Balanced Substructures in Bicolored Graphs

P. S. Ardra[1(✉)], R. Krithika[1], Saket Saurabh[2,3], and Roohani Sharma[4]

[1] Indian Institute of Technology Palakkad, Palakkad, India
111914001@smail.iitpkd.ac.in, krithika@iitpkd.ac.in

[2] The Institute of Mathematical Sciences, HBNI, Chennai, India
saket@imsc.res.in

[3] University of Bergen, Bergen, Norway

[4] Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany
rsharma@mpi-inf.mpg.de

Abstract. An edge-colored graph is said to be *balanced* if it has an equal number of edges of each color. Given a graph G whose edges are colored using two colors and a positive integer k, the objective in the Edge Balanced Connected Subgraph problem is to determine if G has a balanced connected subgraph containing at least k edges. We first show that this problem is NP-complete and remains so even if the solution is required to be a tree or a path. Then, we focus on the parameterized complexity of Edge Balanced Connected Subgraph and its variants (where the balanced subgraph is required to be a path/tree) with respect to k as the parameter. Towards this, we show that if a graph has a balanced connected subgraph/tree/path of size at least k, then it has one of size at least k and at most $f(k)$ where f is a linear function. We use this result combined with dynamic programming algorithms based on *color coding* and *representative sets* to show that Edge Balanced Connected Subgraph and its variants are FPT. Further, using polynomial-time reductions to the Multilinear Monomial Detection problem, we give faster randomized FPT algorithms for the problems. In order to describe these reductions, we define a combinatorial object called *relaxed-subgraph*. We define this object in such a way that balanced connected subgraphs, trees and paths are relaxed-subgraphs with certain properties. This object is defined in the spirit of branching walks known for the Steiner Tree problem and may be of independent interest.

Keywords: Edge-colored graphs · Balanced subgraphs · Parameterized complexity

1 Introduction

Ramsey Theory is a branch of Combinatorics that deals with patterns in large arbitrary structures. In the context of edge-colored graphs where each edge is colored with one color from a finite set of colors, a fundamental problem in the area

L. Gąsieniec (Ed.): SOFSEM 2023, LNCS 13878, pp. 177–191, 2023.
https://doi.org/10.1007/978-3-031-23101-8_12

is concerned with the existence of *monochromatic* subgraphs of a specific type. Here, monochromatic means that all edges of the subgraph have the same color. For simplicity, we discuss only undirected graphs where each edge is colored either red or blue. Such a coloring is called a *red-blue coloring* and a graph associated with a red-blue coloring is referred to as a *red-blue graph*. In this work, we study questions related to the existence of and finding *balanced* subgraphs instead of monochromatic subgraphs, where by a balanced subgraph we mean one which has an equal number of edges of each color. These problems come under a subarea of Ramsey Theory known as Zero-Sum Ramsey Theory. Here, given a graph whose vertices/edges are assigned weights from a set of integers, one looks for conditions that guarantee the existence of a certain subgraph having total weight zero. For example, one may ask when is a graph whose all the edges are given weight -1 or 1 guaranteed to have a spanning tree with total weight of its edges 0. This is equivalent to asking when a red-blue graph is guaranteed to have a balanced spanning tree. Necessary and sufficient conditions have been established for complete graphs, triangle-free graphs and maximal planar graphs [7]. In the same spirit, one may ask a more general question like when is a red-blue graph G guaranteed to have a balanced connected subgraph of size (number of edges) k. An easy necessary condition is that there are at least $k/2$ red edges and at least $k/2$ blue edges in G. This condition is also sufficient (as we show in the proof of Theorem 3) if G is a complete graph (or more generally a split graph). However, we do not think that such a simple characterization will exist for all graphs. This brings us to the following natural algorithmic question concerning balanced connected subgraphs.

EDGE BALANCED CONNECTED SUBGRAPH **Parameter:** k
Input: A red-blue graph G and a positive integer k
Question: Does G have a balanced connected subgraph of size at least k?

When the subgraph is required to be a tree or a path, the corresponding variants of EDGE BALANCED CONNECTED SUBGRAPH are called EDGE BALANCED TREE and EDGE BALANCED PATH, respectively. We show that these problems are NP-complete.

- (Theorems 1, 2, 4) EDGE BALANCED CONNECTED SUBGRAPH, EDGE BALANCED TREE and EDGE BALANCED PATH are NP-complete.

In fact, EDGE BALANCED CONNECTED SUBGRAPH and EDGE BALANCED TREE remain NP-complete on bipartite graphs, planar graphs and chordal graphs. However, EDGE BALANCED CONNECTED SUBGRAPH is polynomial-time solvable on split graphs (Theorem 3). Yet, EDGE BALANCED PATH is NP-complete even on split graphs.

Note that if a graph has a balanced connected subgraph/tree/path of size at least k, then it is not guaranteed that it has one of size equal to k. This brings us to the following combinatorial question: if a graph has a balanced connected subgraph/tree/path of size at least k, then can we show that it has a balanced connected subgraph/tree/path of size equal to $f(k)$ for some function

f? We answer these questions in the affirmative and show the existence of such a function which is linear in k.

- (Theorems 5, 6, 7) If a graph has a balanced connected subgraph/tree of size at least k, then it has one of size at least k and at most $3k + 3$. Further, if a graph has a balanced path of size at least k, then it has a balanced path of size at least k and at most $2k$.

Therefore, in order to find a balanced connected subgraph/tree/path of size at least k, it suffices to focus on the problem of finding a balanced connected subgraph/tree/path of size exactly k. This leads us to the following problem.

EXACT EDGE BALANCED CONNECTED SUBGRAPH **Parameter:** k
Input: A red-blue graph G and a positive integer k
Question: Does G have a balanced connected subgraph of size k?

As before, when the connected subgraph is required to be a tree or a path, the corresponding variants of EXACT EDGE BALANCED CONNECTED SUBGRAPH are called EXACT EDGE BALANCED TREE and EXACT EDGE BALANCED PATH, respectively. These problems are also NP-complete and so we study them from the perspective of parameterized complexity. In this framework, each instance is associated with a non-negative integer ℓ called *parameter*, and a problem is said to be *fixed-parameter tractable* (FPT) with respect to ℓ if it can be solved in $\mathcal{O}^*(f(\ell))$[2] time for some computable function f. Algorithms with such running times are called FPT algorithms or parameterized algorithms. Focusing on solution size k as the parameter, we give randomized FPT algorithms for solving the three problems using reductions to the MULTILINEAR MONOMIAL DETECTION problem.

