Vertex Exponential Algorithms for Connected f-Factors

Geevarghese Philip[1] and M. S. Ramanujan[2]

1 **Max-Planck-Institut für Informatik**
gphilip@mpi-inf.mpg.de
2 **University of Bergen**
ramanujan.sridharan@ii.uib.no

Abstract

Given a graph G and a function $f : V(G) \to [|V(G)|]$, an f-factor is a subgraph H of G such that $deg_H(v) = f(v)$ for every vertex $v \in V(G)$; we say that H is a *connected f-factor* if, in addition, the subgraph H is connected. Tutte (1954) showed that one can check whether a given graph has a specified f-factor in polynomial time. However, detecting a *connected* f-factor is NP-complete, even when f is a *constant* function – a foremost example is the problem of checking whether a graph has a Hamiltonian cycle; here f is a function which maps every vertex to 2. The current best algorithm for this latter problem is due to Björklund (FOCS 2010), and runs in randomized $\mathcal{O}^*(1.657^n)$ time (The $\mathcal{O}^*()$ notation hides polynomial factors). This was the first superpolynomial improvement, in nearly fifty years, over the previous best algorithm of Bellman, Held and Karp (1962) which checks for a Hamiltonian cycle in deterministic $\mathcal{O}(2^n n^2)$ time.

In this paper we present the first vertex-exponential algorithms for the more general problem of finding a connected f-factor. Our first result is a randomized algorithm which, given a graph G on n vertices and a function $f : V(G) \to [n]$, checks whether G has a connected f-factor in $\mathcal{O}^*(2^n)$ time. We then extend our result to the case when f is a mapping from $V(G)$ to $\{0, 1\}$ and the degree of every vertex v in the subgraph H is required to be $f(v)(mod\ 2)$. This generalizes the problem of checking whether a graph has an *Eulerian* subgraph; this is a connected subgraph whose degrees are all even ($f(v) \equiv 0$). Furthermore, we show that the *min-cost editing* and *edge-weighted* versions of these problems can be solved in randomized $\mathcal{O}^*(2^n)$ time as long as the costs/weights are bounded polynomially in n.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, G.2.1 Combinatorics, G.2.2 Graph Theory

Keywords and phrases Exact Exponential Time Algorithms, f-Factors

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2014.61

1 Introduction

The problem of testing whether an input graph has a Hamiltonian cycle – a simple cycle which passes through all vertices of the graph – is one of Karp's original list of 21 NP-complete problems [13], and is one of the most fundamental and well-studied problems in computational complexity. The current best algorithm for this problem is due to Björklund (FOCS 2010), and runs in randomized $\mathcal{O}^*(1.657^n)$ time(The $\mathcal{O}^*()$ notation hides polynomial factors.) [2]. This was the first superpolynomial improvement in nearly fifty years, over the previous best algorithm of Bellman [1], and Held and Karp [12] which checks for a Hamiltonian cycle in deterministic $\mathcal{O}(2^n n^2)$ time.

Another fundamental graph problem is that of deciding whether a given graph contains a regular subgraph. This problem was first stated by Garey and Johnson [9] who asked if

34th Int'l Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2014).
Editors: Venkatesh Raman and S. P. Suresh; pp. 61–71
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

testing the presence of a 3-regular subgraph in a given graph is NP-complete. This was shown to be NP-complete in a proof atrributed to Chvátal [9]. However, testing if a graph has a *spanning* r-regular subgraph is known to be polynomial time solvable by an application of Tutte's f-factor theorem [20], although testing for a *connected* spanning r-regular subgraph clearly generalizes Hamiltonicity.

Given a graph G and a function $f : V(G) \to [|V(G)|]$, an f-factor of G is a subgraph H of G such that $deg_H(v) = f(v)$ for every vertex $v \in V(G)$; we say that H is a *connected f-factor* if, in addition, the subgraph H is connected. Tutte [20] showed that one can check whether a given graph has a specified f-factor in polynomial time. Lovász [14, 15] extended this result to *general* f-factors, where the function f maps each vertex to a list of numbers. Lovász and Cournéjols [4] gave a complete characterization of the complexity of the general f-factor problem.

In this paper we study the problem of finding *connected* f-factors in a given graph. Our main motivation in investigating this problem is the fact that it generalizes the problem of testing for a Hamiltonian cycle in a graph, and also the more general problem of testing for regular connected spanning subgraphs.

Our results and techniques. Although the existence of a connected f-factor in a graph withm m edges can trivially be tested in time $\mathcal{O}^*(2^m)$, it was not known whether it is possible to solve this problem in time which is single-exponential in the number of vertices (a vertex-exponential algorithm) of the graph. In this paper we present the first vertex-exponential algorithms to find a connected f-factor in a given graph. In fact, we give a vertex-exponential algorithm for the *editing* version of this problem, which is much more general than the problem of simply finding a connected f-factor. More formally, in this problem, which we call Min-Cost Edge Editing to f-factor (Min-Cost EFF), the input consists of a graph G, a function $f : V(G) \to [n]$, a cost function c on the edges and non-edges of G, and a target cost $c^\star$. The objective is to check if there is a sequence of non-edge additions and edge deletions with a total cost at most $c^\star$ such that the resulting graph is a connected f-factor. This problem generalizes the problem of finding a connected f-factor in a graph, even with the additional restriction that the edge costs are bounded polynomially in the size of $V(G)$.

