Complexity and inapproximability results for balanced connected subgraph problem

Timothée Martinod, Valentin Pollet, Benoit Darties, Rodolphe Giroudeau, Jean-Claude König

Complexity and inapproximability results for Balanced Connected Subgraph Problem

T. Martinod, V. Pollet, B. Darties, R. Giroudeau, and J.-C. König

LIRMM, Univ Montpellier, CNRS, Montpellier, France
{timothee.martinod,rodolphe.giroudeau,benoit.darties,
valentin.pollet,konig}@lirmm.fr

Abstract. This work is devoted to the study of the Balanced Connected Subgraph Problem (BCS) from a complexity, inapproximability and approximation point of view. The input is a graph $G = (V, E)$, with each vertex having been colored, "red" or "blue"; the goal is to find a maximum connected subgraph $G' = (V', E')$ from G that is color-balanced (having exactly $|V'|/2$ red vertices and $|V'|/2$ blue vertices). This problem is known to be $\mathcal{NP}$-complete in general but polynomial in paths and trees. We propose a polynomial-time algorithm for block graph. We propose some complexity results for bounded-degree or bounded-diameter graphs, and also for bipartite graphs. We also propose inapproximability results for some graph classes, including chordal, planar, or subcubic graphs.

Keywords: Complexity · approximation · color-balanced subgraph

1 Introduction

In this paper, we consider the combinatorial optimization problem Balanced Connected Subgraph. Given a connected bichromatic graph $G = (V, E)$ - a graph having its vertices colored with two colors blue or red, not necessary proper - , the goal is to determine a maximum subset $V' \subseteq V$ such that V' is balanced - with the same number of blue and red vertices - and subgraph induced by V' is connected.

Balanced Connected Subgraph (BCS)
Input: A graph $G = (V, E)$, with vertices set $V = V_{blue} \cup V_{red}$ partitioned into sets of blue and red vertices respectively.
Question: Find $V' = (V'_{blue} \cup V'_{red}) \subseteq V$, such that $V'_{blue} \subseteq V_{blue}$ *and* $V'_{red} \subseteq V_{red}$, that induces a maximum connected subgraph H with $|V'_{blue}| = |V'_{red}|$.

1.1 Related work

Many well-studied combinatorial optimization problems consist in finding induced subgraphs with a given property. For instance, finding a maximum clique or a maximum independent set are one of the 21 $\mathcal{NP}$-complete problems classified by [12]. [9] describe a general version of these problems (GT21-22): maximum induced (connected) subgraph with property Π. If Π is hereditary and non-trivial then the problem is $\mathcal{NP}$-complete and some approximability results hold.

In this article, we investigate the Balanced Connected Subgraph problem (BCS) as introduced by [3]. Given a 2-colored graph (using colors red and blue), find the largest connected subgraph containing as many vertices of each color. Notice that the property of being color-balanced is far from being hereditary, hence the need for an *ad-hoc* study.

BCS remains $\mathcal{NP}$-complete in bipartite graphs, chordal graphs and planar graphs [3]. They also gave polynomial algorithms solving BCS in quadratic time for splits, graphs of diameter two and properly colored bipartite graphs, and in time $O(n^4)$ for trees. The research on this problem remains very active; recently two recent articles related to balanced connected subgraph were proposed [2,5]: they design polynomial-time algorithms for the BCS problem on interval $O(n^4 \log n)$, circular-arc $O(n^6 \log n)$ and permutation graphs $O(n^3)$. The problem remains hard even for unit-disk graphs.

In [13], the authors show that BCS can be solved in $O(n^2)$-time for trees (improving the complexity given in [3]) and in $O(n^3)$-time for interval graphs . The former result can be extended to bounded treewidth graphs. They also consider a weighted version of BCS (WBCS). They prove that this variant is weakly $\mathcal{NP}$-hard even on star graphs and strongly $\mathcal{NP}$-hard even on split graphs and properly colored bipartite graphs, whereas the unweighted counterpart is tractable on those graph classes. Finally, they propose a exact exponential-time algorithm for general graphs with time complexity $2^{n/2}n^{O(1)}$. Their algorithm is based on a variant of Dreyfus-Wagner algorithm for the Steiner Tree problem.

As they point out, BCS is strongly related to the Maximum Weight Connected Subgraph (MWCS) problem mentioned by [11]. Note that BCS is neither a special case nor a generalization of MWCS. In MWCS, the goal is to find a connected subgraph of maximum weight. If BCS were to be formulated as a MWCS with weights say $+1$ for red vertices and -1 for blue vertices, we would search for the largest subgraph of weight exactly 0.

BCS is also related to Steiner Tree problem. In fact, assume that you are given a graph G along with a 2-coloration (using colors red and blue) of its vertices (less red vertices than blue vertices). Asking whether there exists a BCS in G containing all the red vertices can be seen as a special case of Steiner Tree: the red vertices are the terminals, and we search for a Steiner Tree of size r twice the number of terminals. In that case, one can determine the existence of a BCS containing all the red vertices by using efficient exact algorithms for Steiner. If in Steiner tree there is more red than blue vertices, we can always complete with blue vertices (the neighboring vertices of the Steiner tree are blue) to obtain a BCS.

The Graph Motif (GM) problem is tied to BCS as well. GM consists, given a colored graph G and a multiset of colors M, in finding a connected subgraph such that the multiset of colors assigned to its vertices is exactly M. Finding a balanced connected subgraph of size at least $2k$ can be reduced to a polynomial number of motif searches in a 2-colored graph: all one has to do is to search for the motif {`red,...,red, blue,..., blue`} with k occurrences of red and blue, then $k+1$ occurrences of each, and so on, upon either finding a balanced connected subgraph or having proved that none exists.

GM was first introduced by [15] in the context of metabolic networks. They showed that GM is $\mathcal{NP}$-complete even if the input graph is restricted to be a tree. Fellows et al. [7,8] further proved that GM remains $\mathcal{NP}$-complete in trees of maximum degree 3, and even if the input graph is a 2-colored bipartite graph of maximum degree four. As a positive result, they gave an $\mathcal{FPT}$ algorithm for GM parameterized by the size of the motif in the general case. Since BCS can be solved by solving a polynomial number of instances of GM, using their $\mathcal{FPT}$ algorithm to do so would result in an $\mathcal{FPT}$ algorithm for BCS parameterized by the size of the solution.

Consider an optimization variant of GM: find the largest connected sub-graph which multi-set of colors is included in the given motif [6]. Moreover we proved this variant to be $\mathcal{APX}$-hard in trees of maximum degree three.

A rather comprehensive list of applications for MWCS can be found in [3]. While we do not motivate BCS with further practical applications, we believe it may prove to be useful in network design applications (where the colors represent roles assigned to the vertices), social data-mining (colors represent classes of individuals), or even electoral applications (see [1] for a study of *gerrymandering* as a graph partitioning problem with a red-blue colored graph as input).

On the other hand, BCS appears of interest in a purely theoretical point of view. The problem is quite hard complexity-wise, and can be generalized in a lot of different ways. For instance, one can loosen the "balanced" constraint and ask for a connected subgraph minimizing the ratio between the number of red vertices and blue vertices. Other generalizations would be increasing the number of colors in the input coloration, enforcing the subgraph being looked for to have additional properties (being a path, tree, 2-connected ...), or coloring the edges instead of the vertices.

1.2 Our contributions

The contributions of this paper are summarized by Table 1.

We study the computational complexity of BCS for bichromatic graph, in some restricted cases, namely bipartite graphs of diameter four, graphs of diameter three, bipartite sub-cubic graphs ... In each case, we establish the $\mathcal{NP}$-completeness of BCS by polynomial-time reduction from well-known problems and we propose some inapproximability results according to some topologies. We propose a approximation with non constant ratio. We extend the hardness for 3-colored graph even if the diameter is two.