- (Theorems 8, 9, 10) EXACT EDGE BALANCED CONNECTED SUBGRAPH/TREE/PATH can be solved by a randomized algorithm in $\mathcal{O}^*(2^k)$ time.

Many problems reduce to MULTILINEAR MONOMIAL DETECTION [17] and the current fastest algorithm solving it is a randomized algorithm that runs in $\mathcal{O}^*(2^k)$ time [16,17,23]. The reductions that we give to MULTILINEAR MONOMIAL DETECTION use a combinatorial object called *relaxed-subgraph*. This object is defined in the spirit of *branching walks* known for the STEINER TREE problem [20]. We define this object in such a way that balanced connected subgraphs, trees and paths are relaxed-subgraphs with certain properties.

Then, using the *color-coding* technique [1,8] and *representative sets* [8,12,21], we give deterministic dynamic programming algorithms for the problems.

- (Theorems 11, 12, 13) EXACT EDGE BALANCED CONNECTED SUBGRAPH/TREE can be solved in $\mathcal{O}^*((4e)^k)$ time and EXACT EDGE BALANCED PATH can be solved in $\mathcal{O}^*(2.619^k)$ time.

The method of representative sets is a generic approach for designing efficient dynamic programming based parameterized algorithms that may be viewed as

[2] $\mathcal{O}^*$ notation supresses polynomial (in the input size) terms.

a deterministic-analogue to the color-coding technique. Representative sets have been used to obtain algorithms for several parameterized problems [12] and our algorithm adds to this list.

Road Map. The NP-completeness of the problems are given in Sect. 2. In Sect. 3, the combinatorial results related to the existence of small balanced connected subgraphs, trees and paths are proven. Section 4 discusses the deterministic and randomized algorithms for the problems. Section 5 concludes the work by listing some future directions.

Related Work. A variant of Exact Edge Balanced Connected Subgraph has recently been studied [2,3,9,15,18]. In order to state these results using our terminology, we define the notion of *vertex-balanced subgraphs* of *vertex-colored graphs*. A coloring of the vertices of a graph using red and blue colors is called a *red-blue vertex coloring*. A subgraph of a vertex-colored graph is said to be *vertex-balanced* if it has an equal number of vertices of each color. [2] and [3] study the Exact Vertex Balanced Connected Subgraph problem where the interest is in finding a vertex-balanced connected subgraph on k vertices in the given graph associated with a red-blue vertex coloring. This problem is NP-complete and remains so on restricted graph classes like bipartite graphs, planar graphs, chordal graphs, unit disk graphs, outer-string graphs, complete grid graphs, and unit square graphs [2,3]. However, polynomial-time algorithms are known for trees, interval graphs, split graphs, circular-arc graphs and permutation graphs [2,3]. Further, the problem is NP-complete even when the subgraph required is a path [2]. FPT algorithms, exact exponential algorithms and approximation results for the problem are known from [3], [15] and [18]. Observe that finding vertex-balanced connected subgraphs in vertex-colored graphs reduces to finding vertex-balanced trees while the analogous solution in edge-colored graphs may have more complex structures.

Preliminaries. For $k \in \mathbb{N}_+$, $[k]$ denotes the set $\{1, 2, ..., k\}$. In this work, we only consider simple undirected graphs. For standard graph-theoretic terminology not stated here, we refer the reader to the book by Diestel [10]. For the necessary parameterized complexity background, we refer to the book by Cygan et al. [8]. For a graph G, its sets of vertices and edges, are denoted by $V(G)$ and $E(G)$, respectively. The *size* of a graph is the number of its edges and the *order* of a graph is the number of its vertices. An edge between vertices u and v is denoted as $\{u, v\}$ and u and v are the *endpoints* of the edge $\{u, v\}$. Two vertices u, v in $V(G)$ are *adjacent* if there is an edge $\{u, v\}$ in G. The *neighborhood* of a vertex v, denoted by $N_G(v)$, is the set of vertices adjacent to v. Similarly, two edges e, e' in $E(G)$ are *adjacent* if they have exactly one common endpoint and the *neighborhood* of an edge e, denoted by $N_G(e)$, is the set of edges adjacent to e. The *degree* of a vertex v is the size of $N_G(v)$. A *tree* is an undirected connected acyclic graph. A *clique* is a set of pairwise adjacent vertices and a *complete graph* is a graph whose vertex set is a clique. A *split graph* is a graph whose vertex set can be partitioned into a clique and an independent set. Given a graph G, its *line graph* $L(G)$ is defined as $V(L(G)) = \{e \mid e \in E(G)\}$ and $E(L(G)) = \{\{e, e'\} \mid e \text{ and } e' \text{ are adjacent}\}$. It is well-known that a graph G without isolated vertices is connected if and only if $L(G)$ is connected.

Due to space constraints, for results labelled with a [$\star$], proofs are omitted or only a proof sketch is given.

2 NP-hardness Results

We show the NP-hardness of Edge Balanced Connected Subgraph using a polynomial-time reduction from the Steiner Tree problem [14, ND12]. In this problem, given a connected graph G, a subset $T \subseteq V(G)$ (called *terminals*) and a positive integer k, the task is to determine if G has a subtree H (called a *Steiner tree*) with $T \subseteq V(H)$ and $|E(T)| \leq k$. The idea behind the reduction is to color all edges of G of the Steiner Tree instance blue and add exactly k red edges incident to the terminals such that each terminal has at least one red edge incident on it. Any balanced connected subgraph of size (at least) k of the resulting graph is required to include all the red edges and hence includes all the terminals which in turn corresponds to a Steiner tree of G.

Theorem 1. [$\star$] Edge Balanced Connected Subgraph *is* NP*-complete.*

As the variant of the Steiner Tree problem where a tree on exactly k edges is required is also NP-complete, we have the following result.