Our main result is a *randomized* algorithm which, given an instance $(G, f, c, c^\star)$ of Min-Cost EFF where $c(\{v, w\})$ is bounded by a polynomial in $|V(G)|$ for every $v, w \in E(G)$, solves it in time $\mathcal{O}^*(2^n)$.

▶ **Theorem 1.** *There is a randomized algorithm that, given an instance* $(G, c, c^\star)$ *of* Min-Cost Edge Editing to f-factor *with the cost function c being bounded by a polynomial in* $|V(G)|$*, runs in time* $\mathcal{O}^*(2^{|V(G)|})$ *and either returns a solution or correctly (with high probability) concludes that one does not exist.*

We then extend this result to a "parity version" Connected parity f-factor of the problem where, given a graph G and a function f where where f is a mapping from $V(G)$ to $\{0, 1\}$, the objective is to check if G has a connected spanning subgraph H where the degree of every vertex v in the subgraph H is $f(v) \pmod 2$.

▶ **Theorem 2.** *There is a randomized algorithm that, given an instance* (G, f) *of* Connected parity f-factor *where* $|V(G)| = n$*, runs in time* $\mathcal{O}^*(2^n)$ *and either returns a solution or correctly (with high probability) concludes that one does not exist.*

This generalizes the problem of checking whether a graph has an *Eulerian* subgraph; this is a connected subgraph whose degrees are all even ($f(v) \equiv 0$). As our third major result we show that the *edge-weighted* versions of finding connected (parity) f-factors can also be solved in randomized $\mathcal{O}^*(2^n)$ time as long as the weights are bounded polynomially in n.

The main technical ingredients in our solutions for each of these problems have the same flavour: For each problem, we transform the input graph into an auxiliary graph in such a way that the connected solutions which we seek correspond, in a certain sense, to the set of *perfect matchings* of the auxiliary graph. Our algorithms rely on the notion of Tutte matrices of graphs and related algebraic techniques – introduced by Lovász [16] and recently used in [2, 6, 21, 11] – to phrase our problems in terms of looking for "non-zero" monomials of certain polynomials. To solve these latter problems we test whether the polynomials are identically zero over certain fields. The randomization in our algorithm arises from this final step of polynomial identity testing.

Related work. Moser and Thilikos [18] and Mathieson and Szeider [17] initiated the study of the parameterized complexity of editing a given graph to obtain a graph that satisfies certain specified degree constraints. Mathieson and Szeider in particular described an auxiliary graph where perfect matchings captured the editing solutions in the same flavor of Tutte's auxiliary graph capturing f-factors via perfect matchings. More recently Golovach has studied the parameterized complexity of editing to connected graphs under degree constraints [10]. Cai and Yang [3], Cygan et al. [5] and Fomin and Golovach [8] have all studied the parameterized complexity of deleting edges to obtain subgraphs with parity constraints on the degrees.

Organization of the rest of the paper. In Section 2 we describe our notation and some preliminary results. In Section 3 we take up the edge-editing version of our problem, Min-Cost Edge Editing to f-factor, and prove Theorem 1. In Section 4 we take up the parity version Connected parity f-factor and prove Theorem 2. We conclude in Section 5.

2 Preliminaries

We follow the graph notation and terminology of Diestel [7]. For a positive integer n we use $[n]$ to denote the set $\{1, 2, \ldots, n\}$. We use A_{ij} to denote the element in the i^{th} row and j^{th} column of a matrix A. A subgraph H of a graph G is a *spanning subgraph* of G if $V(G) = V(H)$. We use $deg_G(v)$ to denote the degree of a vertex v in graph G and $N_G(v)$ to denote the neighbourhood of v in G; we omit the subscript when there is no scope for ambiguity. For a subset $S \subseteq V(G)$ of the vertex set of a graph G we use $G[S]$ to denote the subgraph *induced* by the set S and $G - S$ to denote the subgraph $G[V(G) \setminus S]$. For a subset S of vertices, we denote by $E(G)[S, V(G) \setminus S]$ the edges of G with an end point each in S and $V(G) \setminus S$. If F is a set of edges in a graph G, then we use $V(F)$ to denote the set of all vertices which form end-points of the edges in F. A *matching* in a graph G is any set M of edges in G such that no two edges of M have an end-point in common, and a matching M of G is a *perfect matching* if $V(M) = V(G)$. Let $w : E(G) \to \mathbb{Z}$ be function which assigns integer weights to the edges of a graph G. The *weight* of a subgraph H of G is then the sum $\sum_{e \in E(H)} w(e)$.

When we refer to expanded forms of *succinct* representations (such as summations and determinants) of polynomials, we use the term *naïve expansion* (or summation) to denote that expanded form of the polynomial which is obtained by merely writing out the operations indicated by the succinct representation. We use the term *simplified expansion* to denote the expanded form of the polynomial which results after we apply all possible simplifications (such as cancellations) to a naïve expansion. We call a monomial m which has a non-zero coefficient in a simplified expansion of a polynomial P, a *surviving* monomial of P in the simplified expansion.

▶ **Definition 3.** (Tutte matrix) The *Tutte matrix* of a graph G with n vertices is an $n \times n$ skew-symmetric matrix T over the set $\{x_{ij} | 1 \leq i < j \leq |V(G)|\}$ of indeterminates whose $(i,j)^{th}$ element is defined to be

$$T(i,j) = \begin{cases} x_{ij} & \text{if } \{i,j\} \in E(G) \text{ and } i < j \\ -x_{ji} & \text{if } \{i,j\} \in E(G) \text{ and } i > j \\ 0 & \text{otherwise} \end{cases}$$

We use $\mathcal{T}(G)$ to denote the Tutte matrix of graph G. We say that the variable x_{ij} is the *label* of edge $\{i,j\} \in E(G)$.

