ARTICLE IN PRESS

Discrete Applied Mathematics xxx (xxxx) xxx

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

The balanced connected subgraph problem ☆

Sujoy Bhore [a], Sourav Chakraborty [b], Satyabrata Jana [b], Joseph S.B. Mitchell [c,1], Supantha Pandit [d,*,2], Sasanka Roy [b]

[a] Algorithms and Complexity Group, TU Wien, Vienna, Austria
[b] Indian Statistical Institute, Kolkata, India
[c] Stony Brook University, Stony Brook, NY, USA
[d] Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar, Gujarat, India

ARTICLE INFO

Article history:
Received 9 August 2019
Received in revised form 10 July 2020
Accepted 26 December 2020
Available online xxxx

Keywords:
Balanced connected subgraph
Trees
Split graphs
Chordal graphs
Planar graphs
Bipartite graphs
NP-hard
Polynomial algorithms

ABSTRACT

The problem of computing induced subgraphs that satisfy some specified restrictions arises in various applications of graph algorithms and has been well studied. In this paper, we consider the following *Balanced Connected Subgraph* (*BCS*) problem. The input is a graph $G = (V, E)$, with each vertex in the set V having an assigned color, "red" or "blue". We seek a maximum-cardinality subset $V' \subseteq V$ of vertices that is *color-balanced* (having exactly $|V'|/2$ red vertices and $|V'|/2$ blue vertices), such that the subgraph induced by the vertex set V' in G is connected. We show that the BCS problem is NP-hard, even for bipartite graphs G (with red/blue color assignment not necessarily being a proper 2-coloring). Further, we consider this problem on various graph classes, e.g., planar graphs, chordal graphs, trees, split graphs, bipartite graphs with a proper red/blue 2-coloring, and graphs with diameter 2. For each of these classes we either prove NP-hardness or design a polynomial time algorithm.

1. Introduction

Given a graph $G = (V, E)$, and a subset of vertices $S \subset V$, the induced subgraph $G[S]$ is the graph whose vertex set is S and whose edge set consists of all of the edges in E that have both endpoints in S. A plethora of problems in graph theory and combinatorial optimization involve determining if a given graph G has an induced subgraph with certain properties. Some of the related optimization problems include finding cliques, independent sets, connected dominating sets, connected vertex cover, induced paths, cycles, matchings, etc. These problems are extensively studied over the years due to their significant theoretical interests as well as many practical applications; see [15,16,23,27,35].

We study the problem in which we are given a simple connected graph $G = (V, E)$ where each vertex in V is colored with either "red" or "blue" (note, the color assignment might not be a proper 2-coloring of the vertices, i.e., we allow

☆ A preliminary version of this paper appeared in the 5th Annual International Conference on Algorithms and Discrete Applied Mathematics (CALDAM) 2019 (Bhore et al., 2019 [8]).

* Corresponding author.
E-mail addresses: sujoy.bhore@gmail.com (S. Bhore), chakraborty.sourav@gmail.com (S. Chakraborty), satyamtma@gmail.com (S. Jana), joseph.mitchell@stonybrook.edu (J.S.B. Mitchell), pantha.pandit@gmail.com (S. Pandit), sasanka.ro@gmail.com (S. Roy).

1 Support from the National Science Foundation, USA (CCF-1526406) and the US-Israel Binational Science Foundation (project 2016116).

2 This work was done while the author was at the Stony Brook University, Stony Brook, NY, USA and was partially supported by the Indo-US Science & Technology Forum (IUSSTF) under the SERB Indo-US Postdoctoral Fellowship scheme with grant number 2017/94, Department of Science and Technology, Government of India.

https://doi.org/10.1016/j.dam.2020.12.030
0166-218X/

Please cite this article as: S. Bhore, S. Chakraborty, S. Jana et al., The balanced connected subgraph problem, Discrete Applied Mathematics (2021), https://doi.org/10.1016/j.dam.2020.12.030.

vertices of the same color to be adjacent in G). We seek a maximum-cardinality subset $V' \subseteq V$ of the vertices such that V' is *color-balanced*, i.e. having same number of red and blue vertices in V', and such that the induced subgraph H by V' in G is connected. We refer to this as the *Balanced Connected Subgraph (BCS)* problem:

Balanced Connected Subgraph (*BCS*) Problem
Input: A graph $G = (V, E)$, with vertex set $V = V_R \cup V_B$ partitioned into red vertices (V_R) and blue vertices (V_B).
Goal: Find a maximum-cardinality color-balanced subset $V' \subseteq V$ that induces a connected subgraph H.

1.1. Connection with the Graph Motif problem

Here we establish a connection between the *BCS* problem and the *Graph Motif* problem [13,25,34]. In the Graph Motif problem, we are given a graph $G = (V, E)$, a coloring $col : V \to \mathcal{C}$ of the vertices in V where $\mathcal{C}$ is a set of colors, and a multiset M of colors of $\mathcal{C}$; the objective is to find a subset $V' \subseteq V$ such that the induced subgraph on V' is connected and $col(V') = M$, where $col(V')$ denotes the multiset of colors of vertices in V'. We note that if $\mathcal{C}$ ={red, blue} and the motif has same number of blues and reds, then the solution of the Graph Motif problem gives a balanced connected subgraph (not necessarily a maximum balanced connected subgraph). Fellows et al. [25] showed that the Graph Motif problem is NP-complete for trees of maximum degree 3 where the given motif is a colorful set instead of a multiset (that is, no color occurs more than once). They also showed that the Graph Motif problem remains NP-hard for bipartite graphs of maximum degree 4 and the motif contains only two colors. It is easy to observe that a solution to the Graph Motif problem (essentially) gives a solution to the *BCS* problem, with an impact of a polynomial factor in the running time. On the other hand the NP-hardness result for the *BCS* problem on a particular graph class implies the NP-hardness result for the Graph Motif problem on the same class. We conclude that the *BCS* problem is a special case of the Graph Motif problem. Note that much of the work on the Graph Motif problem (e.g., [13,25,34]) is addressing the parameterized complexity of the Graph Motif problem.