Theorem 2. Edge Balanced Tree *is* NP*-complete.*

The reduction described in Theorem 1 is a polynomial parameter transformation and hence the infeasibility of the existence of polynomial kernels for Steiner Tree parameterized by the solution size (i.e., the size k of the tree) [8,11] extends to Edge Balanced Connected Subgraph and Edge Balanced Tree as well. Further, since Steiner Tree has no $2^{o(k)}$ time algorithm assuming the Exponential Time Hypothesis, it follows that Edge Balanced Connected Subgraph and Edge Balanced Tree also do not admit subexponential FPT algorithms. Moreover, the reduction in Theorem 1 preserves planarity, bipartiteness and chordality. This property along with the NP-completeness of Steiner Tree (and its variant) on bipartite graphs [14], planar graphs [13] and chordal graphs [22] imply that Edge Balanced Connected Subgraph and Edge Balanced Tree are NP-complete on planar graphs, chordal graphs and bipartite graphs as well.

2.1 Complexity in Split Graphs

Next, we consider Edge Balanced Connected Subgraph on split graphs. Let (G, k) be an instance. An easy necessary condition for G to have a balanced connected subgraph of size (at least) k is that there are at least $k/2$ red edges and at least $k/2$ blue edges in G. We show that this condition is also sufficient if G is a split graph leading to the following result.

Theorem 3. [⋆] EDGE BALANCED CONNECTED SUBGRAPH *is polynomial-time solvable on split graphs.*

Now, we move on to EDGE BALANCED PATH which we show is NP-hard on split graphs by giving a polynomial-time reduction from LONGEST PATH. In the LONGEST PATH problem, given a graph G and a positive integer k, the task is to find a path P in G of length k. It is known that LONGEST PATH is NP-hard [14, ND29] and remains so on split graphs even when the starting vertex u_0 of the path is given as part of the input [14, GT39]. The reduction may be viewed as attaching a red path of length k (consisting of new internal vertices) starting from u_0 to the split graph G (whose edges are colored blue) of the LONGEST PATH instance and adding certain additional edges (colored blue) to ensure that the graph remains a split graph.

Theorem 4. [⋆] EDGE BALANCED PATH *is* NP*-complete on split graphs.*

As LONGEST PATH parameterized by the solution size (i.e., the size k of the path) in general graphs does not admit a polynomial kernel [4,8] and the reduction described (which is adaptable for general graphs) is a polynomial parameter transformation, it follows that EDGE BALANCED PATH does not admit polynomial kernels. Further, it is known that, assuming the Exponential Time Hypothesis, LONGEST PATH has no $2^{o(k)}$ time algorithm in general graphs. Hence, EDGE BALANCED PATH also does not admit subexponential FPT algorithms.

3 Small Balanced Paths, Trees and Connected Subgraphs

In this section, we prove the combinatorial result that if a graph has a balanced connected subgraph/tree/path of size at least k, then it has one of size at least k and at most $f(k)$ where f is a linear function. We begin with balanced paths.

Theorem 5. [⋆] *Let G be a red-blue graph and $k \geq 2$ be a positive integer. Then, if G has a balanced path of length at least $2k$, then G has a smaller balanced path of length at least k.*

Proof. **(sketch)** Let E_B be the set of blue edges and E_R be the set of red edges in G. Consider a balanced path P in G with at least $2k$ edges. If the terminal edges e and e' are of different colors, then delete e and e' to get a smaller path of length at least k. Otherwise, let $P = (v_1, v_2, \ldots, v_\ell)$ where e_i denotes the edge $\{v_i, v_{i+1}\}$ for each $i \in [\ell-1]$. Without loss of generality, let $e_1, e_{\ell-1} \in E_R$. Define the function $h : E(P) \to \mathbb{N}$ as follows.

$$h(e_i) = \begin{cases} 1, & \text{if } i = 1 \\ h(e_{i-1}) + 1, & \text{if } i > 1 \text{ and } e_i \in E_R \\ h(e_{i-1}) - 1, & \text{if } i > 1 \text{ and } e_i \in E_B \end{cases}$$

We show that there is an edge e_i with $i < \ell - 1$ and $h(e_i) = 0$. Then, the subpaths P_1 and P_2 with $E(P_1) = \{e_1, \ldots, e_i\}$ and $E(P_2) = \{e_{i+1}, \ldots, e_{\ell-1}\}$ are two balanced paths strictly smaller than P. Further, as $|E(P)| \geq 2k$, at least one of them has at least k edges. □

Now, we move to the analogous result for balanced trees. An edge with an endpoint that has degree 1 is called a *pendant edge*.

Theorem 6. [$\star$] *Let G be a red-blue graph and $k \geq 2$ be a positive integer. Then, if G has a balanced tree with at least $3k+2$ edges, then G has a smaller balanced tree with at least k edges.*

Proof. **(sketch)** Let E_B be the set of blue edges and E_R be the set of red edges in G. Consider a balanced tree T in G with at least $3k+2$ edges. If T is a path, then by Theorem 5, we obtain the desired smaller tree (path). If T has pendant edges e and e' of different colors, then delete e and e' to get a smaller tree on at least k edges. Otherwise, without loss of generality, let all pendant edges of T be in E_R. Let n denote $|V(T)|$. Root T at an arbitrary vertex of degree at least 3. For a vertex $v \in V(T)$, let T_v denote the subtree of T rooted at v. Let u be a vertex with maximum distance from the root such that $|V(T_u)| > \frac{n}{3}$. Let $u_1, \ldots, u_\ell$ be the children of u. Observe that for each $i \in [\ell]$, $|V(T_{u_i})| \leq \frac{n}{3}$. Let i be the least integer in $[\ell]$ such that $\frac{n}{3} \leq |\bigcup_{1 \leq j \leq i} V(T_{u_j})| \leq \frac{2n}{3}$. Let S denote $\bigcup_{1 \leq j \leq i} V(T_{u_j})$ and R denote $V(T) \setminus S$. As $\frac{n}{3} \leq |S| \leq \frac{2n}{3}$, we have $\frac{n}{3} \leq |R| \leq \frac{2n}{3}$. Further, since $n \geq 3k+3$, we have $\frac{n}{3} \geq k+1$ and $\frac{2n}{3} \leq 2k+2$. Hence, $k+1 \leq |S|, |R| \leq 2k+2$. If $T[S \cup \{u\}]$ or $T[R]$ is balanced, then we get the desired result. Otherwise, consider the case when $T[S \cup \{u\}]$ and $T[R]$ have lesser edges from E_R than from E_B. As $E(T[S \cup \{u\}])$ and $E(T[R])$ partition $E(T)$, this case implies that T has more edges from E_B than from E_R contradicting that T is balanced. The remaining case is when $T[S \cup \{u\}]$ or $T[R]$ has more edges from E_R than from E_B. Suppose $T[R]$ has more edges from E_R than from E_B. Initialize T^* to be $T[R]$. As $T[R]$ has at least $k+1$ vertices (and therefore at least k edges), it has at least $k/2$ edges from E_R. Add the edges of $T[S \cup \{u\}]$ to T^* in the breadth-first order until T^* becomes balanced. We show that $T \neq T^*$ leading to the desired result. $\square$