The following basic facts about the Tutte matrix $\mathcal{T}(G)$ of a graph G are well-known. When evaluated over any field of characteristic two, the determinant and the permanent of the matrix $\mathcal{T}(G)$ (indeed, of any matrix) coincide:

$$\det \mathcal{T}(G) = \text{perm}(\mathcal{T}(G)) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} \mathcal{T}(G)(i, \sigma(i)), \tag{1}$$

where S_n is the set of all *permutations* of $[n]$. Moreover, there is a one-to-one correspondence between the set of all *perfect matchings* of the graph G and the *surviving monomials* in the above expression for $\det \mathcal{T}(G)$ when its simplified expansion is computed over any field of characteristic two:

▶ **Proposition 1.** [16] *If $M = \{(i_1, j_1), (i_2, j_2), \ldots, (i_\ell, j_\ell)\}$ is a perfect matching of a graph G, then the product $\prod_{(i_k,j_k) \in M} x_{i_k j_k}$ appears as a surviving monomial in the sum on the right-hand side of Equation 1 when this sum is expanded and simplified over any field of characteristic two. Conversely, each surviving monomial in a simplified expansion of this sum over a field of characteristic two is of the form $\prod_{(i_k,j_k) \in M} x_{i_k j_k}$ where $M = \{(i_1, j_1), (i_2, j_2), \ldots, (i_\ell, j_\ell)\}$ is a perfect matching of G. In particular, $\det \mathcal{T}(G)$ is identically zero when expanded and simplified over a field of characteristic two if and only if graph G does not have a perfect matching.*

▶ **Lemma 4.** *(Schwartz-Zippel)*[19, 22] *Let $P(x_1, \ldots, x_n)$ be a multivariate polynomial of degree at most d over a field $\mathbb{F}$ such that P is not identically zero. Furthermore, let $r_1, \ldots, r_n$ be chosen uniformly at random from $\mathbb{F}$. Then, $Prob[P(r_1, \ldots, r_n) = 0] \leq \frac{d}{|\mathbb{F}|}$.*

We also require the following well-known interpolation lemma.

▶ **Lemma 5.** *Let $P(x)$ be a univariate polynomial of degree r over a field of size at least $r + 1$. Then, given $r + 1$ evaluations of $P(x)$, the polynomial can be determined in time polynomial in r.*

3 Editing to f-factors

The problem we study in this section is Edge Editing to f-factor. The formal definition of this problem is as follows.

Edge Editing to f-factor
Input: Graph $G = (V, E)$, function $f : V \to \mathbb{N}$, k.
Question: Can G be converted to a connected f-factor with at most k edge deletions and additions?

A set of k edge additions and deletions is referred to as a k-editing. For a given graph G, if $G - S_1 + S_2$ is an f-factor where S_1 is a set of edges of G and S_2 is a set of non-edges, then we refer to (S_1, S_2) as an ℓ-editing of G to an f-factor, where $\ell = |S_1 \cup S_2|$. It is easy to show that Editing to Connected f-factor where $f = 2$ is a generalization of Hamiltonicity (see for example [10]).

We begin with the following observation which relates local editing of subgraphs of G to f-factors on the one hand and the global editing of G to an f-factor on the other. We then subsequently define an auxiliary graph where perfect matchings capture editings to f-factors (see [17]).

▶ **Observation 1.** *Let G be a graph, let $f : V \to \mathbb{N}$, and let $S \subseteq V(G)$. Suppose the subgraphs $G[S]$ and $G - S$ have ℓ_1 and ℓ_2-editing to f-factors (S, F_1) and $((V(G) \setminus S), F_2)$ respectively and let ℓ_3 be the number of edges in the set $E(G)[S, V(G) \setminus S]$. Then, the union of the two editings along with the deletion of the edges in $E(G)[S, V(G) \setminus S]$ is an $(\ell_1 + \ell_2 + \ell_3)$-editing to the* disconnected *f-factor $(V(G), F_1 \uplus F_2)$. Similarly, let (S_1, S_2) be an editing of G to an f-factor $H = (V(G), F)$ and C be the union of some connected components of H. Let $S'_1 = S_1 \cap \binom{V(C)}{2}$ and $S'_2 = S_2 \cap \binom{V(C)}{2}$. Then, (S'_1, S'_2) is an editing to the f-factor C of the induced subgraph $G[V(C)]$.*

▶ **Definition 6** (Editing f-Blowup)**.** Let G be a graph and let $f : V(G) \to \mathbb{N}$ be such that $f(v) \le deg(v)$ for each $v \in V(G)$. Let H be a graph and w be a weight function on the edges of H defined as follows

1. For each vertex v of G, we add a vertex set $A(v)$ of size $f(v)$ to H.
2. For each edge $e = \{v, w\}$ of G we add to H vertices v_e and w_e and edges (u, v_e) for every $u \in A(v)$ and (w_e, u) for every $u \in A(w)$. We assign weight 0 to all these edges. Finally, we add the edge (v_e, w_e) and set $w(v_e, w_e) = 2$.
3. For each non-edge $\bar{e} = \{v, w\}$ of G we add to H vertices $v_{\bar{e}}$ and $w_{\bar{e}}$ and edges $(u, v_{\bar{e}})$ for every $u \in A(v)$ and $(w_{\bar{e}}, u)$ for every $u \in A(w)$. We assign weight 1 to each of these edges. Finally, we add the edge $(v_{\bar{e}}, w_{\bar{e}})$ and set $w(v_{\bar{e}}, w_{\bar{e}}) = 0$.