1.2. Motivation and possible applications

In a biological population, vertex-colored graphs can be used to represent the connections and interactions between species where different species have different colors. In [25,28], the authors mentioned that the vertex-colored graph problems have numerous applications in bioinformatics (Multiple Sequence Alignment Pipeline or for multiple Protein–Protein Interaction networks). However, the Graph Motif problem is motivated by the applications in biological network analysis [34]. This problem also has applications in social or technical networks [7,25] or in the context of mass spectrometry [12,25].

The *BCS* problem is closely related to the *Maximum Node Weight Connected Subgraph (MNWCS)* problem [22,29]. In the *MNWCS* problem, we are given a connected graph $G(V, E)$, with an integer weight associated with each vertex (node) in V, and an integer bound B; the objective is to decide whether there exists a subset $V' \subseteq V$ such that the subgraph induced by V' is connected and the total weight of the vertices in V' is at least B. In the *MNWCS* problem, if the weight of each vertex is either $+1$ (red) or -1 (blue), and if we ask for a largest connected subgraph whose total weight is *exactly* zero, then it is equivalent to the *BCS* problem. The *MNWCS* problem along with its variations have numerous practical application in various fields (see [22] and the references therein). We believe some of these applications also serve well to motivate the *BCS* problem.

1.3. Related work

Bichromatic input points, often referred to as "red–blue" input, has appeared extensively in numerous problems. Recently Bandyapadhyay et al. [5] studied four fundamental graph theoretic problems: Hamiltonian path, Traveling salesman, Minimum spanning tree, and Minimum perfect matching on geometric graphs induced by bichromatic (red and blue) points. Many of these problems are NP-hard on Euclidean plane [3,14]. In [5], author showed almost all of these problems can be solved in linear time in two restricted settings such as colinear points and equidistant points on a circle. For a detailed survey on geometric problems with red–blue points see [30]. In [11,20,21] colored points have been considered in the context of matching and partitioning problems. In [2], Aichholzer et al. considered the balanced island problem and devised polynomial algorithms for points in the plane. On the combinatorial side, Balanchandran et al. [4] studied the problem of unbiased representatives in a set of bicolorings. Kaneko et al. [31] considered the problem of balancing colored points on a line. Later on, Bereg et al. [6] studied balanced partitions of 3-colored geometric sets in the plane.

Finding a certain type of subgraph in a graph is a fundamental algorithmic question. In [24], Feige et al. studied the dense k-subgraph problem in which we are given a graph G and a parameter k, and the goal is to find a set of k vertices with maximum average degree in the subgraph induced by this set. Crowston et al. [17] considered parameterized algorithms for the balanced subgraph problem. Kierstead et al. [32] studied the problem of finding a colorful induced subgraph in a

properly colored graph. In [19], Derhy and Picouleau considered the problem of finding induced trees in both weighted and unweighted graphs and obtained hardness and algorithmic results. They have studied bipartite graphs and triangle-free graphs; moreover, they have considered the case in which the number of prescribed vertices is bounded.

Recently Bhore et al. [10] studied the *BCS* problem on geometric intersection graphs. They showed NP-hardness on unit-disk graphs, outer-string graphs, complete grid graphs, and unit square graphs. Also they designed polynomial-time algorithms for this problem on interval, circular-arc and permutation graphs. Kobayashi et al. [33] provide an exact exponential-time algorithm of the *BCS* problem for general graphs(in $2^{n/2}n^{\mathcal{O}(1)}$ time). They also consider a weighted version of the *BCS* (called as WBCS) problem and showed weakly NP-hardness on star graphs and strongly NP-hardness on split graphs and properly colored bipartite graphs. Darties et al. [18] prove the NP-completeness of the decision variant of the *BCS* problem in bounded-diameter and bounded-degree graphs: bipartite graphs of diameter four, graphs of diameter three and bipartite cubic graphs. Bhore et al. [9] considered a new version of independent set and dominating set problem on vertex colored interval graphs, called *f-Balanced Independent Set* (f-BIS) and *f-Balanced Dominating Set* (f-BDS). Given a vertex colored graph with k color, a subset of vertices is said to be f-balanced if it contains f vertices from each color class. In the f-BIS and f-BDS problems, the goal is to find an independent set and a dominating set, respectively, that is f-balanced. They showed NP-completeness for both the problems on proper interval graphs.

1.4. Our results

In this paper, we consider the balanced connected subgraph problem on various graph families and present several hardness and algorithmic results.

On the hardness side, in Section 2, we prove that the *BCS* problem is NP-hard on general graphs, even for planar graphs, bipartite graphs (with a general red/blue color assignment, not necessarily a proper 2-coloring), and chordal graphs (a chordal graph is a simple graph that does not contain an induced cycle of length at least four). Furthermore, we show that the existence of a balanced connected subgraph containing a specific vertex is NP-complete. In addition to that, we prove that finding the maximum balanced path in a graph is NP-hard. Note that, Fellows et al. [25] showed that the Graph Motif problem is NP-complete for bipartite graphs with two colors. However, their reduction does not imply that the *BCS* problem on bipartite graph is NP-hard since in their reduction the motif is not color-balanced (i.e., does not include the same number of blues and reds).

On the algorithmic side, in Section 3, we devise polynomial-time algorithms for trees (in $\mathcal{O}(|V|^3)$ time), split graphs (in $\mathcal{O}(|V| + |E|)$ time), bipartite graphs with a proper 2-coloring (in $\mathcal{O}(|V| + |E|)$ time), and graphs with diameter 2 (in $\mathcal{O}(|V| + |E|)$ time). Here, V and E are the set of vertices and edges in the input graphs.