Finally, we prove the result for balanced connected subgraphs using line graphs, vertex-balanced subgraphs and vertex-balanced trees. For a red-blue graph G, we define a red-blue coloring on $V(L(G))$ as follows: for each vertex x in $L(G)$ corresponding to a red (blue) edge $\{u, v\}$ in G, color x using red (blue). Then, G has a balanced connected subgraph of size ℓ if and only if $L(G)$ has a vertex-balanced connected subgraph of order ℓ. Now, it remains to show that if $L(G)$ has a vertex-balanced connected subgraph (equivalently, a vertex-balanced tree T) with at least $3k+3$ vertices, then $L(G)$ has a smaller vertex-balanced connected subgraph (equivalently, a vertex-balanced tree T^*) with at least k vertices. The proof of this claim is similar to the proof of Theorem 6.

Theorem 7. [$\star$] *Let G be a red-blue graph and $k \geq 2$ be a positive integer. Then, if G has a balanced connected subgraph with at least $3k+3$ edges, then G has a smaller balanced connected subgraph with at least k edges.*

Due to Theorems 5, 6 and 7, it suffices to give FPT algorithms for Exact Edge Balanced Connected Subgraph/Tree/Path in order to obtain FPT algorithms for Edge Balanced Connected Subgraph/Tree/Path.

4 FPT Algorithms

We now describe parameterized algorithms for Exact Edge Balanced Connected Subgraph/Tree/Path.

4.1 Randomized Algorithms

In this section, we show that Exact Edge Balanced Connected Subgraph, Exact Edge Balanced Path and Exact Edge Balanced Tree admit randomized algorithms that runs in $\mathcal{O}^*(2^k)$ time. We do so by reducing the problems to Multilinear Monomial Detection. In order to define this problem, we state some terminology related to polynomials from [17]. Let X denote a set of variables. A *monomial* of degree d is a product of d variables from X, with multiplication assumed to be commutative. A monomial is called *multilinear* if no variable appears twice or more in the product. A *polynomial* $P(X)$ over a ring is a linear combination of monomials with coefficients from the ring. A polynomial contains a certain monomial if the monomial appears with a non-zero coefficient in the linear combination that constitutes the polynomial. Polynomials can be represented as *arithmetic circuits* which in turn represented as *directed acyclic graphs*. In the Multilinear Monomial Detection problem, given an arithmetic circuit (represented as a directed acyclic graph) representing a polynomial $P(X)$ over $\mathbb{Z}_+$ and a positive integer k, the task is to decide whether $P(X)$ contains a multilinear monomial of degree at most k.

Proposition 1. [16,17,23] *Let $P(X)$ be a polynomial over $\mathbb{Z}_+$ represented by a circuit. The* Multilinear Monomial Detection *problem for $P(X)$ can be decided in randomized $\mathcal{O}^*(2^k)$ time and polynomial space.*

The reductions from Exact Edge Balanced Connected Subgraph/Tree/Path to Multilinear Monomial Detection crucially use the notions of a *color-preserving homomorphism* (also known as an edge-colored homomorphism in the literature [6]) and *relaxed-subgraphs*.

Definition 1. *Given graphs G and H with red-blue edge colorings $\mathtt{col}_G : E(G) \rightarrow \{red, blue\}$ and $\mathtt{col}_H : E(H) \rightarrow \{red, blue\}$, a color-preserving homomorphism from H to G is a function $h : V(H) \rightarrow V(G)$ satisfying the following properties: (1) For each pair $u, v \in V(H)$, if $\{u, v\} \in E(H)$, then $\{h(u), h(v)\} \in E(G)$. (2) For each edge $\{u, v\}$ in H, $\mathtt{col}_H(\{u, v\}) = \mathtt{col}_G(\{h(u), h(v)\})$.*

Definition 2. *Given a red-blue graph G, a relaxed-subgraph is a pair $S = (H, h)$ where H is a red-blue graph and h is a color-preserving homomorphism from H to G.*

The vertex set of a relaxed-subgraph $S = (H, h)$ is $V(S) = \{h(a) \in V(G) \mid a \in V(H)\}$ and the edge set of S is $\{\{h(a), h(b)\} \in E(G) \mid \{a, b\} \in E(H)\}$. We treat the vertex and edge sets of a relaxed-subgraph as multi-sets. The size of S is the number of edges in H (equivalently, the size of $E(S)$). S is said to be connected if H is connected and S is said to be balanced if H has an equal number of red edges and blue edges. S is said to be a *relaxed-path* if H is a path and a *relaxed-tree* if H is a tree. Next, we have the following observation that states that relaxed-subgraphs with certain specific properties correspond to balanced connected subgraphs, trees and paths.

Observation 1. [$\star$] *The following hold for a red-blue graph G.*

- *G has a balanced connected subgraph of size k if and only if there is a balanced connected relaxed-subgraph S of size k such that $E(S)$ consists of distinct elements.*
- *G has a balanced path of size k if and only if there is a balanced relaxed-path (P, h) of size k where h is injective.*
- *G has a balanced tree of size k if and only if there is a balanced relaxed-tree (T, h) of size k where h is injective.*

Now, we are ready to describe the randomized algorithms for Exact Edge Balanced Connected Subgraph/Tree/Path based on Observation 1. First, we consider Exact Edge Balanced Connected Subgraph.