This completes the construction. The graph H along with the weight function $w : E(H) \to \{0, 1, 2\}$ is called the editing *f-blowup* of graph G. We use $\mathcal{E}_f(G)$ to denote the editing f-blowup of G. We omit the subscript when there is no scope for ambiguity.

▶ **Definition 7** (Induced Editing f-blowup)**.** For a subset $S \subseteq V(G)$, we define the editing f-blowup of G *induced* by S as follows. Let the editing f-blowup of G be (H, w). Begin with the graph H and for every edge $e = (v, w) \in E(G)$ such that $v \in S$ and $w \notin S$, delete the vertices v_e and w_e. Similarly, for every non-edge $\bar{e} = (v, w) \notin E(G)$ such that $v \in S$ and $w \notin S$, delete the vertices $v_{\bar{e}}$ and $w_{\bar{e}}$. Let the graph H' be the union of those connected components of the resulting graph which contain the vertex sets $A(v)$ for vertices $v \in S$. Then, the pair (H', w) is called the editing f-blowup of G *induced* by the set S and is denoted by $\mathcal{E}_f(G)[S]$.

The construction of the editing f-blowup of G can be informally described as taking the complete graph on $V(G)$, making $f(v)$ "equivalent copies" of every vertex $v \in V(G)$, replacing every edge and non-edge of G by a path of length 3, and assigning weight 2 to the "middle" edge of the paths corresponding to an edge of G, assigning weight 1 to the "end" edges of the path corresponding to a non-edge of G and weight 0 to all other edges. Similarly, the construction of the editing f-blowup of G induced by a subset $S \subseteq V(G)$ can be described analogously starting with the graph $G[S]$.

We now prove a lemma (see also [17]) which gives an equivalence between editings to f-factors and perfect matchings in the editing f-blowup.

▶ **Lemma 8.** *A graph G has an ℓ-editing to an f-factor with $\ell \leq k$ if and only if the editing f-blowup of G, (H, w), has a perfect matching of weight at most $2k$.*

Proof. Let (S_x, S_y) be an editing to an f-factor $(V(G), F)$ of G such that $|S_x \cup S_y| \leq k$, where $F = (E(G) \setminus S_x) \cup S_y$. We now define the following matching M in H. For every pair $(v, w) \in \binom{V}{2} \setminus F$, if $e = (v, w) \in E(G)$ then we add the edge (v_e, w_e) to M and if $\bar{e} = (v, w) \notin E(G)$ then we add the edge $(v_{\bar{e}}, w_{\bar{e}})$ to M. For every edge $(v, w) \in F$, if $e = (v, w) \in E(G)$ then we add the edges (u, v_e) and (u', w_e) to M where u and u' are two vertices in $A(v)$ and $A(w)$ respectively such that they are as yet unsaturated by M. Similarly, for every edge $(v, w) \in F$, if $\bar{e} = (v, w) \notin E(G)$ then we add the edges $(u, v_{\bar{e}})$ and $(u', w_{\bar{e}})$ to M where u and u' are two vertices in $A(v)$ and $A(w)$ respectively such that they are as yet unsaturated by M. Since $|A(v)| = f(v)$ for every $v \in V(G)$, M indeed saturates the sets $A(v)$ for every $v \in V(G)$ and therefore is a perfect matching. We now consider the weight of M. Clearly, $E(G) \setminus F = S_x$ and the weight contributed to M by the edges of H corresponding these edges is $2|S_x|$. Similarly, the weight contributed to M by the edges of H corresponding to those in $S_y = F \setminus E(G)$ is $2|S_y|$. Therefore, $w(M) \leq 2k$. This completes the proof of the forward direction.

Conversely, suppose that H has a perfect matching M of weight at most $2k$. Let $S_x = \{e = (v, w) | (v, w) \in E(G) \wedge (v_e, w_e) \in M\}$ and $S_y = \{\bar{e} = (v, w) | (v, w) \notin E(G) \wedge (v_{\bar{e}}, w_{\bar{e}}) \notin M\}$. Observe that for every $\bar{e} = (v, w) \in S_y$, there is a vertex $u \in A(v)$ and $u' \in A(w)$ such that $(u, v_{\bar{e}})$,$(u', w_{\bar{e}}) \in M$. This is because the vertex $v_{\bar{e}}$ $(w_{\bar{e}})$ has exactly one neighbor disjoint from $A(v)$ (respectively $A(w)$) and by assumption, $(v_{\bar{e}}, w_{\bar{e}}) \notin M$. Since each edge of the form $(u, v_{\bar{e}})$ (where $u \in A(v)$) has weight 1 and occurs in M along with an edge $(u', w_{\bar{e}})$ of weight 1 (with $\bar{e} = (v, w)$), we conclude that $2|S_x \cup S_y| = w(M) \leq 2k$. We now claim that $(V(G), F)$ is an f-factor, where $F = (E(G) \setminus S_x) \cup S_y$. Let M_A be all those edges of M incident on $\bigcup_{v \in V(G)} A(v)$. Starting from H, we define a subgraph H' of G as follows. For each $v \in V(G)$, we identify all vertices $A(v)$ in H. We then contract every edge in M_A. It is easy to see that the resulting graph is indeed an f-factor of G. Furthermore, by definition, the edges in M_A are precisely those corresponding to the edges in F. This completes the proof of the lemma. ◀

Having established the relation between perfect matchings in the f-blowup and editings to f-factors, we now recall the definition of a "weighted" Tutte matrix (see for example [11]) which allows us to handle edge weights as this will be crucially required to encode the size of the editings.

