, k', C, f, c) is a Yes-instance of Edge Editing to Connected f-Degree Graph with Costs. Then there are sets $D \subseteq E(T)$ and

$A \subseteq \binom{V(T)}{2}$ such that $|D| + |A| \leq k'$, $c(D \cup A) = 0$, $G' = T - D + A$ is a connected graph, and for every $p \in V(G')$, $d_{G'}(p) = f(p)$. Because $f(s) = k + 2$ and $d_T(s) = 2$, A contains at least k edges incident to s. Since $c(A) = 0$, we have that there are $sw_0^{i_1}, \ldots, sw_0^{i_k} \in A$ for some distinct $i_1, \ldots, i_k \in \{1, \ldots, n\}$.

Consider $(s, w_0^{i_j})$ for some $j \in \{1, \ldots, k\}$. Because $f(w_0^{i_j}) = d_T(w_0^{i_j})$, D has an edge of cost 0 incident to $w_0^{i_j}$. Hence, $(w_0^{i_j}, w_1^{i_j}) \in D$. Now we consider $w_1^{i_j}$ and observe that there is an edge of cost 0 in A that is incident to $w_1^{i_j}$ and, therefore, $(w_1^{i_j}, w_2^{i_j}) \in A$. Repeating these arguments, we conclude that

$$R = \{(w_0^{i_j}, w_1^{i_j}), (w_2^{i_j}, w_3^{i_j}), \ldots, (w_{k-1}^{i_j}, w_k^{i_j})\} \subseteq D$$

and

$$S = \{(s, w_0^{i_j}), (w_1^{i_j}, w_2^{i_j}), \ldots, (w_{k-2}^{i_j}, w_{k-1}^{i_j}), (w_k^{i_j}, t)\} \subseteq A.$$

Consider $F = T - R + S$. Observe that for $h \in [\frac{k-1}{2}]$, $F[\{w_{2h-1}^{i_j}, w_{2h}^{i_j}, u_{2h-1}^{i_j}, u_{2h}^{i_j}\}]$ is a component of F. Since G' is connected, A has an edge incident to a vertex of each component of this type. We have that for $h \in [\frac{k-1}{2}]$, $(x_{2h-1}^{i_j}, u_{2h-1}^{i_j}) \in A$ or $(x_{2h}^{i_j}, u_{2h}^{i_j}) \in A$. As $f(u_{2h-1}^{i_j}) = d_T(u_{2h-1}^{i_j})$ and $f(u_{2h}^{i_j}) = d_T(u_{2h}^{i_j})$, D has an edge incident to one of these vertices and, therefore, $(u_{2h-1}^{i_j}, u_{2h}^{i_j}) \in D$ and $(x_{2h-1}^{i_j}, u_{2h-1}^{i_j}), (x_{2h}^{i_j}, u_{2h}^{i_j}) \in A$. Because $f(x_{2h-1}^{i_j}) = d_T(x_{2h-1}^{i_j})$, $(x_{2h-1}^{i_j}, v_{i_j}), (x_{2h}^{i_j}, v_{i_j}) \in D$. We obtain that

$$R' = R \cup \{(u_1^{i_j}, u_2^{i_j}), (u_3^{i_j}, u_4^{i_j}), \ldots, (u_{k-2}^{i_j}, u_{k-1}^{i_j})\} \cup \{(v_{i_j}, x_1^{i_j}), \ldots, (v_{i_j}, x_{k-1}^{i_j})\} \subseteq D$$

and

$$S' = S \cup \{(x_1^{i_j}, u_1^{i_j}), \ldots, (x_{k-1}^{i_j}, u_{k-1}^{i_j})\} \subseteq A.$$

Let $F' = T - R' + S'$. Now we have that for $h \in [\frac{k-1}{2}]$, the induced subgraph $F'[\{w_{2h-1}^{i_j}, w_{2h}^{i_j}, u_{2h-1}^{i_j}, u_{2h}^{i_j}, x_{2h-1}^{i_j}, x_{2h}^{i_j}, y_{2h-1}^{i_j}, z_{2h-1}^{i_j}\}]$ is a component of F'. Because G' is connected, $(r, y_{2h-1}^{i_j}) \in A$ or $(r, z_{2h-1}^{i_j}) \in A$. As $f(y_{2h-1}^{i_j}) = d_T(y_{2h-1}^{i_j})$ and $f(z_{2h-1}^{i_j}) = d_T(z_{2h-1}^{i_j})$, $(y_{2h-1}^{i_j}, z_{2h-1}^{i_j}) \in D$ and $(r, y_{2h-1}^{i_j}), (r, z_{2h-1}^{i_j}) \in A$. We conclude that $D_{i_j} \subseteq D$ and $A_{i_j} \subseteq A$. Let

$$R'' = \bigcup_{j=1}^{k} D_{i_j}, \quad S'' = \bigcup_{j=1}^{k} A_{i_j}.$$

We have that $R'' \subseteq D$ and $S'' \subseteq A$. Note that $|R''| + |S''| = 5k^2 - 2k = k' - k(k-1)/2$. Consider $F'' = T - R'' + S''$. For $j \in \{1, \ldots, k\}$, $d_{F''}(v_{i_j}) = d_T(v_{i_j}) - (k-1)$. It implies that $A' = \{(v_{i_j}, v_{i_h}) | 1 \leq j < h \leq k\} \subseteq A$ and $c(A') = 0$. Hence, $K = \{v_{i_1}, \ldots, v_{i_k}\}$ is a clique in G. ∎