Theorem 8. [$\star$] Exact Edge Balanced Connected Subgraph *admits a randomized $\mathcal{O}^*(2^k)$-time algorithm.*

Proof. **(sketch)** Consider an instance (G, k). Let E_R denote the set of red edges and E_B denote the set of blue edges in G. In order to obtain an instance of Multilinear Monomial Detection that is equivalent to (G, k), we will define a polynomial P over the variable set $\{x_e : e \in E(G)\}$ satisfying the following properties: (1) For each balanced connected relaxed-subgraph $S = (H, h)$ of size k there exists a monomial in P that corresponds to S. We say that a monomial M corresponds to S, if $M = \prod_{e \in E(S)} x_e$. (2) Each multilinear monomial in P corresponds to some balanced connected relaxed-subgraph S of size k where $E(S)$ has distinct elements.

If P is such a polynomial, then from Observation 1, G has a balanced connected subgraph of size k if and only if P has a multilinear monomial of degree k. This way, after the construction of P, we reduce the problem to Multilinear Monomial Detection and use Proposition 1. In order to construct P, we first construct polynomials $P_j(e, r, b)$ for each $e \in E(G)$, $j \in [k]$ and $0 \le r, b \le \frac{k}{2}$ with $r + b \ge 1$. Monomials of $P_j(e, r, b)$ will correspond to connected relaxed-subgraphs $S = (H, h)$ of size j such that H has r red edges, b blue edges and $e \in E(S)$. The construction of $P_j(e, r, b)$ is as follows. For an edge $e \in E(G)$,

$$P_1(e,1,0) = \begin{cases} x_e, & \text{if } e \in E_R \\ 0 & \text{otherwise.} \end{cases} \text{ and } P_1(e,0,1) = \begin{cases} x_e, & \text{if } e \in E_B \\ 0 & \text{otherwise.} \end{cases}.$$

Also, $P_j(e, r, b) = 0$ if $j \neq r + b$. Now, if $e \in E_R$ and $r + b > 1$, then we have

$$P_j(e,r,b) = \sum_{\substack{e' \in N_G(e), \ell < j \\ r'+r''=r, b'+b''=b}} P_\ell(e',r',b')P_{j-\ell}(e,r'',b'') + \sum_{e' \in N_G(e)} x_e P_{j-1}(e',r-1,b).$$

If $e \in E_B$ and $r+b > 1$, then $P_j(e,r,b) = \sum_{\substack{e' \in N_G(e), \ell < j \\ r'+r''=r, b'+b''=b}} P_\ell(e',r',b')P_{j-\ell}(e,r'',b'')$
$+ \sum_{e' \in N_G(e)} x_e P_{j-1}(e',r,b-1)$.

We now show that every multilinear monomial of $P_j(e,r,b)$ corresponds to a connected relaxed-subgraph $S=(H,h)$ of size j such that H has r red edges and b blue edges while $E(S)$ consists of distinct elements with $e \in E(S)$. We prove this claim by induction on j. The base case is easy to verify. Consider the induction step. Suppose $e=\{u,v\} \in E_R$ (the other case is symmetric). Let M be a multilinear monomial of $P_j(e,r,b)$ where $j>1$.

Case 1: $M = x_e M'$ where M' is a multilinear monomial of $P_{j-1}(e',r-1,b)$ such that $e' \in N_G(e)$. Let $e'=\{v,w\}$. By induction, M' corresponds to a connected relaxed-subgraph $S'=(H',h')$ of size $j-1$ such that $e' \in E(S')$ and H' has $r-1$ red edges, b blue edges with $v \in V(h'(H'))$. Let $z = h'^{-1}(v)$ (well-defined due to the multilinearity of M). Note that $E(S')$ consists of distinct elements and $e \notin E(S')$. Let H denote the graph obtained from H' by adding a new vertex z' adjacent to z with the edge $\{z,z'\}$ colored red. Let $h: V(H) \to V(G)$ denote the homomorphism obtained from h' by extending its domain to include z' and setting $h(z')=u$. Then, $S=(H,h)$ is a connected relaxed-subgraph that M corresponds to.

Case 2: $M = M_1 M_2$ where M_1 is a multilinear monomial of $P_{j_1}(e',r',b')$ and M_2 is a multilinear monomial of $P_{j_2}(e,r'',b'')$ such that $e' \in N_G(e)$, $j_1, j_2 < j$, $r'' \leq r$ and $b'' \leq b$. Let $e'=\{v,w\}$. By induction, M_1 corresponds to a connected relaxed-subgraph $S_1=(H_1,h_1)$ of size j_1 such that $e' \in E(S_1)$. Similarly, M_2 corresponds to a connected relaxed-subgraph $S_2=(H_2,h_2)$ of size j_2 such that $e \in E(S_2)$. Further, H_1 has r' red edges, b' blue edges and H_2 has r'' red edges and b'' blue edges. Also, $v \in V(h_1(H_1)) \cap V(h_2(H_2))$ and $E(S_1) \cap E(S_2) = \emptyset$. Without loss of generality, assume that $V(H_1) \cap V(H_2) = \emptyset$ as this can be achieved by a renaming procedure. Let $z_1 = h_1^{-1}(v)$ and $z_2 = h_2^{-1}(v)$. Observe that z_1 and z_2 are well-defined due to the multilinearity of M_1 and M_2. Now, rename z_1 in S_1 and z_2 in S_2 as z. Let H denote the graph with vertex set $V(H_1) \cup V(H_2)$ and edge set $E(H_1) \cup E(H_2)$. Observe that H is a connected graph. Let $h: V(H) \to V(G)$ denote identity map. Then, $S=(H,h)$ is a connected relaxed-subgraph that M corresponds to.

Similarly, we show (by induction on j) that if there is a connected relaxed-subgraph $S=(H,h)$ of size j with r red edges, b blue edges and such that $e=\{u,v\} \in E(S)$, then there is a monomial of $P_j(e,r,b)$ that corresponds to it. Suppose $e \in E_R$ (the other case is symmetric). The base case is trivial. Consider the induction step ($j \geq 2$). Let $a = h^{-1}(u)$, $b = h^{-1}(v)$ and z denote the edge $\{a,b\}$ of H.

Case 1: $H-z$ is connected. Then, $S'=(H-z,h)$ is a connected relaxed-subgraph of size $j-1$ with $r-1$ red edges and b blue edges and contains an edge $e' \in N_G(e)$. By induction, there is a monomial M' corresponding to S' in $P_{j-1}(e',r-1,b)$. Then, the monomial $M = x_e M'$ which is in $P_j(e,r,b)$ corresponds to S.