▶ **Definition 9.** (Weighted Tutte matrix) The *Weighted Tutte matrix* of a graph G with n vertices and a weight function $w : E(G) \to \mathbb{Z}$ is an $n \times n$ skew-symmetric matrix T over the set $\{x_{ij} | 1 \leq i < j \leq |V(G)|\} \cup \{z\}$ of indeterminates whose $(i, j)^{th}$ element is defined to be

$$T(i,j) = \begin{cases} x_{ij} z^{w(i,j)} & \text{if } (i,j) \in E(G) \text{ and } i < j \\ -x_{ji} z^{w(i,j)} & \text{if } (i,j) \in E(G) \text{ and } i > j \\ 0 & \text{otherwise} \end{cases}$$

We use $\mathcal{T}_z(G)$ to denote the Weighted Tutte matrix of graph G.

The following proposition is analogous to Proposition 1 and the proof is identical.

▶ **Proposition 2.** *If $M = \{(i_1, j_1), (i_2, j_2), \ldots, (i_\ell, j_\ell)\}$ is a perfect matching of a graph G with a weight function w on its edges, then the product $(\prod_{(i_k, j_k) \in M} x_{i_k j_k}) \cdot z^{\Sigma_{(i_k, j_k) \in M} w(i_k, j_k)}$ appears as a surviving monomial in the sum on the right-hand side of Equation 1 when* *applied to* $\mathcal{T}_z(G)$ *(instead of* $\mathcal{T}(G)$*) and the sum is expanded and simplified over any field of characteristic two. Conversely, each surviving monomial in a simplified expansion of this sum over a field of characteristic two is of the form* $(\prod_{(i_k,j_k)\in M} x_{i_k j_k}) \cdot z^{\sum_{(i_k,j_k)\in M} w(i_k,j_k)}$ *where* $M = \{(i_1,j_1),(i_2,j_2),\ldots,(i_\ell,j_\ell)\}$ *is a perfect matching of* G*. In particular,* $\det \mathcal{T}_z(G)$ *is identically zero when expanded and simplified over a field of characteristic two if and only if graph* G *does not have a perfect matching.*

▶ **Definition 10.** With every set $S \subseteq V(G)$, we associate a specific monomial m_S which is defined to be the product taken over all $e = (v,w) \in E(G)[S, V(G) \setminus S]$ of the terms $x_{ij} z^{w(i,j)}$ where $\{i,j\} = \{v_e, w_e\}$ and over all $\bar{e} = (v,w) \in E(\bar{G})[S, V(G) \setminus S]$ of the terms $x_{ij} z^{w(i,j)}$ where $\{i,j\} = \{v_{\bar{e}}, w_{\bar{e}}\}$, where the terms $v_e, w_e, v_{\bar{e}}, w_{\bar{e}}$ are as in Definition 6 of the editing f-blowup $\mathcal{E}(G)$ of G. If $S = V(G)$, then we set $m_S = 1$.

In the spirit of [6], we now fix an arbitrary vertex $v^\star$ of G and define a polynomial $P(\bar{x}, z)$ over the indeterminates from the weighted Tutte matrix $\mathcal{T}_z(\mathcal{E}(G))$ of the f-blowup of G, as follows:

$$P(\bar{x}, z) = \sum_{S \subseteq V(G)\,;\, v^\star \in S} (\det \mathcal{T}_z(\mathcal{E}(G)[S])) \cdot (\det \mathcal{T}_z(\mathcal{E}(G)[V(G) \setminus S])) \cdot m_S, \qquad (2)$$

where if a graph H has no vertices or edges then we set $\det \mathcal{T}(H) = 1$. In the sequel we use $\mathcal{F}$ to denote an arbitrary field of characteristic two. Observe that $P(\bar{x}, z)$ can be rewritten as $\sum_{i=0}^{r} Q_i(\bar{x}) \cdot z^i$ where r is an upper bound on the degree of z in any term of the polynomial $P(\bar{x}, z)$. We refer to the polynomial $Q_i(\bar{x})$ as the *coefficient* of z^i in $P(\bar{x}, z)$. Furthermore, every monomial m in the naïve expansion of $Q_i(\bar{x})$ is also referred to as a coefficient of z^i.

▶ **Definition 11.** We say that an editing (S_1, S_2) of G to an f-factor $(V(G), F)$ *contributes* a monomial $x_{i_1 j_1} \ldots x_{i_r j_r}$ to the naïve expansion of the right-hand side of Equation 2 if and only if the following conditions hold.