2. Hardness results

2.1. The BCS problem on bipartite graphs

In this section, we prove that the *BCS* problem is NP-hard for bipartite graphs with a general red/blue color assignment, not necessarily a proper 2-coloring. We give a reduction from the *Exact-Cover-by-3-Sets (EC3Set)* problem [26]. In this *EC3Set* problem, we are given a set U with $3k$ elements and a collection S of m subsets of U such that each $s_i \in S$ contains exactly 3 elements. The objective is to find an exact cover for U (if one exists), i.e., a sub-collection $S' \subseteq S$ such that every element of U occurs in exactly one member of S'. During the reduction, we generate an instance $G = (R \cup B, E)$ of the *BCS* problem from an instance $X(S, U)$ of the *EC3Set* problem as follows:

Reduction: For each set $s_i \in S$, we take a blue vertex $s_i \in B$. For each element $u_j \in U$, we take a red vertex $u_j \in R$. Now consider a set $s_i \in S$ containing three elements, u_α, u_β, and u_γ, and add the three edges (s_i, u_α), (s_i, u_β), and (s_i, u_γ) to the edge set E. Additionally, we consider a path of $5k$ blue vertices starting and ending with vertices b_1 and b_{5k}, respectively. Similarly, we consider a path of $3k$ red vertices starting and ending with vertices r_1 and r_{3k}, respectively. We connect these two paths by joining the vertices r_{3k} and b_1 by an edge. Finally, we add edges connecting each vertex s_i with b_{5k}. This completes the construction. See Fig. 1 for the complete construction. Clearly, the numbers of vertices and edges in G are polynomial in terms of the numbers of elements and sets in X; hence, the construction can be done in polynomial time. We now prove the following lemma.

Lemma 1. *The instance X of the EC3Set problem has a solution if and only if the instance G of the BCS problem has a connected balanced subgraph T with $12k$ vertices ($6k$ red and $6k$ blue).*

Proof. Assume that the *EC3Set* problem has a solution. Let S^* be an optimal solution in it. We choose the corresponding vertices of S^* in T. Since this solution covers all u_j's. So we select all u_j's in T. Finally we select all the $5k$ blue and $3k$ red vertices in T, resulting in a total of $6k$ red and $6k$ blue vertices.

On the other hand, assume that there is a balanced tree T in G with $6k$ vertices of each color. The solution must pick the $5k$ blue vertices $b_1, \ldots, b_{5k}$. Otherwise, it exclude the $3k$ red vertices $r_1, \ldots, r_{3k}$, and reducing the size of the solution. Since the graph G has at most $6k$ red vertices, at most k vertices can be picked from the set $s_1, \ldots, s_m$ and need to cover all the $3k$ red vertices corresponding to u_j for $1 \leq j \leq 3k$. Hence, these k sets give an exact cover. □

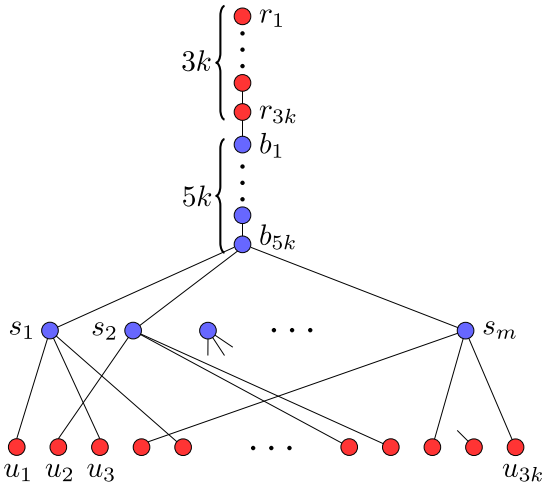

Fig. 1. Construction of the instance G of the *BCS* problem.

It is easy to see that the graph we constructed from the (*EC3Set*) problem in Fig. 1 is indeed a bipartite graph. Hence we conclude the following theorem.

Theorem 1. *The BCS problem is NP-hard for bipartite graphs.*

2.2. NP-Hardness: The BCS problem on special classes of graphs

In this section, we show that the *BCS* problem is NP-hard even if we restrict the graph classes to be planar or chordal graphs.

2.2.1. Planar graphs

In this section we prove that the *BCS* problem is NP-hard for planar graphs. We give a reduction from the *Steiner Tree problem in planar graphs (STPG)* [26]. In this problem, we are given a planar graph $G = (V, E)$, a subset $X \subseteq V$, and a positive integer $k \in \mathbb{N}$. The objective is to find a tree $T = (V', E')$ with at most k edges such that $X \subseteq V'$. Without loss of generality we assume that $k \geq |X| - 1$, otherwise the *STPG* problem has no solution.

Reduction: We generate an instance $H = (R \cup B, E(H))$ for the *BCS* problem from an instance $G = (V, E)$ of the *STPG* problem. We color all the vertices, V, in G as blue. We create a set of $|X|$ red vertices as follows: for each vertex $u_i \in X$, we create a red vertex u'_i in H, and we connect u'_i to u_i via an edge. Additionally, we take a set Z of $(k + 1 - |X|)$ red vertices in H and the edges (z_j, u'_1) into $E(H)$, for each $z_j \in Z$. Hence we have, $B = V$, and $R = Z \cup \{u'_i;\ 1 \leq i \leq |X|\}$. Note that $|R| < |B|$ and $|R| = (k + 1)$. This completes the construction. For an illustration see Fig. 2. Clearly the number of vertices and edges in H are polynomial in terms of vertices in G. Hence the construction can be done in polynomial time. We now prove the following lemma.

Lemma 2. *The STPG problem has a solution if and only if the instance H of the BCS problem has a balanced connected subgraph with $(k + 1)$ vertices each of the two colors.*

Proof. Assume that *STPG* has a solution. Let $T = (V', E')$ be the resulting Steiner tree, which contains at most k edges and $X \subseteq V'$. If $|V'| = (k + 1)$ then the subgraph of H induced by $(V' \cup R)$ is connected and balanced with $(k + 1)$ vertices of each color. If $|V'| < (k + 1)$ then we take a set Y of $((k + 1) - |V'|)$ many vertices from V such that the subgraph of G induced by $(V' \cup Y)$ is connected. Clearly $|V' \cup Y| = (k + 1)$. Now the subgraph of H induced by $(V' \cup Y \cup R)$ is connected and balanced with $(k + 1)$ vertices of each red and blue color.