Case 2: $H-z$ is disconnected. Then H has two components H_a (containing a) and H_b (containing b). Without loss of generality let H_a have at least one edge. Let H_b' denote the subgraph of H obtained from H_b by adding the vertex a and edge $\{a,b\}$. Let j_1 and j_2 denote the number of edges in H_a and H_b', respectively. Let r' and r'' be the number of red edges in H_a and H_b', respectively. Similarly, let b' and b'' be the number of blue edges in H_a and H_b', respectively. Then $j = j_1 + j_2, r = r' + r'', b = b' + b''$. Let h_a and h_b denote the color-preserving homomorphism obtained from h by restricting the domain to $V(H_a)$ and $V(H_b)$, respectively. Now, considering the connected relaxed-subgraphs $S_1 = (H_a, h_a)$ and $S_2 = (H_b, h_b)$, by induction there is a monomial M_1 in $P_{j_1}(e', r', b')$ that corresponds to S_1 and there is a monomial M_2 in $P_{j_2}(e, r'', b'')$ that corresponds to S_2. Then, the monomial M_1M_2 which is in $P_j(e,r,b)$ corresponds to S. Finally, let $P = \sum_{e\in E(G)} P_k(e, \frac{k}{2}, \frac{k}{2})$. Every monomial in P has degree k. Then from the arguments above, P is the desired polynomial. As these polynomials can be represented as a polynomial-sized arithmetic circuit, the reduction runs in polynomial time. $\square$

Next, we move on to Exact Edge Balanced Tree. We define a polynomial over the variable set corresponding to vertices (as opposed to variable set corresponding to edges as in Exact Edge Balanced Connected Subgraph) leading to the following result.

Theorem 9. [$\star$] Exact Edge Balanced Tree *admits a randomized* $\mathcal{O}^*(2^k)$-*time algorithm.*

Proof. **(sketch)** We define a polynomial $P = \sum_{e\in E(G)} P_k(e, \frac{k}{2}, \frac{k}{2})$ over the variable set $\{y_v : v \in V(G)\}$ with subtle differences from the one defined for balanced connected subgraphs where $P_j(e,r,b)$ is defined as follows with $e = \{u,v\}$.

$$P_1(e,1,0) = \begin{cases} y_u y_v, & \text{if } e \in E_R \\ 0 & \text{otherwise.} \end{cases} \text{ and } P_1(e,0,1) = \begin{cases} y_u y_v, & \text{if } e \in E_B \\ 0 & \text{otherwise.} \end{cases}.$$

For $j > 1$, if $e \in E_R$ (the polynomial for $e \in E_B$ is similar),

$$\begin{aligned} P_j(e,r,b) = &\sum_{\substack{e',e''\in N_G(e)\\ u\in V(e'),v\in V(e'')\\ r'<r,b'\le b,\ell<j}} P_\ell(e',r',b')P_{j-1-\ell}(e'',r-1-r',b-b') \\ &+ \sum_{\substack{e'\in N_G(e)\\ v\in V(e')}} y_u P_{j-1}(e',r-1,b) + \sum_{\substack{e'\in N_G(e)\\ u\in V(e')}} y_v P_{j-1}(e',r-1,b). \end{aligned}$$

The properties of $P = \sum_{e\in E(G)} P_k(e, \frac{k}{2}, \frac{k}{2})$ then lead to the claimed result. $\square$

Finally, we define a (simpler) polynomial for Exact Edge Balanced Tree satisfying certain properties leading to the following result.

Theorem 10. [$\star$] Exact Edge Balanced Path *admits a randomized* $\mathcal{O}^*(2^k)$-*time algorithm.*

4.2 Deterministic Algorithms

We first describe deterministic algorithms for Exact Edge Balanced Connected Subgraph and Exact Edge Balanced Tree using the color-coding technique [1,8,19]. Consider an instance (G, k) of Exact Edge Balanced Connected Subgraph/Tree. Let E_R denote the set of red edges and E_B denote the set of blue edges in G. Let $\sigma : E(G) \to [k]$ be a coloring of edges of G and $\tau : V(G) \to [k+1]$ be a coloring of vertices of G. For $L \subseteq [k+1]$, a subgraph $H \subseteq G$ is said to be L-edge-colorful if $|E(H)| = |L|$ and $\bigcup_{e \in E(H)} \sigma(e) = L$. Similarly, H is said to be L-vertex-colorful if $|V(H)| = |L|$ and $\bigcup_{v \in V(H)} \tau(v) = L$. We describe dynamic programming algorithms to find a $[k]$-edge-colorful balanced connected subgraph and a $[k+1]$-vertex-colorful balanced tree in G (if they exist) in $\mathcal{O}^*(4^k)$ time. Then, a standard derandomization technique using *perfect hash families* [1,8,19] leads to the following results.

Theorem 11. [$\star$] Exact Edge Balanced Connected Subgraph *can be solved in* $\mathcal{O}^*((4e)^k)$ *time.*

Theorem 12. [$\star$] Exact Edge Balanced Tree *can be solved in* $\mathcal{O}^*((4e)^k)$ *time.*

Analogous to Theorems 11 and 12, we can show that Exact Edge Balanced Path can be solved in $\mathcal{O}^*((2e)^k)$ time. Subsequently, we describe a faster algorithm using representative sets. We begin with some definitions and results related to representative sets. For a finite set U, let $\binom{U}{p}$ denote the set of all subsets of size p of U. Given two families $\mathcal{S}_1, \mathcal{S}_2 \subseteq 2^U$, the *convolution* of $\mathcal{S}_1$ and $\mathcal{S}_2$ is the new family defined as $\mathcal{S}_1 * \mathcal{S}_2 = \{X \cup Y \mid X \in \mathcal{S}_1, Y \in \mathcal{S}_2, X \cap Y = \emptyset\}$.