Organization of the paper In the next section, we propose a polynomial time algorithm for block graph. Section 3 is dedicated to $\mathcal{NP}$-completeness and non-approximation proofs in bounded-diameter graphs: bipartite graphs of diameter four, and graph of diameter three. Complexity results is extended to the case of three colors and diameter two is also presented. Section 4 focuses on bipartite

Topology Or Parameter	Complexity	Approximation results
Block graph	$\mathcal{P}$ - $O(n^5)$ - Theo. 1	-
$D = 3$ Bipartite $D = 4$	$\mathcal{NPC}$ Theo. 4 $\mathcal{NPC}$ Theo.2	
Planar $\Delta \leq 4$ Bipartite $\Delta \leq 4$ Bipartite $\Delta \leq 3$	$\mathcal{NPC}$ [3] $\mathcal{NPC}$ Theo. 7 $\mathcal{NPC}$ Theo. 6	No-$\mathcal{APX}$ Theo. 8 No-$\mathcal{APX}$ Cor.2
Chordal	$\mathcal{NPC}$ [3] and Theo. 9	No-$\mathcal{APX}$ Cor. 5

Table 1: Complexity and inapproximability results discussed in this paper for BCS according to structural parameter and/or graph topology.

(sub)cubic graphs, on planar graph with bounded degree in complexity and inapproximation viewpoint.

2 Polynomial time algorithm for block graph

A block graph is a graph where all the biconnected components are cliques. Recall that block graphs are a particular case of chordal graphs. In this part, we give a polynomial-time algorithm for the BCS problem on block graphs. We focus on this graph class since for trees the problem admits a polynomial-time algorithm while it is $\mathcal{NP}$-complete for chordal graphs [3].

Let $G = (V, E)$ be a block graph with $V_{red} < V_{blue}$. We begin by observing that, at least one articulation point of G belongs to one of optimal solutions for each instance. Indeed, if optimal solution has no articulation point, it is part of only one clique. Then this clique contains vertices of the solution of both colors. So we can exchange one of the articulation points without changing the size of the solution. Thus if at least one vertex of a clique is in the optimal solution then there is also at least one articulation point in it.

Let Q be the set of all blue articulation points, and let $G' = (V', E')$ be the graph G induced by $Q \cup V_{red}$. Clearly, we have $V'_{red} = V_{red}$. Without loss of generality, we suppose that $V'_{red} < V'_{blue}$ (otherwise there is a trivial optimal solution containing $V'_{red} \cup V'_{blue}$ and $|V'_{red}| - |V'_{blue}|$ random vertices from $_{blue} - Q$).

Lemma 1. *For each optimal solution on a graph G, there exists a solution on G of same size which is included into graph G'.*

Proof. Let S be an optimal solution on graph G, we note S_{red} (resp. S_{blue}) the red vertices (resp. blue vertices) from S.

Recall that $|S_{red}| = |S_{blue}|$, and $|S_{red}| \leq |V_{red}| = |V'_{red}| < |V'_{blue}|$, implying that $|S_{blue}| < |V'_{blue}|$. If there exists a blue vertex $x \in S_{blue} - V'_{blue}$ then $x \notin Q$, and can be replaced in S by any other vertex y from $V'_{blue} - S_{blue}$. Note y always exists as $|V'_{blue}| > 1$. Thus one obtain a solution S^* included in V'.

As a consequence of Lemma 1, it is sufficient to compute a optimal solution on G' instead of on G.

Observation 1 *Let us consider an optimal solution S of G'. Let us construct a breadth-first search tree in G' restricted to vertices from S and starting from an* *articulation point x of G'. This tree is contained in the breadth-first search tree starting in G' starting from the same articulation point x but not restricted to vertices from S.*

Based on Lemma 1 and Observation 1, we propose Algorithm 1 for solving block graph instances. This algorithm consists in (1) deleting the edge between two vertices if none is an articulation point (2) generating a spanning tree for each articulation point following a breadth-first search, and (3) computing an optimal solution on each spanning tree using algorithm proposed in [3].

Let $BFS(G, x)$ be a spanning tree of G computed by following a breadth-first search from x. Finally, $bhore(T)$ is the optimal solution obtained on T using algorithm [3].

Algorithm 1: Polynomial time algorithm on block graphs

```
Data: a connected bichromatic block graph G = (V, E)
Result: a subgraph G′ ⊆ G of order |G′|
G′ ← ∅;
/* Step 1: we delete the edge between two vertices if none is an
   articulation point.                                              */
for {x, y} ∈ E do
    if G[V − {x}] is connected and G[V − {y}] is connected] then
        E ← E − {x, y}
end
/* Step 2: we generate a spanning tree for each articulation point
   following a breadth-first search                                 */
for x ∈ V do
    if G[V − {x}] is not connected then
        /* we generate a spanning tree for each articulation point
           following a breadth-first search.                        */
        T ← BFS(G, x);
        /* We compute a solution on T with Bhore's algorithm        */
        S ← bhore(T);
        if |S| > |G′| then
            G′ = T;
end
return G′
```

Theorem 1. *Let G be a connected bichromatic block graph. Then one can compute a BCS of G in $O(n^5)$ time.*

Proof. First note that Bhore's improved algorithm proposed in [3], Lemma 5, produces all possible balanced subtrees rooted at a root t in $O(n^4)$ time complexity for trees.

We suppose G contains at least one articulation point - otherwise the solution is trivial as G is a clique.

It is not difficult to see that for each articulation point x, all vertex sets that contain x and induce a connected subgraph of G also induce a connected subgraph of T, thus all solutions containing x are considered by our algorithm. Since there exists an optimal solution that contains an articulation point, this proves the correctness of our algorithm.

For each articulation point x we construct a BFS-spanning tree rooted on x (at most n trees) and run Bhore's improved algorithm on it.

There will necessarily be at least a spanning tree that contains the optimal solution for G.

The overall time complexity is $O(n^5)$.

Corollary 1. *In a recent result proposed in [14], BCS can be solved in $O(n^2)$ time for trees. Using this result we can improve the time complexity of our algorithm to $O(n^3)$.*

3 Bounded-diameter graphs

3.1 Bipartite graphs of diameter four

In the following, we prove that BCS remains $\mathcal{NP}$-complete in graphs of diameter four. The reduction is based on Dominating Set in graphs of diameter two, which is $\mathcal{NP}$-complete [16]. The following construction transforms any graph of diameter two into a 2-colored bipartite graph of diameter four.

> Dominating Set (DM)
> **Input:** $G = (V, E)$ and k a integer.
> **Question:** Does G contain $D \subset V(G)$ s.t. $|D| \leq k$ and for all $x \in V(G)$, either $x \in D$ or $x \in N(y)$ for some $y \in D$?

Construction 1 *Let $G = (V, E)$ be a graph on n vertices and $k \in \mathbb{N}$. We build $G' = (V', E')$ an instance of BCS as follows:*

- *add $2n$ blue vertices $V_1 = \{v_1^1, \dots, v_n^1\}$ and $V_2 = \{v_1^2, \dots, v_n^2\}$;*
- *add $n + k$ red vertices $Q = \{q_1, \dots, q_n\}$ and $P = \{p_1, \dots, p_k\}$;*
- *for all $i \in [n]$, add the edge $v_i^2 q_i$;*
- *for all $i \in [n]$, for all $w \in N[v]$ add the edge $v_i^1 w^2$;*
- *for all $i \in [n]$, for all $j \in [k]$ add the edge $v_i^1 p_j$.*

Construction 1 is clearly done in polynomial time and illustrated by Figure 1. If the base graph G has diameter two, then the graph G' obtained after transformation has diameter four. Indeed, each couple $(x, y) \in V_1 \times V_1$ or $V_2 \times V_2$ can be connected by a path of length two between x and y. It follows that each couple $(x, y) \in V_1 \times V_2$ can be connected by a path of length at most three. Finally, since all the red vertices have a neighbor in V_1 or V_2, and $G'[P \cup V_1]$ is a complete bipartite graph, any pair of red vertices can be connected by a path of length at most four.

Theorem 2. Balanced Connected Subgraph *remains $\mathcal{NP}$-complete on bipartite graphs of diameter four.*

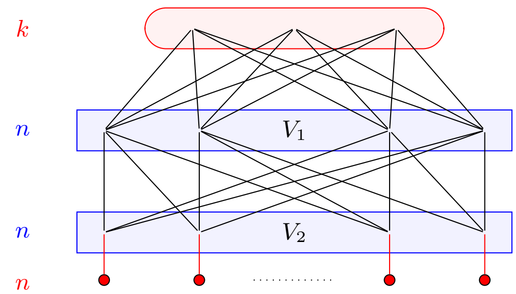


Fig. 1: Building an instance of BCS of diameter 4 from a graph of diameter 2.

Proof. Let G be a graph of diameter two and $k \in \mathbb{N}^*$. Let G' be the graph obtained from G using 1. Let us prove that G contains a dominating set of size k if and only if G' has a BCS of size $2(n+k)$.