On the other hand, assume that there is a balanced connected subgraph H' of H with $(k+1)$ vertices of each color. Note that, except vertex u'_1, in H all the red vertices are of degree 1 and connected to blue vertices. Let G' be the subgraph of G induced by all blue vertices in H'. Since H is connected and there is no edge between any two red vertices, G' is connected. Since G' contains $(k + 1)$ vertices, any spanning tree T of H' contains k edges. So T is a solution of the *STPG* problem. □

Theorem 2. *The BCS problem is NP-hard for planar graphs.*

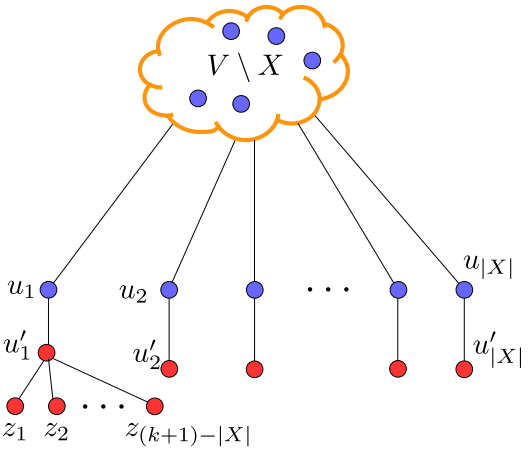

Fig. 2. Schematic construction for planar graphs.

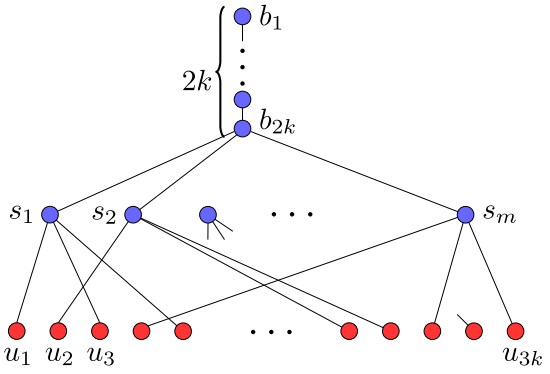

Fig. 3. Construction of the instance G of the *BCS* problem containing b_1.

2.2.2. Chordal graphs

In this section we prove that the *BCS* problem is NP-hard when the input graph is a chordal graph. The hardness construction is similar to the construction in Section 2.1; we modify the construction so that the graph is chordal. In particular, we add edges between s_i and s_j for each $i \neq j$, $1 \leq i, j \leq m$. For this modified graph, it is easy to see that a lemma identical to Lemma 1 holds. Hence, we conclude that the *BCS* problem is NP-hard for chordal graphs.

2.3. NP-Hardness: The BCS problem with a specific vertex

In this section, we prove that the existence of a balanced subgraph containing a specific vertex is NP-complete. We call this problem the *BCS-existence* problem. The reduction is similar to the reduction used in showing the NP-hardness of the *BCS* problem; we also use here a reduction from the *EC3Set* problem (see Section 2.1 for the definition).

Reduction: Assume that we are given a *EC3Set* problem instance $X = (U, S)$, where set U contains $3k$ elements and a collection S of m subsets of U such that each $s_i \in S$ contains exactly 3 elements. We generate an instance $G(R, B, E)$ of the *BCS*-existence problem from X as follows. The red vertices R are the elements $u_j \in U$; i.e., $R = U$. The blue vertices B are the 3-element sets $s_i \in S$; i.e., $B = S$. For each blue vertex $s_i = \{u_\alpha, u_\beta, u_\gamma\} \in S = B$, we add the 3 edges (s_i, u_α), (s_i, u_β), and (s_i, u_γ) to the set E of edges of G. We instantiate an additional set of $2k$ blue vertices, $\{b_1, \ldots, b_{2k}\}$, and add edges to E to link them into a path $(b_1, b_2, \ldots, b_{2k})$. Finally, we add an edge from b_{2k} to each of the blue vertices s_i. Refer to Fig. 3.

Clearly, the number of vertices and edges in G are polynomial in terms of number of elements and sets in the *EC3Set* problem instance X, and hence the construction can be done in polynomial time. We now prove the following lemma.

Lemma 3. *The instance X of the EC3Set problem has a solution iff the instance G of the corresponding BCS existence problem has a balanced subgraph T containing the vertex b_1.*

Proof. Assume that the *EC3Set* problem has a solution, and let S^* be the collection of $k = |S^*|$ sets of S in the solution. Then, we obtain a balanced subgraph T that contains b_1 as follows: T is the induced subgraph of the $3k$ red vertices U, together with the k blue vertices S^* and the $2k$ blue vertices $b_1, \ldots, b_{2k}$. Note that T is balanced and connected and contains b_1.

Conversely, assume there is a balanced connected subgraph T containing b_1. Let t be the number of (blue) vertices of S within T. First, note that $t \leq k$. (Since T is balanced and contains at most $3k$ red vertices, it must contain at most $3k$ blue vertices, $2k$ of which must be $\{b_1, \ldots, b_{2k}\}$, in order that T is connected.) Next, we claim that, in fact, $t \geq k$. To see this, note that each of the t blue vertices of T that corresponds to a set in S is connected by edges to 3 red vertices; thus, T has at most $3t$ red vertices. Now, T has $2k + t$ blue vertices (since it has t vertices other than the path $(b_1, \ldots, b_{2k})$), and T is balanced; thus, T has exactly $2k + t$ red vertices, and we conclude that $2k + t \leq 3t$, implying $k \leq t$, as claimed. Therefore, we need to select exactly k blue vertices corresponding to the sets S, and these vertices connect to all $3k$ of the red vertices. The k sets corresponding to these k blue vertices is a solution for the *EC3Set* problem. □

Clearly, the *BCS* existence problem is in NP. Hence, we conclude the following theorem.