- For every $e = (v,w) \in F \cap E(G)$ ($\bar{e} = (v,w) \in F \setminus E(G)$), there is a $u \in A(v)$, $u' \in A(w)$ and $1 \le p, q \le r$ such that $\{u, v_e\} = \{i_p, j_p\}$ and $\{u', w_e\} = \{i_q, j_q\}$ (respectively $\{u, v_{\bar{e}}\} = \{i_p, j_p\}$ and $\{u', w_{\bar{e}}\} = \{i_q, j_q\}$).
- For every $e = (v,w) \in E(G) \setminus F$ ($\bar{e} = (v,w) \notin E(G) \cap F$), there is a $1 \le p \le r$ such that $\{v_e, w_e\} = \{i_p, j_p\}$ (respectively $\{v_{\bar{e}}, w_{\bar{e}}\} = \{i_p, j_p\}$).
- For every $1 \le p, q \le r$, if $\{u, v_e\} = \{i_p, j_p\}$ and $\{u', w_e\} = \{i_q, j_q\}$ for some $e \in F \cap E(G)$(respectively $\{u, v_{\bar{e}}\} = \{i_p, j_p\}$ and $\{u', w_{\bar{e}}\} = \{i_q, j_q\}$ for some $\bar{e} \in F \setminus E(G)$), then e (respectively $\bar{e}$) is in F.
- For every $1 \le p \le r$, if $\{i_p, j_p\} = \{v_e, w_e\}$ for some $e \in E(G)$ ($\{i_p, j_p\} = \{v_{\bar{e}}, w_{\bar{e}}\}$ for some $\bar{e} \notin E(G)$), then e (respectively $\bar{e}$) is not in F.
- For every $S \subseteq V(G)$ containing $v^\star$, such that S is a union of the vertex sets of (some) connected components of $(V(G), F)$, there is a pair of monomials m_1 and m_2 such that m_1 is a surviving monomial in the simplified expansion of $\det \mathcal{T}(\mathcal{E}(G)[S])$, m_2 is a surviving monomial in the simplified expansion of $\det \mathcal{T}(\mathcal{E}(G)[V(G) \setminus S])$, and $m_1 \cdot m_2 \cdot m_S = x_{i_1 j_1} \ldots x_{i_r j_r} \cdot z^{\sum_{k=1}^{r} w(i_k, j_k)}$.

Having set up the required notation, we now state the main lemma which allows us to show that monomials contributed by "undesirable editings" do not survive in the simplified expansion of the right hand side of Equation 2.

▶ **Lemma 12.** *Let* G *be a graph and* (S_1, S_2) *be an* ℓ*-editing of* G *to an* f*-factor* $(V(G), F)$*. Then,*

1. *All monomials contributed by (S_1, S_2) are coefficients of $z^{2\ell}$ in the naïve expansion of the right-hand side of Equation 2.*
2. *If $(V(G), F)$ is a* disconnected *f-factor of G then every monomial contributed by (S_1, S_2) occurs an* even *number of times in the polynomial $Q_{2\ell}(\bar{x})$ in the naïve expansion of the right-hand side of Equation 2.*
3. *If $(V(G), F)$ is a* connected *f-factor of G, then every monomial contributed by (S_1, S_2) occurs exactly once in the polynomial $Q_{2\ell}(\bar{x})$ in the naïve expansion of the right-hand side of Equation 2.*

As a consequence of the above lemma, we prove the following.

► **Lemma 13.** *The coefficient of $z^{2\ell}$ in the naïve expansion of $P(\bar{x}, z)$ is not identically zero over $\mathcal{F}$ if and only if G has an ℓ-editing to a connected f-factor.*

Proof. Observe that as a consequence of Proposition 2 combined with the proof of Lemma 8, we have that each surviving monomial in the naïve expansion of the right-hand side of Equation 2 is contributed by some editing to an f-factor of the graph G.

By this observation, every monomial which is a coefficient of $z^{2\ell}$ is contributed by an ℓ-editing to an f-factor and by Lemma 12, we have that every monomial contributed by this editing occurs an even number of times if and only if the resulting f-factor is disconnected. This completes the proof of the lemma. ◄

We now prove the main result of this section by giving an algorithm for editing to connected f-factors.

► **Theorem 14.** *There is a randomized algorithm that, given an instance (G, k) of* Editing to f-factor*, runs in time $\mathcal{O}^*(2^{|V(G)|})$ and either returns a solution or correctly (with high probability) concludes that one does not exist.*

Proof. Observe that the total degree of the polynomial $P(\bar{x}, z)$ is bounded by $n^2 + 2\binom{n}{2} + 2\binom{n}{2} \leq 3n^2$, where the sum of the first two terms is an upper bound on the number of vertices in the editing f-blowup which gives a bound on the degree of a monomial in $P(\bar{x}, z)$ due to $\bar{x}$ and the third term is a bound on the degree of a monomial due to z. We select values for the variables in $\bar{x}$ uniformly at random from a field $\mathcal{F}$ of characteristic 2 and size at least $3n^d$ for some fixed $d \geq 5$. Having fixed this instantiation of the variables in $\bar{x}$, we select $r = 2\binom{n}{2} + 1$ values for z from the field $\mathcal{F}$ and evaluate the polynomial $P(\bar{x}, z)$ for each of these r instantiations and return Yes if and only if for some $\ell \leq 2k$, the coefficient of z^ℓ is non-zero in the univariate polynomial $R(z)$ obtained by evaluating $P(\bar{x}, z)$ at the randomly selected points for $\bar{x}$. The r evaluations of the polynomial can be done in time $\mathcal{O}^*(2^n)$ by determinant computation and testing for a z^ℓ with non-zero coefficient in $R(z)$ can be done in polynomial time by interpolation (Lemma 5). This proves the stated bound on the running time of the algorithm. Therefore, it only remains to prove the correctness of the algorithm.