⇒ if G contains a dominating set D of size k, then let $D_1 \subset V'$ be the vertices of V_1 in G' corresponding to D. $B = \{v \in V' : v \text{ is red }\} \cup V_2 \cup D_1$ contains $n+k$ red vertices and $n+k$ blue vertices. Since D is a dominating set in G, every vertex in V_2 is connected to a vertex of D_1. $G'[B]$ is thus connected and balanced.

⇐ if G' has a BCS S of size $2(n+k)$, then it has to include all the red vertices. For the pendant red vertices to be connected in S, S must include V_2. Since V_2 is of size n, S contains exactly k other blue vertices, and those vertices belong to V_1. Moreover, V_2 being an independent set and S being connected, every vertex of V_2 must be connected to at least one vertex in $V_1 \cap S$. The vertices of G corresponding to $V_1 \cap S$ in G' thus form a dominating set of size k.

Since DM is $\mathcal{NP}$-complete in graphs of diameter two [16], and BCS being in $\mathcal{NP}$, the discussion above proves the theorem.

We propose to extend previous complexity result to derive a lower bound for Exact Algorithms. Assuming $\mathcal{ETH}$, there is no $2^{o(n)}$ time algorithm for DM [17] and since Construction 1 leads a linear transformation we obtain the following Theorem.

Theorem 3. *Assuming $\mathcal{ETH}$, there is no $2^{o(n)}$ time algorithm for* Balanced Connected Subgraph *in presence of bipartite graphs of diameter four.*

3.2 Graphs of diameter three

To prove that BCS remains $\mathcal{NP}$-hard in graphs of diameter three, we design a reduction from Colorful Connected Subgraph which is stated as follows.

Colorful Connected Subgraph (CCS)
Input: $G = (V, E)$ a p-colored graph, $k \in \mathbb{N}$ s.t. $k \leq p$.
Question: Does G contain a connected subgraph of size at least k which each color appears at most once in ?

To the best of our knowledge, the complexity of CCS – as stated above – is not clearly established. To show that CCS is $\mathcal{NP}$-complete, we use a result given by [7] on the Graph Motif problem. Recall that GM consists, given a colored graph $G = (V, E)$ and a multiset of colors $\mathcal{M}$, in finding a connected subgraph of G which multiset of colors is exactly $\mathcal{M}$.

[7] show in their Theorem 1 that GM remains $\mathcal{NP}$-hard even if the motif $\mathcal{M}$ is colorful, that is each color appears at most once in $\mathcal{M}$. In their reduction, the instance of GM they obtain is such that the motif $\mathcal{M}$ is exactly the set of all colors. Therefore, GM remains $\mathcal{NP}$-complete even if $\mathcal{M}$ is the set containing each color once. Now, one can observe that CCS is $\mathcal{NP}$-hard because in case p is equal to k, then CCS is equivalent to GM in the aforementioned case. The following lemma holds, since CCS is clearly in $\mathcal{NP}$.

Lemma 2. Colorful Connected Subgraph *is $\mathcal{NP}$-complete.*

We now reduce CCS to BCS in graphs of diameter three. The idea of the construction is to create a clique containing one red vertex and "a lot" of blue ones for each color. The cliques are then interconnected, making sure that any pair of cliques is connected by at least one edge.

Construction 2 *Let $G = (V, E)$ be a graph and $c : V \to \{1, \ldots, p\}$ a p-coloring of its vertices. We build $G' = (V', E')$ a 2-colored graph in the following way (refer to Figure 2).*

- *$V' = C_1 \cup C_2 \ldots \cup C_p$ with $C_i = \{x \in V : c(x) = i\} \cup \{r_i\}$.*
  - *For all $i \in \{1, \ldots, p\}$, $G'[C_i]$ is connected as a clique, and each one of its vertices, except r_i, is blue.*
- *For all $uv \in E$, add the corresponding edge to G'.*
- *For all C_i, C_j such that there is no edge between C_i and C_j, add a blue vertex x_i^j to C_i (connected to every vertex in C_i) and a blue vertex x_j^i to C_j (connected to every vertex in C_j), as well as the edge $x_i^j x_j^i$.*

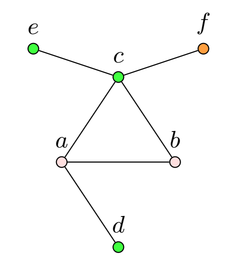
Construction 2 can be applied in polynomial time. The resulting graph has diameter three since it is composed of pairwise connected cliques. Figure 2 gives a example.

Theorem 4. Balanced Connected Subgraph *is $\mathcal{NP}$-complete in graphs of diameter three.*

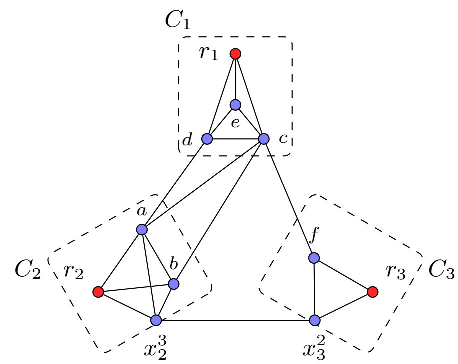
Proof. Let $G = (V, E)$ be a graph and $c : V \to \{1, \ldots, p\}$ a p-coloring of its vertices. Let $G' = (V', E')$ be the 2-colored graph obtained by applying Construction 2 to G. Let $k \in \mathbb{N}$, $k \geq 3$. We claim that G has a CCS of size at least k if and only if G' has a BCS of size at least $2k$.

$\Rightarrow$ If G has a CCS, S, of size at least k, then the corresponding vertices in G' are blue and induce a connected subgraph intersecting each clique at most once. For each clique that S intersects, just add said clique's red vertex to S. Doing so, we build a balanced connected subgraph in G' of size at least $2k$.

$\Leftarrow$ If G' has a BCS, S, of size at least $2k$ then its contains at least k red vertices belonging to at least k different cliques. Since the neighborhoods of



(a) 3-colored graph.



(b) Graph obtained after applying Construction 2.

Fig. 2: Applying Construction 2 to a 3-colored graph. The clique C_1 corresponds the green vertices c, d and e, the clique C_2 to the pink vertices a and b, and the clique C_3 to the orange vertex f. Vertices x_2^3 and x_3^2 originate from the absence of a yellow-pink edge and ensure r_2 and r_3 are connected by a path of length three.

red vertices are pairwise disjoint, each red vertex must have exactly one blue neighbor (in its clique) belonging to S. Assume that a vertex x_i^j belongs to S, then it is the sole neighbor of r_i. In order to connect r_i to other red vertices, x_j^i has to belong to S. Now since x_j^i is assumed to belong to S, r_j cannot have any other blue neighbor in S. Under those assumptions, S cannot be of size greater than four which absurd because we assumed $k \geq 3$. Therefore, vertices of type x_i^j cannot belong to S, and every blue vertex in S corresponds to a vertex in G. Since the red vertices have degree 1 in S, removing them from S does not break the connectivity and thus the set of blue vertices in S corresponds to a connected subgraph in G. Since S contains at most one blue vertex per clique, the set of blue vertices of S is a CCS of size at least k in G.

The previous discussion concludes the polynomial-time reduction from CCS to BCS. Since the instance of BCS obtained through Construction 2 have diameter three, the theorem holds.