Definition 3. *Let U be a set and $\mathcal{S} \subseteq \binom{U}{p}$. A subfamily $\widehat{\mathcal{S}} \subseteq \mathcal{S}$ is said to q-represent $\mathcal{S}$ (denoted as $\widehat{\mathcal{S}} \subseteq^q_{rep} \mathcal{S}$) if for every set $Y \subseteq U$ of size at most q such that there is a set $X \in \mathcal{S}$ with $X \cap Y = \emptyset$, there is a set $\widehat{X} \in \widehat{\mathcal{S}}$ with $\widehat{X} \cap Y = \emptyset$. If $\widehat{\mathcal{S}} \subseteq^q_{rep} \mathcal{S}$, then $\widehat{\mathcal{S}}$ is called a* q-representative family *for $\mathcal{S}$.*

Representative families (also called representative sets) are transitive and have nice union and convolution properties [8, Lemmas 12.26, 12.27 and 12.28]. A classical result due to Bollobás states that small representative families exist [5] and a result due to [12] and [21] (see also [8]) shows that such families can be efficiently computed.

Theorem 13. [$\star$] Exact Edge Balanced Path *can be solved in* $\mathcal{O}^*(2.619^k)$ *time.*

Proof. **(sketch)** Consider an instance (G, k). Let E_R and E_B denote the sets of red and blue edges of G. For a pair of vertices $u, v \in V(G)$ and non-negative integers r and b with $r + b \geq 1$, define the family $\mathcal{P}^{(r,b)}_{uv}$ as follows.

$$\mathcal{P}^{(r,b)}_{uv} = \{X : X \subseteq V(G),\ |X| = r + b + 1 \text{ and there is a } uv\text{-path } P \text{ with } V(P) = X,\ |E_R \cap E(P)| = r \text{ and } |E_B \cap E(P)| = b\}.$$

Now, it suffices to determine if $\mathcal{P}_{uv}^{(\frac{k}{2},\frac{k}{2})}$ is non-empty for some $u, v \in V(G)$. The families $\mathcal{P}_{uv}^{(r,b)}$ can be computed using the following formula. For $r+b=1$,

$$\mathcal{P}_{uv}^{(1,0)} = \begin{cases} \{\{u,v\}\}, & \text{if } \{u,v\} \in E_R \\ \emptyset, & \text{otherwise} \end{cases} \text{ and } \mathcal{P}_{uv}^{(0,1)} = \begin{cases} \{\{u,v\}\}, & \text{if } \{u,v\} \in E_B \\ \emptyset, & \text{otherwise} \end{cases}$$

For $r+b>1$, $\mathcal{P}_{uv}^{(r,b)} = (\bigcup_{\{w,v\}\in E_R} (\mathcal{P}_{uw}^{(r-1,b)} * \{\{v\}\})) \bigcup (\bigcup_{\{w,v\}\in E_B} (\mathcal{P}_{uw}^{(r,b-1)} * \{\{v\}\}))$.

Clearly, a naive computation of $\mathcal{P}_{uv}^{(r,b)}$ is not guaranteed to result in an FPT (in k) algorithm. Therefore, instead of computing $\mathcal{P}_{uv}^{(r,b)}$, we compute $\widehat{\mathcal{P}}_{uv}^{(r,b)} \subseteq_{rep}^{k-(r+b)} \mathcal{P}_{uv}^{(r,b)}$ and use the fact that $\widehat{\mathcal{P}}_{uv}^{(\frac{k}{2},\frac{k}{2})} \subseteq_{rep}^{0} \mathcal{P}_{uv}^{(\frac{k}{2},\frac{k}{2})}$. Now, we describe a dynamic programming algorithm to compute $\widehat{\mathcal{P}}_{uv}^{(\frac{k}{2},\frac{k}{2})}$ for every $u, v \in V(G)$. For $r+b=1$, set $\widehat{\mathcal{P}}_{uv}^{(1,0)} = \mathcal{P}_{uv}^{(1,0)}$ and $\widehat{\mathcal{P}}_{uv}^{(0,1)} = \mathcal{P}_{uv}^{(0,1)}$. Clearly, $\widehat{\mathcal{P}}_{uv}^{(1,0)} \subseteq_{rep}^{k-1} \mathcal{P}_{uv}^{(1,0)}$ and $\widehat{\mathcal{P}}_{uv}^{(0,1)} \subseteq_{rep}^{k-1} \mathcal{P}_{uv}^{(0,1)}$. Further, $|\widehat{\mathcal{P}}_{uv}^{(1,0)}|, |\widehat{\mathcal{P}}_{uv}^{(0,1)}| \leq 1$ and this computation is polynomial time. Now, we proceed to computing $\widehat{\mathcal{P}}_{uv}^{(r,b)} \subseteq_{rep}^{k-(r+b)} \mathcal{P}_{uv}^{(r,b)}$ for $k \geq r+b > 1$ in the increasing order of $r+b$. Towards this, we compute a new family $\widetilde{\mathcal{P}}_{uv}^{(r,b)}$ as follows.

$$\widetilde{\mathcal{P}}_{uv}^{(r,b)} = (\bigcup_{\{w,v\}\in E_R} (\widehat{\mathcal{P}}_{uw}^{(r-1,b)} * \{\{v\}\})) \bigcup (\bigcup_{\{w,v\}\in E_B} \widehat{\mathcal{P}}_{uw}^{(r,b-1)} * \{\{v\}\}))$$

Using the union and convolution properties of representative sets, $\widetilde{\mathcal{P}}_{uv}^{(r,b)} \subseteq_{rep}^{k-(r+b)} \mathcal{P}_{uv}^{(r,b)}$. Further, $|\widetilde{\mathcal{P}}_{uv}^{(r,b)}| = \mathcal{O}^*(|\widehat{\mathcal{P}}_{uv}^{(r-1,b)}| + |\widehat{\mathcal{P}}_{uv}^{(r,b-1)}|)$. Then, we use the result of [12] and [21] to compute a family $\widehat{\mathcal{P}}_{uv}^{(r,b)} \subseteq_{rep}^{k-(r+b)} \widetilde{\mathcal{P}}_{uv}^{(r,b)}$. By the transitivity property of representative sets, it follows that $\widehat{\mathcal{P}}_{uv}^{(r,b)} \subseteq_{rep}^{k-(r+b)} \mathcal{P}_{uv}^{(r,b)}$. Further, from the running time analysis given in [12] and [21], the overall running time of the algorithm is $\mathcal{O}^*(2.619^k)$. $\square$

5 Concluding Remarks

To summarize our work, we studied the complexity of finding balanced connected subgraphs, trees and paths in red-blue graphs. We gave fixed-parameter tractability results using color-coding, representative sets and reductions to Multilinear Monomial Detection. En route, we showed combinatorial results on the existence of small balanced connected subgraphs, trees and paths. We observe that these results also extend to vertex-balanced connected subgraphs, trees and paths. As a result the algorithms described in this work also generalize to solve the vertex-analogue of the problems. Note that using line graphs, one can reduce Edge Balanced Connected Subgraph to Vertex Balanced Connected Subgraph, however, when the solution is required to be a path or a tree, this reduction is not useful. An interesting next direction of research is determining the complexity of finding other balanced substructures. Also, studying the problems on graphs that are colored using more than two colors and on colored weighted graphs are natural questions in this context.