Theorem 3. *It is NP-complete to decide if there exists a connected balanced subgraph that contains a specific vertex.*

2.4. NP-hardness: Balanced Connected Path Problem

In this section we consider the *Balanced Connected Path (BCP)* Problem and prove that it is NP-hard. In this problem instead of finding a balanced connected subgraph, our goal is to find a balanced path with a maximum cardinality of vertices. To prove the *BCP* problem is NP-hard we give a polynomial time reduction from the *Hamiltonian Path (Ham-Path)* problem which is known to be NP-complete [26]. In this problem, we are given an undirected graph Q, and the goal is to find a Hamiltonian path in Q i.e., a path which visits every vertex in Q exactly once. In the reduction we generate an instance G of the *BCP* problem from an instance Q of the *Ham-Path* problem as follows:
Reduction: We make a new graph Q' from Q. Let us assume that the graph Q contains m vertices. If m is even then $Q' = Q$. If m is odd, then we add a dummy vertex u in Q, connect to every other vertices in Q by edges with u and attach a path of length 2 to u. The resulting graph is our desired Q'. It is easy to observe that, Q has a Hamiltonian path if and only if Q' has a Hamiltonian path.

Now we have a *Ham-Path* instance Q' with even number of vertices, say n. We arbitrarily choose any $n/2$ vertices in Q' and color them red and color the remaining $n/2$ vertices blue. Let G be the colored graph. This completes the construction. Clearly, this can be done in polynomial time.

Lemma 4. *Q' has a Hamiltonian path T if and only if G has a balanced path P with exactly n vertices.*

Proof. Assume that Q' has a Hamiltonian path T. This implies that, T visits every vertex in Q'. Since by the construction there are exactly half of the vertices in G is red and remaining are blue, the same path T is balanced with $n/2$ vertices of each color. On the other hand, assume that there is a balanced path P in G with exactly $n/2$ vertices of each color. Since, G has a total of n vertices, the path P visits every vertex in G. Hence, P is a Hamiltonian path. □

Therefore, we have the following observation.

Observation 1. *The BCP problem is NP-hard for general graph.*

3. Algorithmic results

In this section, we consider several graph families and devise polynomial time algorithms for the *BCS* problem. Notice that, if the graph is a path or cycle, the optimal solution is just a path. Hence, one can do brute-force search to obtain the maximum balanced path. In case of a complete graph K_n, we output a sub-graph H of K_n induced by V, where $|V| = 2|B|$, $B \subset V$, and B is the set of all blue vertices in K_n (assuming that, the number of blue vertices is at most the number of red vertices in K_n). Clearly, H is the maximum-cardinality balanced connected subgraph in K_n. We consider trees, split graphs, bipartite graphs (properly colored), graphs of diameter 2, and present polynomial algorithms for each of them.

3.1. Trees

In this section we give a polynomial-time algorithm for the *BCS* problem in the case that the input graph is a tree $T = (V, E)$, with vertices $V = V_R \cup V_B$ colored red (V_R) or blue (V_B). Our goal is to find a maximum-cardinality balanced subtree of T.

Fix an embedding of the tree $T = (V, E)$ in the plane [1]. For a non-leaf vertex $v \in V$ in the tree T, let m be the number of children of v. (If v is a leaf, $m = 0$.) In the embedding of T, assume that the children of each vertex v are drawn in a horizontal row below v, so that we can speak of the left-to-right (i.e., counter-clockwise about v) order of the children,

denoted $u_1, u_2, \ldots, u_m$. For a non-leaf vertex v and $1 \leq i \leq m$, let $T_{v,i}$ denote the subtree of T, rooted at v, consisting of the subtrees rooted at the children $u_1, \ldots, u_i$, together with v and the edges linking v to these i children.

For any vertex $v \in V$ and integer $\Delta \in [-|V_B|, |V_R|]$, let $f_v(\Delta)$ be the maximum number of vertices in a subtree of T rooted at v for which the *red excess* is Δ, where the red excess of a subtree is defined to be the number of red vertices in the subtree minus the number of blue vertices in the subtree. (Thus, a color-balanced subtree has red excess $\Delta = 0$, and a subtree with more blue vertices than red vertices has red excess $\Delta < 0$.) If there is no subtree rooted at v with red excess Δ, then we define $f_v(\Delta) = -\infty$. We solve the BCS problem by tabulating $f_v(\Delta)$ for all choices of $v \in V$ and Δ; there are $\mathcal{O}(n^2)$ values to tabulate. The size of a largest color-balanced subtree rooted at v is $f_v(0)$, and the size of an overall optimal solution to the BCS problem is then given by $\max_v f_v(0)$.

Note that if v is a leaf of T, then $f_v(1) = 1$ if v is red, $f_v(-1) = 1$ if v is blue, and $f_v(\Delta) = -\infty$ otherwise.

If $v \in V$ is not a leaf of T, let $F_v(i, \Delta)$ be the maximum cardinality of a subtree of $T_{v,i}$ that is rooted at v, and has red excess of Δ. Then, for each (non-leaf) v, $f_v(\Delta) = F_v(m_v, \Delta)$, where m_v is the number of children of vertex v.

We tabulate the values $F_v(i, \Delta)$ from the leaves of T upwards, and for increasing values of i.

The main recursion that allows us to tabulate $F_v(i+1, \Delta)$ is obtained by considering all possible red excess values δ for the subtree rooted at child u_{i+1} that might be included in the subtree of $T_{v,i+1}$ rooted at v: We have the option to attach a tree (rooted at v) of size $F_v(i, \Delta - \delta)$ to a tree (rooted at u_{i+1}) of size $f_{u_{i+1}}(\delta)$, via the edge (v, u_{i+1}), resulting in a subtree of $T_{v,i+1}$, rooted at v, with red excess $\Delta - \delta + \delta = \Delta$. We compare this option with that of not using u_{i+1} at all (yielding a tree of size $F_v(i, \Delta)$). Thus, we have

$$F_v(i+1, \Delta) = \max\left\{F_v(i, \Delta), \max_{\delta}\{F_v(i, \Delta - \delta) + f_{u_{i+1}}(\delta)\}\right\}.$$

At the base of this recursion, we compute, for each non-leaf v,

$$F_v(1, \Delta) = \begin{cases} f_{u_1}(\Delta - 1) + 1 & \text{if } v \text{ is red,} \\ f_{u_1}(\Delta + 1) + 1 & \text{if } v \text{ is blue.} \end{cases}$$

The overall evaluation proceeds by first tabulating the values of $f_v(\Delta)$ for all leaves v and all values of Δ. Then, for non-leaf vertices v, we tabulate values of $F_v(i, \Delta)$, for each choice of Δ, for values of $i = 1, 2, \ldots, m_v$.