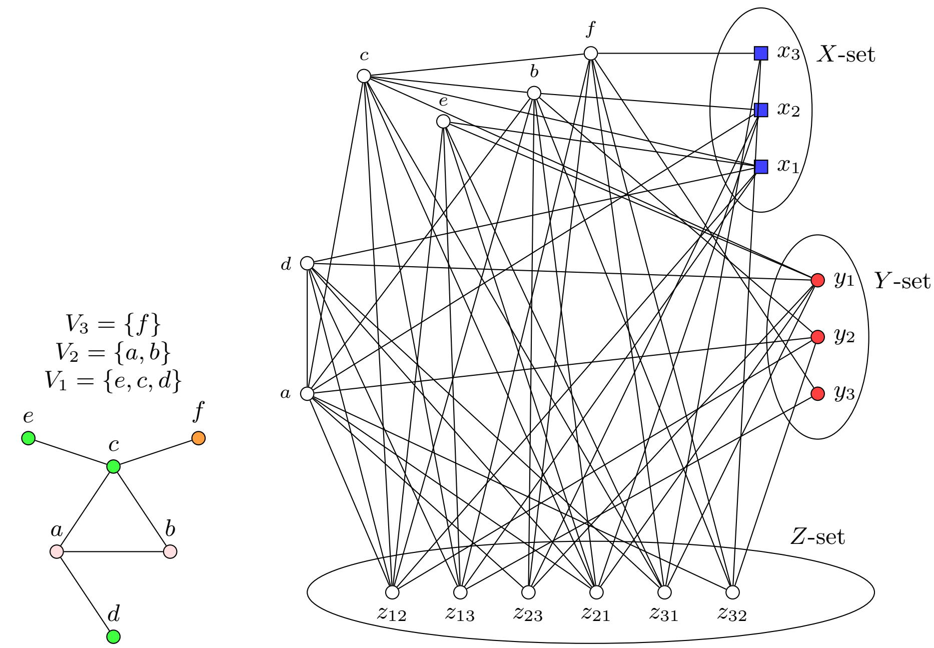
3.3 Three colors and diameter two

In this section we extend the model by considering a 3-colored graph instead of bichromatic. In such context we present a $\mathcal{NP}$-complete result for graph admitting a diameter two. Recall that for two colors and diameter two the problem admits a polynomial-time algorithm [3]. Clearly, for diameter one the problem is trivial.

Construction 3 *Let I be an instance of CCS with k-colors, we will construct an instance I' of BCS with three colors in the following way:*

Let G an input graph and let V_i be the set of vertices with color i in an instance I of CCS.

- *We add $2k$ vertices $X = \{x_1, x_2, \ldots, x_k\}$ and $Y = \{y_1, y_2, \ldots, y_k\}$ to the graph G.*



(a) 3-colored graph. (b) Graph obtained after applying Construction 3.

Fig. 3: Encoding an instance of CCS into an instance of BCS with 3 colors. The resulting graph admits a diamater two.

- *A edge from x_i-vertex (resp. y_i-vertex) to V_i-vertices is added.*
- *We add $k(k-1)$ vertices $Z = \{z_{ij} | i \neq j, i, j \in \{1, \ldots, k\}\}$.*
- *For each z_{ij}-vertex we add edge from z_{ij} to x_i (resp. y_j) and add a complete graph between z_{ij} and V_i, $\forall i$ (resp. $V_j, \forall j$).*
- *The X-vertices (resp. Y-vertices) are colored in blue (resp. red). All remaining vertices are colored in white.*

The resulting graph is denoted by $H = (V_H, E_H)$. This construction is accomplished in polynomial-time. The transformation is illustrated by Figure 3.

Lemma 3. *The graph H obtained by Construction 3 admits a diameter two.*

Proof.
- Since vertex $\forall i$ x_i are neighbors of vertices z_{ij} so in two steps the sets X, Y, V and Z are visited.
- Each vertex z_{ij} is neighbor of vertices z_{kl} (going through z_{il}), V_i (by x_i) and V_j (by y_j), x_k (by z_{kj}), y_k (by z_{ik}).
- From V_i-vertex to V_j-vertex going through x_i if $i = j$ by z_{ij} otherwise.

Theorem 5. Balanced Connected Subgraph *is $\mathcal{NP}$-complete for three colors even if the diameter is two.*

Proof. Let $G = (V, E)$ be a graph and $c : V \to \{1, \ldots, p\}$ a p-coloring of its vertices. Let $H = (V_H, E_H)$ be the 3-colored graph obtained by applying Construction 3 to G.

BCS with three colors is in $\mathcal{NP}$. Assume that $k \in \mathbb{N}$ and $k \geq 3$.

We claim that G has a CCS of size at least k if and only if H has a BCS of size at least $3k$. Let S^* be the set of the BCS-vertices.

$\Rightarrow$ If G has a CCS denoted S, of size at least k, we obtain a BCS of size at least $3k$ by taking the X-set (resp. Y-set) and the k vertices in graph G solution of CCS.

$\Leftarrow$ If H has a BCS denoted S of size at least $3k$. Let S_{white} be the set of white vertices in the solution S.

All X-vertices and Y-vertices are in S. Each white vertex from $V \cup Z$ admits two neighbors in $X \cup Y$. Since there are only k white-vertices, the white neighborhoods $N(S_{white}) \cap (X \cup Y)$ are disjoints.

1. Assume that $z_{ij} \in S$. Therefore, the following vertices cannot be in S: none vertex from V_i (resp. V_j), none vertex $z_{ik}, \forall k \neq i$, (resp. $z_{kj}, \forall k \neq j$).
   So the graph H cannot be connected.
2. So, the solution $S \cap V_i \neq 0$. The X-vertices and Y-vertices admit a degree one.
   The subgraph induced by $S^* - (X \cup Y)$ contains a unique vertex in each V_i and remains connected.

4 Bounded-degree graphs

4.1 Cubic bipartite

We prove BCS to be $\mathcal{NP}$-complete in cubic bipartite graphs by reduction from Exact Cover by 3-Sets-3 [10].

> Exact Cover by 3-Sets-3 (X3C3)
> **Input:** A universe $X = (x_1, \ldots, x_{3q})$ and a collection $C = (c_1, \ldots, c_{m=3q})$ of triples of X such that each element in X belongs to exactly three triples of C.
> **Question:** Does G contain $D \subset V(G)$ s.t. $|D| \leq k$ and for all $x \in V(G)$, either $x \in D$ or $x \in N(y)$ for some $y \in D$?

The construction consists in encoding each set by a blue subgraph (a set-gadget) and the elements by red vertices. The graph is then completed by making sure there are less red vertices than blue vertices. In the end, the starting X3C3 instance is positive if and only if there is a BCS containing all the red vertices in the constructed graph.

Construction 4 *Let (X, C) be an instance of X3C3, $C = \{c_1, \ldots, c_{3q}\}$ and $X = \{x_1, \ldots, x_{3q}\}$, with q even. Let $G = (V, E)$ be an instance of BCS obtained from (X, C) as follows (see Figure 4).*

- *For each $x_i \in X$ add a vertex x_i (an element-vertex);*
- *for each $c_i \in C$, $c_i = \{x_j, x_k, x_l\}$:*
  - *add a blue gadget H_i on 14 vertices. Denote t_i^1, t_i^2 and t_i^3 (resp. s_i^1, s_i^2 and s_i^3) the vertices of degree 2 at the top (resp. bottom) of H_i;*
  - *add the edges $s_i^1 x_j$, $s_i^2 x_k$ and $s_i^3 x_l$;*

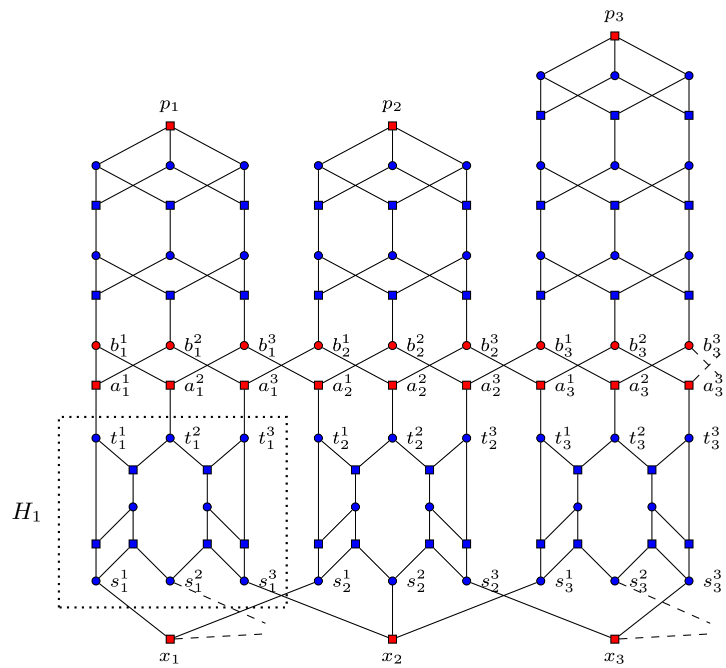


Fig. 4: Encoding an instance of X3C3 into an instance of BCS (bipartite cubic). From bottom to top – $3q$ red element-vertices; $3q$ blue set-gadgets on 14 vertices; $3q$ red connectivity-gadgets on 6 vertices connected as an accordion; $3q$ chains of 2 to 3 blue 2-regular bipartite graphs on 6 vertices; and $3q$ red terminal vertices.