6. Conclusion. In this paper we showed that editing to a connected graph satisfying degree constraints given by a function f is FPT. In particular, we showed that Edge Editing to Connected f-Degree Graph admits an algorithm with running time $2^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$. We complemented this result by showing that the weighted version of the problem with costs 1 and 0 is W[1]-hard being parameterized by k and the maximum value taken by f even when the input graph is a tree. Our algorithm combined the ideas of color-coding and the representative family based approach to obtain our FPT algorithm. We believe that this approach could act as a template for solving other edge editing problems. Finally, we would like to point out that we used a nonstandard matroid, namely, ℓ-elongation of co-graphic matroid to get our desired algorithm. These matroid based methodology for designing parameterized algorithm hold a lot of promise and it seems the area is still pretty much unexplored. Finally, we conclude with the following interesting question: Does Edge Editing to Connected f-Degree Graph admit a polynomial kernel?

REFERENCES

[1] J. Akiyama and M. Kano, *Factors and factorizations of graphs—a survey*, J. Graph Theory, 9 (1985), pp. 1–42.

[2] N. Alon, D. Lokshtanov, and S. Saurabh, *Fast FAST*, in Proceedings of the 36th International Colloquium of Automata, Languages and Programming (ICALP), Lecture Notes in Comput. Sci. 5555 , Springer, Berlin, 2009, pp. 49–58.

[3] R. P. Anstee, *An algorithmic proof of Tutte's f-factor theorem*, J. Algorithms, 6 (1985), pp. 112–131.

[4] R. Chitnis, M. Cygan, M. Hajiaghayi, M. Pilipczuk, and M. Pilipczuk, *Designing FPT algorithms for cut problems using randomized contractions*, SIAM J. Comput., 45 (2016), pp. 1171–1229, https://doi.org/10.1137/15M1032077.

[5] R. Diestel, *Graph Theory*, 3rd ed., Grad. Texts in Math. 173, Springer-Verlag, Berlin, 2005.

[6] R. G. Downey and M. R. Fellows, *Fundamentals of Parameterized Complexity*, Texts Comput. Sci., Springer, London, 2013.

[7] H. Fleischner, *Eulerian Graphs and Related Topics*, Part 1, Volume 1, Ann. Discrete Math. 45, North-Holland Publishing, Amsterdam, 1990.

[8] F. V. Fomin, D. Lokshtanov, F. Panolan, and S. Saurabh, *Efficient computation of representative families with applications in parameterized and exact algorithms*, J. ACM, 63 (2016), 29.

[9] F. V. Fomin and Y. Villanger, *Subexponential parameterized algorithm for minimum fill-in*, SIAM J. Comput., 42 (2013), pp. 2197–2216, https://doi.org/10.1137/11085390X.

[10] P. A. Golovach, *Editing to a connected graph of given degrees*, Inform. and Comput., 256 (2017), pp. 131–147.

[11] P. Goyal, P. Misra, F. Panolan, G. Philip, and S. Saurabh, *Finding even subgraphs even faster*, J. Comput. System Sci., 97 (2018), pp. 1–13.

[12] K. Kawarabayashi and M. Thorup, *The minimum k-way cut of bounded size is fixed-parameter tractable*, in Proceedings of the 52nd Annual Symposium on Foundations of Computer Science (FOCS), IEEE, Los Alamitos, CA, 2011, pp. 160–169.

[13] T. Kirkman, *On a problem in combinatorics*, Cambridge and Dublin Math. J., 2 (1847), pp. 191–204.

[14] M. Kouider and P. D. Vestergaard, *Connected factors in graphs - a survey*, Graphs Combin., 21 (2005), pp. 1–26.

[15] D. Lokshtanov, P. Misra, F. Panolan, and S. Saurabh, *Deterministic truncation of linear matroids*, ACM Trans. Algorithms, 14 (2018), 14.

[16] L. Lovász and M. D. Plummer, *Matching Theory*, AMS Chelsea Publishing, Providence, RI, 2009.

[17] D. Marx, *A parameterized view on matroid optimization problems*, Theoret. Comput. Sci., 410 (2009), pp. 4471–4479.

[18] L. Mathieson and S. Szeider, *Editing graphs to satisfy degree constraints: A parameterized approach*, J. Comput. System Sci., 78 (2012), pp. 179–191.

[19] M. Naor, L. J. Schulman, and A. Srinivasan, *Splitters and near-optimal derandomization*, in Proceedings of the 36th Annual Symposium on Foundations of Computer Science (FOCS), IEEE, Los Alamitos, CA, 1995, pp. 182–191.

[20] J. G. Oxley, *Matroid Theory*, 2nd ed., Oxf. Grad. Texts Math. 21, Oxford University Press, 2011.

[21] J. Petersen, *Die Theorie der regulären graphs*, Acta Math., 15 (1891), pp. 193–220.

[22] M. D. Plummer, *Graph factors and factorization:* 1985–2003*: A survey*, Discrete Math., 307 (2007), pp. 791–821.

[23] H. Robbins, *A remark on Stirling's formula*, Amer. Math. Monthly, 62 (1955), pp. 26–29.

[24] V. Vassilevska Williams, *Multiplying matrices faster than Coppersmith-Winograd*, in Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC), ACM, New York, 2012, pp. 887–898.