Suppose that (S_1, S_2) is a p-editing to a connected f-factor for some $p \leq k$. Then, by Lemma 13, we have that the coefficient of z^{2p}, $Q_{2p}(\bar{x})$, is not identically zero over $\mathcal{F}$. By the Schwartz-Zippel Lemma, we have that since $Q_{2p}(\bar{x})$ is not identically zero, then with probability at least $1 - \frac{1}{n^3}$ the evaluation of $Q_{2p}(\bar{x})$ at the randomly chosen points results in a non-zero value, implying that the coefficient of z^{2p} is non-zero in the polynomial $R(z)$. By the union bound, the probability that the coefficient of z^ℓ is "erroneously" zero in $R(z)$ for every $1 \leq \ell \leq 2k$ is at most $\frac{2k}{n^3} \leq \frac{1}{n}$. Therefore, if G has a p-editing to a connected f-factor with $p \leq k$, then with probability at least $1 - \frac{1}{n}$, we will detect the presence of such an editing. This completes the proof of the theorem. ◄

Finally, we note that if we are also given costs on the edges of the graph that are bounded polynomially in n, then we can also solve the version of the problem where costs are placed on the editing operations, in the same asymptotic running time with the only change appearing in the choice of the field from which $\bar{x}$ is instantiated at random. More precisely, we have the following theorem.

▶ **Theorem 1.** *There is a randomized algorithm that, given an instance* $(G, c, c^{\star})$ *of* Min-Cost Edge Editing to f-factor *with the cost function* c *being bounded by a polynomial in* $V(G)$*, runs in time* $\mathcal{O}^*(2^{|V(G)|})$ *and either returns a solution or correctly (with high probability) concludes that one does not exist.*

The problem of finding a connected f-factor in a given graph is special case of Min-Cost Edge Editing to f-factor and hence we have the following corollary.

▶ **Corollary 15.** *There is a randomized algorithm that, given an instance* (G, f) *of* Connected f-factor *where* $|V(G)| = n$*, runs in time* $\mathcal{O}^*(2^n)$ *and either returns a solution or correctly (with high probability) concludes that one does not exist.*

4 Parity f-factors

In this section we extend our approach to handle the parity version. Most of the proof is identical to the arguments in the previous section, and so we focus on defining the new kind of f-blowup which we need, and a description of the corresponding matching characterization.

▶ **Definition 16.** Given a graph G and a function $f : V(G) \to \{0, 1\}$, a *parity f-factor* of graph G is a spanning subgraph H of G in which every vertex v has degree exactly $f(v) \pmod 2$. A *connected parity f-factor* of G is such a *connected* subgraph H of G.

▶ **Definition 17** (Parity f-Blowup)**.** Let G be a graph and let $f : V(G) \to \{0, 1\}$. Let H be a graph defined as follows

1. For each vertex v of G, we add a vertex set $A(v)$ which has size $deg(v)$ if $deg(v) \equiv f(v) \pmod 2$ and size $deg(v) - 1$ otherwise.
2. For each edge $e = \{v, w\}$ of G we add vertices v_e and w_e and edges (u, v_e) for every $u \in A(v)$ and (w_e, u) for every $u \in A(w)$. Finally, we add the edge (v_e, w_e).
3. For each v such that $f(v) = 0$, we choose an arbitrary pair of vertices a_v and a'_v in $A(v)$ and make a clique on the rest of the vertices of $A(v)$. For each v such that $f(v) = 1$, we choose an arbitrary vertex a_v in $A(v)$ and make a clique on the rest of the vertices of $A(v)$.

This completes the construction. The graph H is called the *parity f-blowup* of graph G. We use $\mathcal{P}_f(G)$ to denote the parity f-blowup of G . We omit the subscript when there is no scope for ambiguity.

▶ **Definition 18** (Induced Parity f-blowup)**.** For a subset $S \subseteq V(G)$, we define the parity f-blowup of G *induced* by S as follows. Let the parity f-blowup of G be H. Begin with the graph H and for every edge $e = (v, w) \in E(G)$ such that $v \in S$ and $w \notin S$, delete the vertices v_e and w_e. Let the union of connected components of the resulting graph containing the vertices of the set S be the graph H'. Then, the graph H' is called the parity f-blowup of G *induced* by the set S and is denoted by $\mathcal{P}_f(G)[S]$.

▶ **Lemma 19.** *A graph G has a parity f-factor if and only if the parity f-blowup of G has a perfect matching.*

Proof. Suppose that G has a parity f-factor $(V(G), F)$. We now define a matching M in the parity f-blowup of G as follows. For every $e \in E(G) \setminus F$, we add the edge (v_e, w_e) to M. For every edge $(v, w) \in F$, we add the edges (u, v_e) and (u', w_e) to M where u and u' are two vertices in $A(v)$ and $A(w)$ respectively such that they are as yet unsaturated by M. However, if either of a_v or a'_v is unsaturated at this point, we chose to saturate one of these and similarly for a_w and a'_w.