- *add* 6 *red vertices partitioned into* $A_i = \{a_i^1, a_i^2, a_i^3\}$ *and* $B_i = \{b_i^1, b_i^2, b_i^3\}$. *Add the edges* $a_i^1 t_i^1$, $a_i^2 t_i^2$, $a_i^3 t_i^3$ $a_i^1 b_i^2$, $a_i^3 b_i^2$, $a_i^2 b_i^1$ *and* $a_i^2 b_i^3$*;*
- *add* 2 *(if* $i \leq \frac{q}{2}$*) or* 3 *(elsewise)* 2*-regular bipartite graphs on* 6 *blue vertices connected by matchings. Connect the first of these graphs to* B_i *by a matching. Enforce* 3*-regularity by adding a red terminal vertex* p_i *connected to the remaining* 3 *vertices of degree* 2*;*

– *for each* $i \in \{1, \ldots, 3q-1\}$*, add the edges* $a_i^3 b_{i+1}^1$ *and* $b_i^3 a_{i+1}^1$*;*
– *add the edges* $a_1^1 b_1^1$ *and* $a_{3q}^3 b_{3q}^3$*.*

Observe that the graph G obtained through Construction 4 is bipartite: in Figure 4 bipartiteness is depicted by the shape of the vertices, edges all have one square-shaped endpoint and one circle-shaped endpoint. G is also cubic. It contains $24q$ red vertices and $93q$ blue vertices. The construction is done in polynomial time. The following lemma proves that in order to connect all the red terminal vertices (p_i) to the red vertices B_i, "a lot" of blue vertices are mandatory.

Lemma 4. *Let* (X, C) *be an instance of X3C3,* $C = \{c_1, \ldots, c_{3q}\}$ *and* $X = \{1, \ldots, 3q\}$*. Let* $G = (V, E)$ *be an instance of BCS obtained from* (X, C) *through Construction 4. Let* S *be a BCS of* G *of size* $48q$*. Then* S *contains at most* $7q$ *blue vertices belonging to set-gadgets.*

Proof. Since $|S| = 48q$, S contains all the red vertices. In particular S contains $\{p_1, \ldots, p_{3q}\}$. Since $G[S]$ is connected, each p_i must be connected to B_i. Denote λ_i the number of internal vertices in a shortest path from p_i to B_i in $G[S]$. Observe that λ_i is at least 4 if $i \leq \frac{q}{2}$, and at least 6 otherwise. The number of vertices required to connect $\{p_1, \ldots, p_{3q}\}$ to the rest of S is:

$$\sum_{i=1}^{3q} \lambda_i = \sum_{i=1}^{\frac{q}{2}} \lambda_i + \sum_{i=\frac{q}{2}+1}^{3q} \lambda_i \tag{1}$$

$$\geq 4 \times \frac{q}{2} + 6 \times \frac{5q}{2} \tag{2}$$

$$\geq 17q \tag{3}$$

Since S contains $24q$ blue vertices and at least $17q$ form paths connecting $\{p_1, \ldots, p_{3q}\}$ to $\{B_1, \ldots, B_{3q}\}$, the number of vertices belonging to set-gadgets cannot exceed $7q$ thus the lemma holds.

We now prove the existence and unicity of a solution to an integer linear program that will describe how "expensive" (in the number of blue vertices) it is to connect all the element-vertices to the other red vertices.

Lemma 5. *Let $q \in \mathbb{N}$ and (L) be the following integer linear program:*

$$\begin{array}{ll} \min & 7u_1 + 5u_2 + 3u_3 \\ \hline s.t. & \\ & 3u_1 + 2u_2 + u_3 = 3q \quad (1) \\ & \\ & u_1, u_2, u_3 \in \mathbb{N} \end{array}$$

$u_2 = u_3 = 0$ and $u_1 = q$ is the unique optimal solution to (L).

Proof. Let (u_1, u_2, u_3) be an optimal solution to (L).

- If $u_2 > 0$, then $u_3 = 0$ because if $u_3 > 0$ then $(u_1 + 1, u_2 - 1, u_3 - 1$ has better cost which is absurd). Assuming $u_3 = 0$, the constraint (1) becomes $3u_1 + 2u_2 = 3q$ which implies $u_2 \geq 3$ since $u_2 > 0$. Now $(u_1 + 2, u_2 - 3, u_3)$ has lower cost, this is absurd therefore $\underline{u_2 = 0}$.
- Assume $u_3 > 0$. With $u_2 = 0$, the constraint (1) becomes $3u_1 + u_3 = 3q$ which implies $u_3 \geq 3$. Now $(u_1 + 1, u_2, u_3 - 3)$ has lower cost, this is absurd therefore $\underline{u_2 = 0}$.

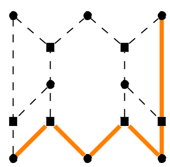
We now have $u_2 = u_3 = 0$ and necessarily $u_1 = q$ to satisfy (1) and the lemma holds.

Theorem 6. Balanced Connected Subgraph*is $\mathcal{NP}$-complete in bipartite (sub)cubic graphs.*

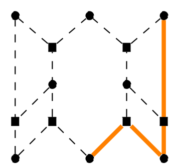
Proof. Let (X, C) be an instance of X3C3 and $G = (V, E)$ the graph obtained from (X, C) through Construction 4. Let us prove that (X, C) is positive if and only if G contains a BCS of size $48q$.

$\Leftarrow$ If G contains a BCS, S, of size $48q$ we claim that S contains exactly q paths of length 7, spread across q distinct set gadgets. To prove this, we reason on how the set-gadgets are used to connect elements. Denote $R \subset V$ the set of red vertices in G, and $R^{\uparrow} \subset R$ the red vertices that are not element-vertices.

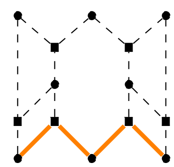
Since $|S| = 48q$, S contains all the red vertices. In particular, S contains all the element-vertices. For an element x to be connected to $R^{\uparrow}$, S must contain a path from x to $R^{\uparrow}$ and that path goes through vertices of set-gadgets. Either the vertices of this path belong to a unique set-gadget, or they belong to several. In case the path spreads across several set-gadgets, it must go through other element-vertices, one of them being connected to $R^{\uparrow}$ by a path belonging to a unique set-gadget.



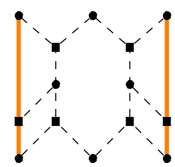
(a) Type-1



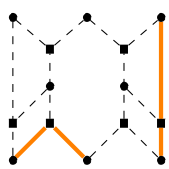
(b) Type-2



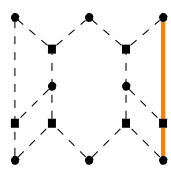
(c) Type-2



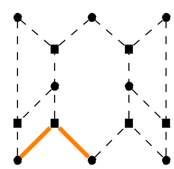
(d) Type-2



(e) Type-2



(f) Type-3



(g) Type-3

Fig. 5: Different types of gadget usage in a BCS. Type-1 gadgets connect 3 element to the top using at least 7 vertices. Type-2 gadgets connect 2 elements to the top using at least 5 vertices. Type-3 gadgets connect 1 element to the top using at least 3 vertices.

Now, see Figure 5. We classify the gadgets in three types, depending on how they connect elements in S. We abuse notation by saying that S contains a gadget when we actually mean that S contains some vertices of said gadget.

- Type-1 gadgets connect three elements to $R^{\uparrow}$. At least seven vertices of those gadgets must belong to S.
- Type-2 gadgets connect two elements. Here, we have three cases:
  - the two elements are connected to $R^{\uparrow}$ by a structure reaching one of the top vertices of the gadget (see Figure 5b and Figure 5d);
  - one of them directly to the top and the other to another element (Figure 5e)
  - the two elements are connected to another element (Figure 5c)

  In all cases, S contains at least five vertices per type-2 gadget.

– Type-3 gadgets connect 1 element. Either directly to $R^{\uparrow}$ or to another element somehow connected. In both cases, S contains at least 3 vertices per type-2 gadget.