Now, there are $\mathcal{O}(n)$ choices of the pair v, i (more precisely, there are at most $n-1$ such choices, since these choices correspond to edges of T, linking v to one of its children), and $\mathcal{O}(n)$ choices of Δ (since $-|V_B| \leq \Delta \leq |V_R|$). The optimization in the recursion is over $\mathcal{O}(n)$ choices of δ, so the overall tabulation takes time $\mathcal{O}(n^3)$, using $\mathcal{O}(n^2)$ space.

The actual tree realizing $F_v(i, \Delta)$ or $f_v(\Delta)$ is found by keeping track, during the recursive evaluation of $F_v(i+1, \Delta)$, of whether or not the maximization was achieved using the subtree rooted at u_{i+1} (of size $f_{u_{i+1}}(\delta)$, for an optimal choice of δ), which means using the edge (v, u_{i+1}). This data tells us exactly which children of each vertex v are connected to v in an optimizing tree.

Theorem 4. *Let T be a tree whose n vertices are colored either red or blue. Then, in $\mathcal{O}(n^3)$ time and $\mathcal{O}(n^2)$ space, one can compute a maximum-cardinality balanced subtree of T.*

Remark. The algorithm resulting in Theorem 4 generalizes to the case in which vertices of T may take on colors other than just red and blue, and we desire a maximum-cardinality subtree having any specified target ratio of cardinalities among the vertices of the subtree; the running time is polynomial, for a fixed number c of colors, with time bounded by $n^{\mathcal{O}(c)}$.

3.2. Split graphs

A graph $G = (V, E)$ is defined to be a split graph if there is a partition of V into two sets S and K such that S is an independent set and K is a complete graph. There is no restriction on edges between vertices of S and K. Here we give a polynomial time algorithm for the *BCS* problem where the input graph $G = (V, E)$ is a split graph. Let V be partitioned into S and K where S and K induce an independent set and a clique respectively in G. Also, let S_B and S_R be the sets of blue and red vertices in S, respectively. Similarly, let K_B and K_R be the sets of blue and red vertices in K, respectively. We argue that there exists a balanced connected subgraph in G, having $\min\{|S_B \cup K_B|, |S_R \cup K_R|\}$ vertices of each color.

Note that if $|S_B \cup K_B| = |S_R \cup K_R|$ then G itself is balanced. Now, w.l.o.g., we can assume that $|S_B \cup K_B| < |S_R \cup K_R|$. We will find a connected balanced subgraph H of G, where the number of vertices in H is exactly $2|S_B \cup K_B|$. To do so, we first modify the graph $G = (V, E)$ to a graph $G' = (V, E')$. Then, from G', we will find the desired balanced subgraph with $|S_B \cup K_B|$ many vertices of each color. Moreover, this process is done in two steps.

Step 1: Construct $G' = (V, E')$ from $G = (V, E)$. For each $u \in S_B$, if u is adjacent to at least a vertex u' in K_R, then remove all incident edges with u except the edge (u, u'). Similarly, for each $v \in S_R$, if v is adjacent to at least a vertex v' in K_B, then remove all incident edges with v except the edge (v, v').

Step 2: Let $k = |S_R \cup K_R| - |S_B \cup K_B|$. Now we have following cases.

- **Case 1:** $|S_R| \geq k$. We remove k vertices from S_R in G'. Clearly, after this modification, G' is connected, and we get a balanced subgraph having $|S_B \cup K_B|$ vertices of each color.

- **Case 2:** $|S_R| < k$. Then we know, $|K_R| > |K_B \cup S_B|$. Let $S'_B \subseteq S_B$ be the set of vertices in G' such that each vertex of S'_B has exactly one neighbor in K_R. Then, we take a set $X \subset K_R$ with cardinality $|K_B \cup S_B|$ such that X contains all adjacent vertices of S'_B. Now we take the subgraph H of G' induced by $(S_B \cup K_B \cup X)$. Now H is optimal and balanced.

Running time: Step 1 takes $\mathcal{O}(|E|)$ time to construct G' from $G = (V, E)$. Now in step 2, both Case 1 and Case 2 take $\mathcal{O}(|V|)$ time to delete $|S_R \cup K_R| - |S_B \cup K_B|$ vertices from G'. Hence, the total time taken is $\mathcal{O}(|V| + |E|)$. We conclude in the following theorem.

Theorem 5. *Given a split graph $G = (V, E)$, with r red and b blue ($|V| = r + b$) vertices, then, in $\mathcal{O}(|V| + |E|)$ time we can find a balanced connected subgraph of G having $\min\{b, r\}$ vertices of each color.*

3.3. Bipartite graphs, properly colored

In this section, we describe a polynomial-time algorithm for the *BCS* problem where the input graph is a bipartite graph whose vertices are colored red/blue according to proper 2-coloring of vertices in a graph. We show that there is a balanced connected subgraph of G having $\min\{b, r\}$ vertices of each color where G contains r red vertices and b blue vertices. Note that we earlier showed that the *BCS* problem is NP-hard in bipartite graphs whose vertices are colored red/blue arbitrarily; here, we insist on the coloring being a proper coloring (the construction in the hardness proof had adjacent pairs of vertices of the same color). We begin with the following lemma.

Lemma 5. *Consider a tree T (which is necessarily bipartite) and a proper 2-coloring of its vertices, with r red vertices and b blue vertices. If $r < b$, then T has at least one blue leaf.*

Proof. We prove it by contradiction. Let there is no blue leaf. Now assign any blue vertex, say b_r, as a root. Note that it always exists. Now b_r is at level 0 and b_r has degree at least 2. Otherwise, b_r is a leaf with blue color. We put all the adjacent vertices of b_r in level 1. This level consists of only red vertices. In level 2 we put all the adjacent vertices of level 1. So level 2 consists of only blue vertices. This way we traverse all the vertices in T and let that we stop at kth-level. k cannot be even as all the vertices in even level are blue. So k must be odd. Now for each $0 \leqslant i \leqslant \frac{k-1}{2}$, in the vertices of (level $2i \cup$ level $(2i+1)$), number of blue vertices is at most the number of red vertices. Which leads to the contradiction that $r < b$. Hence there exists at least one leaf with blue color. □

Now we describe the algorithm. We first find a spanning tree T in G. If $r = b$ then T itself is a maximum balanced subtree (subgraph also) of G. Without loss of generality assume that $r < b$. So by Lemma 5, T has at least 1 blue leaf. Now we remove that blue leaf from T. Using similar reason, we repetitively remove $(b - r)$ blue vertices from T. Finally, T becomes balanced subgraph of G, with r vertices of each color.
Running time: Finding a spanning tree in $G = (V, E)$ requires $\mathcal{O}(|E|)$ time. To find all the leaves in the tree T requires $\mathcal{O}(|V|)$ time (breadth first search). Hence the total time needed is $\mathcal{O}(|V| + |E|)$.
Now, we state the following theorem.