Since $|A(v)| \equiv f(v) \pmod 2$ and $|A(v)| \geq deg(v)-1$ for every $v \in V(G)$, we conclude that M saturates $B(v)$ vertices from the set $A(v)$ for every $v \in V(G)$, where $B(v) \equiv f(v) \pmod 2$. Furthermore, since $(V(G), F)$ is a parity f-factor, $\{a_v, a'_v\} \subseteq B(v)$ for every v. The only unsaturated vertices in H at this point are the vertices in $A(v) \setminus B(v)$ for every $v \in V(G)$. However, since $B(v) \equiv f(v) \pmod 2$, we have that $B(v) \equiv |A(v)| \pmod 2$, implying that $|A(v) \setminus B(v)| \equiv 0 \pmod 2$. Since $\{a_v, a'_v\} \subseteq B(v)$ for every v, the subgraph $H[A(v) \setminus B(v)]$ is an even-sized clique and therefore we pick an arbitrary perfect matching in this clique and add it to M to get a perfect matching.

Conversely, suppose that M is a perfect matching of H. We define the set F as follows. For every $e = (v, w) \in E(G)$ such that $(v_e, w_e) \notin M$, we add the edge (v, w) to F. It can be argued along similar lines as before that $(V(G), F)$ is indeed a parity f-factor of G. This completes the proof of the lemma. ◀

Given the above definition of f-blowups and the structural lemma "equating" parity f-factors to perfect matchings in the f-blowup, the proof of the following theorem is identical to the proof of Theorem 14.

▶ **Theorem 2.** *There is a randomized algorithm that, given an instance (G, f) of* Connected Parity f-factor *where $|V(G)| = n$, runs in time $\mathcal{O}^*(2^n)$ and either returns a solution or correctly (with high probability) concludes that one does not exist.*

▶ **Corollary 20.** *There is a randomized algorithm that, given a graph G, $|V(G)| = n$, runs in time $\mathcal{O}^*(2^n)$ and either returns a connected Eulerian subgraph of G with the maximum (or minimum) number of edges, or correctly (with high probability) concludes that one does not exist.*

5 Conclusion

In this paper we studied certain generalizations of the well-studied NP-hard problems Hamiltonicity and Max/Min-Eulerian Subgraph. We gave $\mathcal{O}^*(2^n)$ time randomized algorithms for the problems of finding connected f-factors in a graph, minimum editing to obtain a connected f-factor and finding a connected parity f-factor. The most natural direction forward in this line of research would be towards obtaining a *deterministic* vertex exponential algorithm as well as algorithms that handle super-polynomial weights.

References

1 Richard Bellman. Dynamic programming treatment of the travelling salesman problem. *Journal of the Association of Computing Machinery*, 9(1):61–63, 1962.
2 Andreas Björklund. Determinant sums for undirected hamiltonicity. In *FOCS*, pages 173–182, 2010.
3 Leizhen Cai and Boting Yang. Parameterized complexity of even/odd subgraph problems. *J. Discrete Algorithms*, 9(3):231–240, 2011.
4 Gérard Cornuéjols. General factors of graphs. *J. Comb. Theory, Ser. B*, 45(2):185–198, 1988.

5 Marek Cygan, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Ildikó Schlotter. Parameterized complexity of eulerian deletion problems. *Algorithmica*, 68(1):41–61, 2014.
6 Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michal Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In *FOCS*, pages 150–159, 2011.
7 Reinhard Diestel. *Graph Theory.* Springer-Verlag, Heidelberg, 3rd edition, 2005.
8 Fedor V. Fomin and Petr A. Golovach. Parameterized complexity of connected even/odd subgraph problems. *J. Comput. Syst. Sci.*, 80(1):157–179, 2014.
9 Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., New York, NY, USA, 1979.
10 Petr A. Golovach. Editing to a connected graph of given degrees. *CoRR*, abs/1308.1802, 2013.
11 Gregory Gutin, Magnus Wahlström, and Anders Yeo. Parameterized rural postman and conjoining bipartite matching problems. *CoRR*, abs/1308.2599, 2013.
12 Michael Held and Richard M. Karp. A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied Mathematics*, 10(1):196–210, 1962.
13 Richard M. Karp. Reducibility Among Combinatorial Problems. In *Proceedings of a symposium on the Complexity of Computer Computations, held March 20–22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972.
14 László Lovász. The factorization of graphs. *Combinatorial Structures and Their Applications*, pages 243–246, 1970.
15 László Lovász. The factorization of graphs. ii. *Acta Mathematica Academiae Scientiarum Hungarica*, 23(1–2):223–246, 1972.
16 László Lovász. On determinants, matchings, and random algorithms. In L. Budach, editor, *Fundamentals of Computation Theory FCT'79*, pages 565–574, Berlin, 1979. Akademie-Verlag.
17 Luke Mathieson and Stefan Szeider. Editing graphs to satisfy degree constraints: A parameterized approach. *J. Comput. Syst. Sci.*, 78(1):179–191, 2012.
18 Hannes Moser and Dimitrios M. Thilikos. Parameterized complexity of finding regular induced subgraphs. *J. Discrete Algorithms*, pages 181–190, 2009.
19 J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, October 1980.
20 William Thomas Tutte. A short proof of the factor theorem for finite graphs. *Canadian Journal of Mathematics*, 6:347–352, 1954.
21 Magnus Wahlström. Abusing the tutte matrix: An algebraic instance compression for the k-set-cycle problem. In *STACS*, pages 341–352, 2013.
22 Richard Zippel. Probabilistic algorithms for sparse polynomials. In *Symbolic and Algebraic Computation*, volume 72, pages 216–226, 1979.