Denote u_1 the number of type-1 gadgets in S, u_2 the number of type-2 gadgets and u_3 the number of type-3 gadgets. By construction, $3u_1 + 2u_2 + u_3 = 3q$. In addition, the number of blue vertices in S belonging to set-gadgets is at least $7u_1 + 5u_2 + 3u_3$. (u_1, u_2, u_3) is thus a solution to the linear program (L) described in Lemma 5 and $7u_1 + 5u_2 + 3u_3$ is the objective function of (L). Since (L) has a unique optimal solution costing $7q$, (u_1, u_2, u_3) must be optimal because otherwise $7u_1 + 5u_2 + 3u_3 > 7q$ which is absurd by Lemma 4. We thus have $u_1 = q$ and $u_2 = u_3 = 0$ i.e. S contains exactly q type-1 gadgets, and for each of those gadgets, exactly 7 blue vertices belong to S. Since each type-1 gadget connects 3 red vertices, the set of type-1 gadgets in S corresponds to an exact cover in (X, C), therefore if G contains a BCS of size $48q$ then (X, C) is positive.

$\Rightarrow$ If (X, C) is positive, then there exists q sets in C covering X exactly. Take all the red vertices in G, and for each set in the exact cover pick a path of length 7 in the corresponding set gadget (see Figure 5a). Connect the red accordion to the sinks p_i using exactly $17q$ blue vertices. The structure thus obtained is a BCS of size $48q$ in G.

BCS belonging to $\mathcal{NP}$, and X3C3 being $\mathcal{NP}$-complete, the discussion above proves the $\mathcal{NP}$-completeness in cubic bipartite graphs. Observe that some edges incident to the terminal vertices p_i can be removed, and these deletions do not impact the reduction, therefore the problem remains $\mathcal{NP}$-complete on subcubic graphs, the theorem holds.

Inapproximability bipartite with bounded degree In this section, we show that BCS is not approximable within any constant ratio, even if restricted to bipartite graphs of maximum degree four. We reuse the construction given in [3], when the authors prove that finding a BCS containing a specific vertex is $\mathcal{NP}$-complete. The reduction is based on X3C problem which is $\mathcal{NP}$-complete [10].

EXACT COVER BY 3-SETS (X3C)
Input: A universe $X = (x_1, \ldots, x_{3q})$ and a collection $C = (c_1, \ldots, c_m)$
Question: Is there a subcollection $C' \subset C$ such that each element of X is contained in exactly one subset of C'?

The idea is to encode an X3C instance into a bipartite graph (the classical graph representation of an X3C instance) plus a "long" path, and connect all the vertices encoding sets to one extremity of the path. The X3C instance is then positive if and only if there is a BCS containing the whole path. We then use this construction in a gap reduction by making copies of the obtained graph. For the sake of clarity, Theorem 7 states inaproximability in bipartite graphs, we then give clues to the reader as to why the result holds for bipartite graphs of maximum degree four.

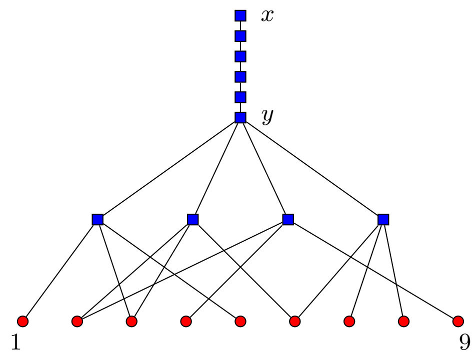


Fig. 6: Construction of an instance of BCS from an instance of X3C. Here $C = (\{1,3,5\},\{2,3,6\},\{2,4,9\},\{6,7,8\})$, $X = \{1,\dots,9\}$.

Construction 5 *Let (X, C) be an instance of X3C, $X = (x_1,\dots,x_{3q})$, $C = (c_1,\dots,c_m)$. We build $G = (V, E)$ an instance of BCS as follows:*

- *for all $x_i \in X$, add a red vertex x_i to G (element-vertex);*
- *for all $c_i \in C$, add a blue vertex c_i to G (set-vertex), and for all $x_k \in c_i$, add the edge $x_k c_i$ to G;*
- *add a blue vertex y to G along with the edges yc_i for all $c_i \in C$;*
- *add a blue path of length $2q$ starting at y and call x the last vertex of the path.*

G contains $5q + m$ vertices and $4m + 2q - 1$ edges.

The transformation is done in polynomial time, and illustrated by Figure 6.
Lemma 6 uses the construction above to reduce X3C to BCS with a compulsory vertex (*à la* [3]).

Lemma 6. *Let (X, C) be an instance of X3C and $G = (V, E)$ the graph obtained through Construction 5. There is a BCS containing x (of size $6q$) in G if and only if (X, C) is a positive instance.*

Proof. $\Rightarrow$ Let X be a BCS containing x. x being balanced and connected, it contains the path from x to y, then some blue set-vertices and finally some red element-vertices. Let l be the number of set-vertices contained in X. X contains $2q+l$ blue-vertices. For it to be balanced it has to contain $2q+l$ red vertices and therefore at least $\frac{2q+l}{3}$ set-vertices (because sets contain three elements). We thus have $l \geq \frac{2q+l}{3}$ giving $l \geq q$.
On the other hand, X contains $2q + l$ red vertices but there are only $3q$ of them, we thus have $l \leq q$.
In the end, $l = q$ and X contains exactly q set-vertices and all the element-vertices, the corresponding sets thus form a solution to (X, C).
$\Leftarrow$ Trivial.

Construction 6 *Let (X, C) be an instance of X3C and $G = (V, E)$ the graph obtained from (X, C) through Construction 5. We build an instance G' of BCS using ρ copies of G, $(G^0,\dots,G^\rho)$, and adding a path connecting all the copies by their x-vertex. Each x-vertex for each copy of $G^i, \forall i \in \{0,\dots,\rho\}$, i.e. the last* *vertex of a path beginning by y in Construction 5 will be denoted by $x^i, \forall i \in \{0, \ldots, \rho\}$ in the following.*

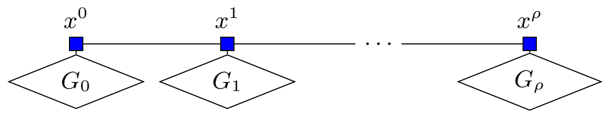


Fig. 7: Building an arbitrarily large instance of BCS for bipartite bounded graph from an instance of X3C by considering x^j-vertex $\forall j$. A similar graph is also used for planar bounded (resp. chordal) graph inapproximation result.

This construction is illustrated by Figure 7.

Theorem 7. *BCS in bipartite graphs is not approximable within any constant ratio unless $\mathcal{P} = \mathcal{NP}$.*

Proof. Assume that there exists $\rho \in \mathbb{R}^+$ and a polynomial-time running ρ-approximation algorithm $\mathcal{A}$ for BCS *i.e.* for any instance of BCS, if we denote K the size of the solution returned by $\mathcal{A}$ and K^* the size of an optimal one then we have $\rho \geq \frac{K^*}{K}$.

Let $p = \rho + 1$ the number of G-copies.

Now run $\mathcal{A}$ on G' and denote K the size of solution it returns. Let K^* be the size of an optimal solution. Using Lemma 6, we have that either $K^* > 6q$ and (X, C) is a yes-instance, or $K^* = 6q$ and (X, C) is a no-instance. Indeed, if $K^* > 6q$ then this solution contains red vertices taken from at least two copies of G: G^i and G^j. For the solution to be connected, it must contain x^i and x^j. Observe that, by construction, a connected subgraph of G containing x contains necessarily more blue vertices than red vertices. Now, using Lemma 6, since x^i belongs to the solution, (X, C) is positive. Conversely if $K^* = 6q$, it means that there is no BCS in G reaching the x vertex and therefore (X, C) is a no-instance.

– If $K \leq 6q$ then $K^* \leq 6\rho q < 6pq$ because A is $\rho - approximate$. In this case we can conclude that (X, C) is a no-instance.
– If $K > 6q$ then $K^* > 6q$ and (X, C) is a yes-instance.

In both cases, $\mathcal{A}$ solves X3C in polynomial time. That is absurd unless $\mathcal{P} = \mathcal{NP}$. Moreover, the graph G' is bipartite by construction, the theorem thus holds.

Construction 5 can be adapted to obtain a bipartite graph of maximum degree four. Instead of an instance of X3C, start with an instance of X3C3. After the transformation, add some dummy sets and elements to ensure that the number of sets is a power of two. Add a full binary tree rooted at y which leaves are the set-vertices. Add pendant red vertices to all the internal nodes of the tree. Under those assumptions, Lemma 6 still hold and so does Theorem 7.