Theorem 6. *Given a bipartite graph $G = (V, E)$ with a proper 2 coloring (r red or b blue vertices), then in $\mathcal{O}(|V| + |E|)$ time we can find a balanced connected subgraph in G having $\min\{b, r\}$ vertices of each color.*

3.4. Graphs of diameter 2

In this section, we give a polynomial time algorithm for the *BCS*-problem where the input graph has diameter 2. Let $G(V, E)$ be such a graph which contains b blue vertex set B and r red vertex set R. We find a balanced connected subgraph H of G having $\min\{b, r\}$ vertices of each color. Assume that $b < r$. This can be done in two phases. In phase 1, we generate an induced connected subgraph G' of G such that (i) G' contains all the vertices in B, and (ii) the number of vertices in G' is at most $(2b - 1)$. In phase 2, we find H from G'.

Phase 1. To generate G', we use the following observation regarding graphs of diameter 2.

Observation 2. *Let $G = (V, E)$ be a graph of diameter 2. Then for any pair of non adjacent vertices u and v from G, there always exists a vertex w such that both $(u, w) \in E$ and $(v, w) \in E$.*

We first include B in G'. Now we have the following two cases.
Case 1: The induced subgraph $G[B]$ of B is connected. In this case, G' is $G[B]$.
Case 2: The induced subgraph $G[B]$ of B is not connected. Assume that $G[B]$ has $k(> 1)$ components. Let $B_1, B_2, \ldots, B_k$ be k disjoints sets of vertices such that each induced subgraph $G[B_i]$ of B_i in G is connected. Now using Observation 2, any two vertices $v_i \in B_i$ and $v_j \in B_j$ are adjacent to a vertex say $u_\ell \in R$. We repetitively apply Observation 2 to merge all the k subgraphs into a larger graph. We need at most $(k - 1)$ red vertices to merge k subgraph. We take this larger graph as the graph G'.

Phase 2. In this phase, we find the balanced connected subgraph H with b vertices of each color. Note that the graph G' generated in phase 1 contains b blue and at most $(b-1)$ red vertices. Assume that G' contains b' red vertices. We add $(b-b')$ red vertices from $G \setminus G'$ to G'. This is possible since G in connected.
Running time: In phase 1, first finding all the blue vertices and its induced subgraph takes $\mathcal{O}(|V|+|E|)$ time. Now to merge all the k components into a single component which is G' needs $\mathcal{O}(|E|)$ time. In phase 2, adding $(b-b')$ red vertices to G' takes $\mathcal{O}(|E|)$ time as well. Hence, total time requirement is $\mathcal{O}(|V|+|E|)$.

Theorem 7. *Given a graph $G=(V,E)$ of diameter 2, where the vertices in G are colored either red or blue. If G has b blue and r red vertices then, in $\mathcal{O}(|V|+|E|)$ time we can find a balanced connected subgraph in G having $\min\{b,r\}$ vertices of each color.*

4. Conclusion

We have studied the problem of finding a largest size (cardinality of the vertex set) balanced connected subgraph in a simple connected graph. We have seen that this problem is NP-complete for bipartite graphs, chordal graphs, or planar graph. We have given polynomial time algorithms for solving this problem for trees, graphs with proper 2 coloring, split graphs, and graphs with diameter 2. So the obvious question is can other special classes of graphs be found to yield polynomial time algorithms? For example, outer planar graphs, regular graphs, graphs with small treewidth, graphs with small bandwidth, etc. Let G be a given graph and OPT be the number of vertices in an optimal solution of the *BCS* problem. Is there any polynomial time (α,β) approximation algorithm which yields a solution H such that minimum number of blue and red vertices in H is at least $\alpha \times OPT$, where $0 < \alpha \leq 1$ and difference between the number of blue and red vertices in H is at most β, where $\beta > 0$?

CRediT authorship contribution statement

Sujoy Bhore: Conceptualization, Writing - original draft. **Sourav Chakraborty:** Conceptualization. **Satyabrata Jana:** Writing - original draft, Methodology, Writing - review & editing, Project administration. **Joseph S.B. Mitchell:** Writing - original draft, Methodology, Writing - review & editing, Funding acquisition. **Supantha Pandit:** Conceptualization, Writing - original draft, Methodology, Writing - review & editing, Project administration, Funding acquisition. **Sasanka Roy:** Conceptualization.

Acknowledgments

We thank Florian Sikora for pointing out the connection with the Graph Motif problem. The authors would like to thank the anonymous reviewers for their valuable comments to improve the quality of the paper.