References

1. Alon, N., Yuster, R., Zwick, U.: Color-coding. J. ACM **42**(4), 844–856 (1995). https://doi.org/10.1145/210332.210337. https://doi.org/10.1145/210332.210337
2. Bhore, S., Chakraborty, S., Jana, S., Mitchell, J.S.B., Pandit, S., Roy, S.: The balanced connected subgraph problem. In: Pal, S.P., Vijayakumar, A. (eds.) CALDAM 2019. LNCS, vol. 11394, pp. 201–215. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11509-8_17
3. Bhore, S., Jana, S., Pandit, S., Roy, S.: Balanced connected subgraph problem in geometric intersection graphs. In: Li, Y., Cardei, M., Huang, Y. (eds.) COCOA 2019. LNCS, vol. 11949, pp. 56–68. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-36412-0_5
4. Bodlaender, H.L., Downey, R.G., Fellows, M.R., Hermelin, D.: On problems without polynomial kernels. J. Comput. Syst. Sci. **75**(8), 423–434 (2009). https://doi.org/10.1016/j.jcss.2009.04.001. https://doi.org/10.1016/j.jcss.2009.04.001
5. Bollobás, B.: On generalized graphs. Acta Math. Hungar. **16**(3–4), 447–452 (1965)
6. Brewster, R.C., Dedic, R., Huard, F., Queen, J.: The recognition of bound quivers using edge-coloured homomorphisms. Discret. Math. **297**(1-3), 13–25 (2005). https://doi.org/10.1016/j.disc.2004.10.026. https://doi.org/10.1016/j.disc.2004.10.026
7. Caro, Y., Hansberg, A., Lauri, J., Zarb, C.: On zero-sum spanning trees and zero-sum connectivity. Electron. J. Comb. **29**(1), P1.9 (2022). https://doi.org/10.37236/10289. https://doi.org/10.37236/10289
8. Cygan, M., et al.: Parameterized Algorithms. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21275-3
9. Darties, B., Giroudeau, R., Jean-Claude, K., Pollet, V.: The balanced connected subgraph problem: complexity results in bounded-degree and bounded-diameter graphs. In: Li, Y., Cardei, M., Huang, Y. (eds.) COCOA 2019. LNCS, vol. 11949, pp. 449–460. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-36412-0_36
10. Diestel, R.: Graph Theory. GTM, vol. 173. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-53622-3
11. Dom, M., Lokshtanov, D., Saurabh, S.: Kernelization lower bounds through colors and ids. ACM Trans. Algorithms **11**(2), 1–20 (2014). https://doi.org/10.1145/2650261. https://doi.org/10.1145/2650261
12. Fomin, F.V., Lokshtanov, D., Panolan, F., Saurabh, S.: Efficient computation of representative families with applications in parameterized and exact algorithms. J. ACM **63**(4), 1–60 (2016). https://doi.org/10.1145/2886094. https://doi.org/10.1145/2886094
13. Garey, M.R., Johnson, D.S.: The rectilinear steiner tree problem in NP complete. J. SIAM Appl. Math. **32**, 826–834 (1977)
14. Garey, M.R., Johnson, D.S.: Computers and Intractability: a Guide to the Theory of NP-Completeness. W. H. Freeman (1979)
15. Kobayashi, Y., Kojima, K., Matsubara, N., Sone, T., Yamamoto, A.: Algorithms and hardness results for the maximum balanced connected subgraph problem. In: Li, Y., Cardei, M., Huang, Y. (eds.) COCOA 2019. LNCS, vol. 11949, pp. 303–315. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-36412-0_24
16. Koutis, I.: Faster algebraic algorithms for path and packing problems. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008. LNCS, vol. 5125, pp. 575–586. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-70575-8_47

17. Koutis, I., Williams, R.: LIMITS and applications of group algebras for parameterized problems. ACM Trans. Algorithms **12**(3), 1–18 (2016). https://doi.org/10.1145/2885499. https://doi.org/10.1145/2885499
18. Martinod, T., Pollet, V., Darties, B., Giroudeau, R., König, J.: Complexity and inapproximability results for balanced connected subgraph problem. Theor. Comput. Sci. **886**, 69–83 (2021). https://doi.org/10.1016/j.tcs.2021.07.010. https://doi.org/10.1016/j.tcs.2021.07.010
19. Naor, M., Schulman, L.J., Srinivasan, A.: Splitters and near-optimal derandomization. In: 36th Annual Symposium on Foundations of Computer Science, Milwaukee, Wisconsin, USA, 23–25 October 1995, pp. 182–191 IEEE Computer Society (1995). https://doi.org/10.1109/SFCS.1995.492475. https://doi.org/10.1109/SFCS.1995.492475
20. Nederlof, J.: Fast polynomial-space algorithms using inclusion-exclusion. Algorithmica **65**(4), 868–884 (2013). https://doi.org/10.1007/s00453-012-9630-x. https://doi.org/10.1007/s00453-012-9630-x
21. Shachnai, H., Zehavi, M.: Representative families: a unified tradeoff-based approach. J. Comput. Syst. Sci. **82**(3), 488–502 (2016). https://doi.org/10.1016/j.jcss.2015.11.008. https://doi.org/10.1016/j.jcss.2015.11.008
22. White, K., Farber, M., Pulleyblank, W.R.: Steiner trees, connected domination and strongly chordal graphs. Networks **15**(1), 109–124 (1985). https://doi.org/10.1002/net.3230150109. https://doi.org/10.1002/net.3230150109
23. Williams, R.: Finding paths of length k in $O^*(2^k)$ time. Inf. Process. Lett. **109**(6), 315–318 (2009). https://doi.org/10.1016/j.ipl.2008.11.004. https://doi.org/10.1016/j.ipl.2008.11.004