Corollary 2. Balanced Connected Subgraph *in bipartite graphs of maximum degree four is not approximable within any constant ratio unless $\mathcal{P} = \mathcal{NP}$.*

Inapproximability for planar graph with bounded degree In this part we consider the BCS problem in planar graph with bounded degree. We show that there is no hope to find an efficient approximation algorithm with a constant ratio. The proof is based on the Steiner Tree problem [12].

> Steiner Tree (ST)
> **Input:** A graph $G_{st} = (V_{st}, E_{st})$, a subset $X \subset V_{st}$, an integer k.
> **Question:** Is there $T \subset E_{st}$ such that $|T| \leq k$, $G_{st}[T]$ is connected and $X \subset V(G_{st}[T])$?

ST problem remains $\mathcal{NP}$-complete even for subcubic planar graph [4].

Construction 7 *Let (G_{st}, X, k) be an instance of ST with $G_{st} = (V_{st}, E_{st})$, $X \subseteq V$, $|V| = n$ and $|X| = m$. We generate $G = (V, E)$ an instance of BCS as follows:*

- *Let G be a copy of G' in which all vertices are blue.*
- *For each blue-vertex $x_i \in X$, add a red vertex x'_i and the edge $x_i x'_i$.*
- *For each vertex x'_i, take a path of n red vertices starting at y^i_1 ending by y^i_n.*
- *Create a path of red vertex of size $k + 1 - m$) beginning at z_1 and ending at $z_{(k+1)-m}$.*
- *Add an edge $y^1_n z_1$.*
- *Add a path of blue vertex of size mn beginning at w_1 and ending at w_{mn}.*
- *Add an edge $y^m_n w_1$.*

G' contains $2mn + n + k + 1$ vertices.

Assume that w_{mn} lies on the boundary face of a plane embedding. Lemma 7 uses Construction 7 above (illustrated by Figure 8) to reduce an instance of ST to an instance of BCS with an obligatory vertex.

Lemma 7. *Let (G_{st}, X, k) be a instance of ST and $G = (V, E)$ the graph obtained by Construction 7. There is a connected balanced subgraph H containing w_{mn} of size $2((k + 1) + mn)$ if and only if (G_{st}, X, k) is a yes-instance.*

Proof. Assume that there is a connected balanced subgraph H, of size $2((k+1)+mn)$ containing w_{mn}. Since H is balanced, connected and has $2((k + 1) + mn)$ vertices including w_{mn}, it must contain all the path from w_1 to w_{mn}, all the red vertices from G, the m vertices of X and l vertices from $V \backslash X$. Thus H contains exactly $l = ((k + 1) - m)$ vertices from $V \backslash X$, which corresponds to a positive solution for (G_{st}, X, k).

Assume that S is yes-instance of (G_{st}, X, k). The subgraph H is composed of the path from w_1 to w_{mn}, all the red vertices and the vertices corresponding to S. It is connected, balanced and has a size of $2((k + 1) + mn)$.

Construction 8 *Let (G_{st}, X, k) be a instance of ST. We consider the graph $\mathbb{G}$, obtained by $\rho + 1$ copies of G (the graph G is the graph obtained using Construction 7) and the path between the w_{mn}-vertex of all copies. As the same as previously, each w_{mn}-vertex for each copy of $G^i, \forall i \in \{0, \ldots, \rho\}$, will be denoted by $w^i_{mn}, \forall i \in \{0, \ldots, \rho\}$ in the following.*

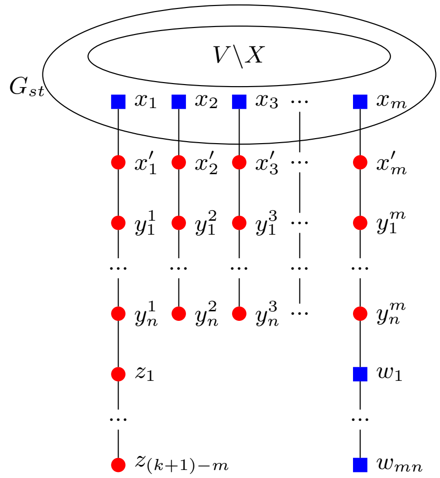


Fig. 8: Graph G obtained from Construction 7.

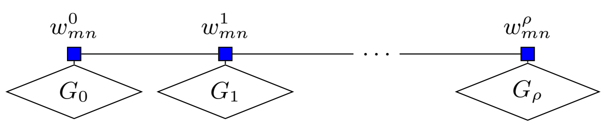


Fig. 9: Building an arbitrarily large instance of BCS for planar bounded graph from an instance of ST.

This construction is illustrated by Figure 9, by considering $w_{mn}^j - vertex$.

Theorem 8. Balanced Connected Subgraph *problem cannot be approximated by any constant factor for planar graphs with an obligatory vertex unless $\mathcal{P} = \mathcal{NP}$.*

Proof. Assume that there is $\rho \in \mathbb{R}^+$ and an ρ-approximate algorithm A in polynomial-time for the BCS in planar graphs. Let $p = \rho + 1$. Let K be the size of the solution obtained by A on $\mathbb{G}$ and K^* the size of the optimal solution. Thanks to Lemma 7 we have either $K^* > 2((k+1)+mn)$ and (G_{st}, X, k) is a yes-instance, or $K^* \leq 2((k+1)+mn)$ and (G_{st}, X, k) is a no-instance.

Thus if $K^* > 2p((k+1)+mn)$, like $K \geq \frac{K^*}{\rho}$, then $K > 2((k+1)+mn)$. In contrast if $K^* \leq 2((k+1)+mn)$ then $K \leq 2((k+1)+mn)$.

So, using the ρ-approximation algorithm A we can distinguish between yes- and no-instances, thus solving the ST problem.

The prescription vertex condition for w_{mn} can be replaced by a blue vertex u_1, a red vertex u_2 and the edges $w_{mn}u_1$, u_1u_2. Thus the search for a balanced graph of size $2(k+1+mn)$ containing w_{mn} becomes a search for a balanced graph of size $2(k+1+mn+1)$. We now have the following corollary.

Corollary 3. Balanced Connected Subgraph *problem cannot be approximated by any constant factor for planar graphs unless $\mathcal{P} = \mathcal{NP}$.*

The ST problem is still $\mathcal{NP}$-complete for subcubic planar graphs [4], so we can state the following corollary.

Corollary 4. Balanced Connected Subgraph *problem cannot be approximated by any constant factor for planar graphs with maximum degree four unless* $\mathcal{P} = \mathcal{NP}$.

4.2 Inapproximability results for chordal graph

In this section, we show that BCS problem cannot be approximated by any constant factor unless $\mathcal{P} = \mathcal{NP}$ for chordal graph. Recall first that BCS remains $\mathcal{NP}$-complete even for chordal graph [3].

Construction 9 *Let* (X, C) *be an instance of X3C, with* $X = (x_1, \ldots, x_{3q})$ *and* $C = (C_1, \ldots, C_m)$*. We generate an instance* $G = (V, E)$ *of BCS from the instance* (X, C) *of X3C as follows:*

- $\forall x_i \in X$*, create a red vertex* x_i *(vertex-element), and* $\forall c_i \in C$*, create a blue vertex* c_i *(vertex-collection).*
- $\forall x_k \in c_i$*, take the edge* $x_k c_i$*, and* $\forall c_i, c_j \in C$*, add the edge* $c_i c_j$*.*
- *Create a blue vertex* y *and the edges* yc_i $\forall c_i \in C$*.*
- *Add a blue path of size* $2q + 1$*, beginning at* $t_1, \ldots t_{2q-2}, z$ *and* z'*.*
- *Create a red vertex* z'' *and the edge* $z'z''$*.*

G contains $5q + m + 2$ *vertices within* $1 + 3q$ *red vertices.*

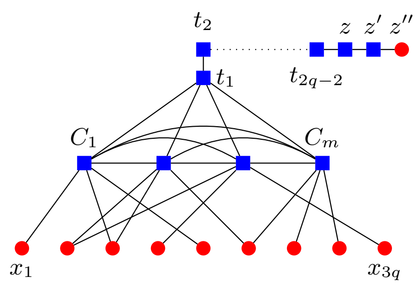


Fig. 10: An instance of BCS obtained by Construction 9 from an instance (X, C) of X3C.

Lemma 8 uses Construction 9 (based on the construction proposed in [3]) above (illustrated by Figure 10) to reduce an instance X3C to an instance of BCS.