References

[1] M. Abellanas, J. Garcia-Lopez, G. Hernández-Peñalver, M. Noy, P.A. Ramos, Bipartite embeddings of trees in the plane, Discrete Appl. Math. 93 (2–3) (1999) 141–148.
[2] O. Aichholzer, N. Atienza, R. Fabila-Monroy, P. Perez-Lantero, J.M. Dıaz-Báñez, D. Flores-Peñaloza, B. Vogtenhuber, J. Urrutia, Balanced Islands in two colored point sets in the plane, 2015, arXiv preprint arXiv:1510.01819.
[3] S. Arora, Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems, J. ACM 45 (5) (1998) 753–782.
[4] N. Balachandran, R. Mathew, T.K. Mishra, S.P. Pal, System of unbiased representatives for a collection of bicolorings, 2017, arXiv preprint arXiv:1704.07716.
[5] S. Bandyapadhyay, A. Banik, S. Bhore, M. Nöllenburg, Geometric planar networks on bichromatic points, in: CALDAM 2020, in: Lecture Notes in Computer Science, vol. 12016, Springer, 2020, pp. 79–91.
[6] S. Bereg, F. Hurtado, M. Kano, M. Korman, D. Lara, C. Seara, R.I. Silveira, J. Urrutia, K. Verbeek, Balanced partitions of 3-colored geometric sets in the plane, Discrete Appl. Math. 181 (2015) 21–32.
[7] N. Betzler, R. van Bevern, M.R. Fellows, C. Komusiewicz, R. Niedermeier, Parameterized algorithmics for finding connected motifs in biological networks, IEEE/ACM Trans. Comput. Biol. Bioinform. 8 (5) (2011) 1296–1308.
[8] S. Bhore, S. Chakraborty, S. Jana, J.S.B. Mitchell, S. Pandit, S. Roy, The balanced connected subgraph problem, in: CALDAM 2019, in: Lecture Notes in Computer Science, vol. 11394, Springer, 2019, pp. 201–215.
[9] S. Bhore, J. Haunert, F. Klute, G. Li, M. Nöllenburg, Balanced independent and dominating sets on colored interval graphs, 2020, CoRR abs/2003.05289.
[10] S. Bhore, S. Jana, S. Pandit, S. Roy, Balanced connected subgraph problem in geometric intersection graphs, in: COCOA 2019, in: Lecture Notes in Computer Science, vol. 11949, Springer, 2019, pp. 56–68.
[11] A. Biniaz, A. Maheshwari, M.H. Smid, Bottleneck bichromatic plane matching of points, in: CCCG, 2014.
[12] S. Böcker, F. Rasche, T. Steijger, Annotating fragmentation patterns, in: S.L. Salzberg, T. Warnow (Eds.), Algorithms in Bioinformatics, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 13–24.
[13] E. Bonnet, F. Sikora, The Graph Motif problem parameterized by the structure of the input graph, Discrete Appl. Math. 231 (2017) 78–94.
[14] M.G. Borgelt, M.J. van Kreveld, M. Löffler, J. Luo, D. Merrick, R.I. Silveira, M. Vahedi, Planar bichromatic minimum spanning trees, J. Discrete Algorithms 7 (4) (2009) 469–478.
[15] K. Cameron, Induced matchings, Discrete Appl. Math. 24 (1–3) (1989) 97–102.
[16] J. Cheriyan, S. Maheshwari, Finding nonseparating induced cycles and independent spanning trees in 3-connected graphs, J. Algorithms 9 (4) (1988) 507–537.

[17] R. Crowston, G. Gutin, M. Jones, G. Muciaccia, Maximum balanced subgraph problem parameterized above lower bound, Theoret. Comput. Sci. 513 (2013) 53–64.
[18] B. Darties, R. Giroudeau, J. König, V. Pollet, The balanced connected subgraph problem: Complexity results in bounded-degree and bounded-diameter graphs, in: COCOA 2019, in: Lecture Notes in Computer Science, vol. 11949, Springer, 2019, pp. 449–460.
[19] N. Derhy, C. Picouleau, Finding induced trees, Discrete Appl. Math. 157 (17) (2009) 3552–3557.
[20] A. Dumitrescu, R. Kaye, Matching colored points in the plane: some new results, Comput. Geom. 19 (1) (2001) 69–85.
[21] A. Dumitrescu, J. Pach, Partitioning colored point sets into monochromatic parts, Int. J. Comput. Geom. Appl. 12 (05) (2002) 401–412.
[22] M. El-Kebir, G.W. Klau, Solving the maximum-weight connected subgraph problem to optimality, 2014, CoRR abs/1409.5308.
[23] B. Escoffier, L. Gourvès, J. Monnot, Complexity and approximation results for the connected vertex cover problem in graphs and hypergraphs, J. Discrete Algorithms 8 (1) (2010) 36–49.
[24] U. Feige, D. Peleg, G. Kortsarz, The dense k-subgraph problem, Algorithmica 29 (3) (2001) 410–421.
[25] M.R. Fellows, G. Fertin, D. Hermelin, S. Vialette, Upper and lower bounds for finding connected motifs in vertex-colored graphs, J. Comput. Syst. Sci. 77 (4) (2011) 799–811.
[26] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman, 1979.
[27] J. Håstad, Clique is hard to approximate within $n^{1-\epsilon}$, in: 37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October, 1996, IEEE Computer Society, 1996, pp. 627–636.
[28] G.F. Italiano, Y. Manoussakis, N.K. Thang, H.P. Pham, Maximum colorful cliques in vertex-colored graphs, in: COCOON 2018, in: Lecture Notes in Computer Science, vol. 10976, Springer, 2018, pp. 480–491.
[29] D.S. Johnson, The NP-completeness column: An ongoing guide, J. Algorithms 6 (1) (1985) 145–159.
[30] A. Kaneko, M. Kano, Discrete geometry on red and blue points in the plane—a survey—, in: Discrete and Computational Geometry, Springer, 2003, pp. 551–570.
[31] A. Kaneko, M. Kano, M. Watanabe, Balancing colored points on a line by exchanging intervals, J. Inf. Process. 25 (2017) 551–553.
[32] H.A. Kierstead, W.T. Trotter, Colorful induced subgraphs, Discrete Math. 101 (1–3) (1992) 165–169.
[33] Y. Kobayashi, K. Kojima, N. Matsubara, T. Sone, A. Yamamoto, Algorithms and hardness results for the maximum balanced connected subgraph problem, in: COCOA 2019, in: Lecture Notes in Computer Science, vol. 11949, Springer, 2019, pp. 303–315.
[34] V. Lacroix, C.G. Fernandes, M. Sagot, Motif search in graphs: Application to metabolic networks, IEEE/ACM Trans. Comput. Biol. Bioinform. 3 (4) (2006) 360–368.
[35] J. Wu, H. Li, On calculating connected dominating set for efficient routing in ad hoc wireless networks, in: Proceedings of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, 1999, pp. 7–14.