Lemma 8. *Let* (X, C) *be an instance of X3C and* $G = (V, E)$ *the graph obtained by Construction 9. There is a BCS denoted* H *of size* $6q + 2$ *in* G *if and only if* (X, C) *admits yes-instance solution.*

Proof. Assume that there is a connected balanced subgraph H of size $6q + 2$ in G. Since H is balanced, connected and has $6q + 2$ vertices, it must contain the vertex z, the path from y to x, l blue vertex-collections and all the $3q$ red vertex-elements. Since $l = q$, H contains exactly q vertex-collections and all the vertex-elements, we can trivially obtain a positive solution for (X, C).

On the other hand, assume that instance (X, C) admits a positive solution S. The subgraph H composed of the path from z'' to t_1, all the vertex-elements and the vertex-collections corresponding to the collections in S is a BCS of G of size $6q + 2$.

Recall that the following Theorem was proposed first by [3] (see Lemma 3 of Section 2.3.). The construction of an instance BCS will be used in Construction 9.

Theorem 9. *There exists a feasible solution containing z for BCS in chordal graph iff X3C admits a positive solution and this solution is equal to* $6q$.

Proof. Consider Construction 9 in which the vertices z' and z'' are omitted. Assume that z is in a solution S of size strictly less than $6q$. The size of S is $6q - 2l$ with $l \geq 1$. So there are $3q - l$ blue (resp. red) vertices in S. Since the path between z and t_1 must be in the solution, there are at most $(q - l - 1)$ vertices among vertices-collection in S. Therefore at most $3(q - l - 1)$ red-vertices is covered by the previous vertices, impossible.

Construction 10 *Let (X, C) be a instance of X3C and $G = (V, E)$ the graph obtained by Construction 9 in which vertices z' and z'' are omitted. We generated an instance G' of BCS with ρ copies of G $((G_0, \ldots, G_\rho))$ and a path between the z-vertex of all copies.*

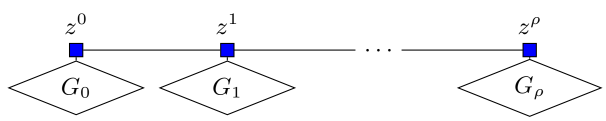


Fig. 11: Building an arbitrarily large instance of BCS for chordal graph from an instance of X3C.

This construction is illustrated by Figure 11.

Corollary 5. Balanced Connected Subgraph *problem cannot be approximated by any constant factor for chordal graphs unless* $\mathcal{P} = \mathcal{NP}$.

Proof. Based on Construction 10, clearly we have the two following cases:

– if there exists positive solution on a graph G_i, for a fixed i, the value of a solution is $6q$ with z in the solution, and the value on graph G' is obvious $6q\rho$.
– whereas in the negative case the value is at most or equal $6q$ but the vertex z cannot be in this solution, and the value of the solution on graph G' is also $6q$.

Thus the gap between yes/no-instance is ρ, which proves the inapproximability of BCS even if the input graph is a chordal graph.

5 Conclusion

In this article we pursued the classification of the BCS problem related to graph classes.

A further interesting question leads to the study of parameterized complexity for these problems. The parameter could be the difference between the number of red vertices and blue vertices. In this paper, we improved the complexity results for BCS. We gave a proof of $\mathcal{NP}$-completeness in bipartite cubic graphs, graphs of diameter three and bipartite graphs of diameter four. Our results nicely complement the ones of [3]. Indeed, they proved BCS to be polynomially solvable in graphs of diameter two and in graphs of maximum degree two.

Despite remaining computationally difficult in restrictive settings, BCS parameterized by the size of the solution belongs to $\mathcal{FPT}$. Indeed, as we stated in the introduction, BCS can be Turing-reduced to the graph motif problem. Since this problem parameterized by the size of the motif is $\mathcal{FPT}$, it implies that BCS parameterized by the size of the solution is $\mathcal{FPT}$as well. On the negative side, our results in graphs of bounded degree and bounded diameter imply that BCS parameterized by the diameter or the maximum degree of the input graph is in fact not $\mathcal{FPT}$.

References

1. Apollonio, N., Becker, R., Lari, I., Ricca, F., Simeone, B.: Bicolored graph partitioning, or: gerrymandering at its worst. Discrete Applied Mathematics **157**(17), 3601 – 3614 (2009), sixth International Conference on Graphs and Optimization 2007
2. Bhore, S., Jana, S., Pandit, S., Roy, S.: Balanced connected subgraph problem in geometric intersection graphs. In: Combinatorial Optimization and Applications - 13th International Conference, COCOA 2019, Xiamen, China, December 13-15, 2019, Proceedings. pp. 56–68 (2019)
3. Bhore, S., Chakraborty, S., Jana, S., Mitchell, J.S.B., Pandit, S., Roy, S.: The balanced connected subgraph problem. In: Algorithms and Discrete Applied Mathematics - 5th International Conference, CALDAM 2019, Kharagpur, India, February 14-16, 2019, Proceedings. pp. 201–215 (2019)
4. Borradaile, G., Klein, P., Mathieu, C.: An o(n log n) approximation scheme for steiner tree in planar graphs. ACM Trans. Algorithms **5**(3), 31:1–31:31 (Jul 2009)
5. Darties, B., R.Giroudeau, König, J.C., Pollet, V.: The balanced connected subgraph problem: Complexity results in bounded-degree and bounded-diameter graphs. In: Combinatorial Optimization and Applications - 13th International Conference, COCOA 2019, Xiamen, China, December 13-15, 2019, Proceedings. pp. 449–460 (2019)
6. Dondi, R., Fertin, G., Vialette, S.: Maximum motif problem in vertex-colored graphs. In: Combinatorial Pattern Matching. pp. 221–235. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
7. Fellows, M.R., Fertin, G., Hermelin, D., Vialette, S.: Sharp tractability borderlines for finding connected motifs in vertex-colored graphs. In: Automata, Languages and Programming, 34th International Colloquium, ICALP 2007, Wroclaw, Poland, July 9-13, 2007, Proceedings. pp. 340–351 (2007)

8. Fellows, M.R., Fertin, G., Hermelin, D., Vialette, S.: Upper and lower bounds for finding connected motifs in vertex-colored graphs. J. Comput. Syst. Sci. **77**(4), 799–811 (2011)
9. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York, NY, USA (1979)
10. Gonzalez, T.F.: Clustering to minimize the maximum intercluster distance. Theoretical Computer Science **38**, 293–306 (1985)
11. Johnson, D.S.: The np-completeness column: An ongoing guide. Journal of Algorithms **6**(1), 145 – 159 (1985)
12. Karp, R.: Reducibility among combinatorial problems. In: Complexity of Computer Computations. vol. 40, pp. 85–103 (01 1972)
13. Kobayashi, Y., Kojima, K., Matsubara, N., Sone, T., Yamamoto, A.: Algorithms and hardness results for the maximum balanced connected subgraph problem. In: Combinatorial Optimization and Applications - 13th International Conference, COCOA 2019, Xiamen, China, December 13-15, 2019, Proceedings. pp. 303–315 (2019)
14. 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.) Combinatorial Optimization and Applications. pp. 303–315. Springer International Publishing, Cham (2019)
15. Lacroix, V., Fernandes, C.G., Sagot, M.F.: Motif search in graphs: Application to metabolic networks. IEEE/ACM Trans. Comput. Biol. Bioinformatics **3**(4), 360–368 (Oct 2006)
16. Lokshtanov, D., Misra, N., Philip, G., Ramanujan, M.S., Saurabh, S.: Hardness of r-dominating set on graphs of diameter (r + 1). In: Parameterized and Exact Computation - 8th International Symposium, IPEC 2013, Sophia Antipolis, France, September 4-6, 2013, Revised Selected Papers. pp. 255–267 (2013)
17. Lokshtanov, D., Marx, D., Saurabh, S.: Lower bounds based on the exponential time hypothesis. Bulletin of the EATCS **105**, 41–72 (2011)

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Highlights

- we study Balanced Connected Subgraph Problem (BCS) , its weighted version (WBCS) and a related problem : Specified Red/Blue Cardinality Connected Subgraph (SCCS).
- we propose several new complexity and inapproximation results for these problems:
- BCS is NPC and noAPX on chordal graphs.
- BCS is NPX and noAPX on planar or bipartite graphs with maximum degree 4.
- BCS is also NPC on graphs with diameter 3 , on bipartite graphs with diameter 4, and on bipartite graphs with maximum degree 3.