(Strong) conflict-free connectivity: Algorithm and complexity

Article *in* Theoretical Computer Science · October 2019

DOI: 10.1016/j.tcs.2019.10.043

3 authors:

Meng Ji
Nankai University

Xueliang Li
Nankai University

Xiaoyu Zhu
Nankai University

Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

(Strong) conflict-free connectivity: Algorithm and complexity ☆

Meng Ji, Xueliang Li *, Xiaoyu Zhu

Center for Combinatorics and LPMC, Nankai University, Tianjin 300071, China

A R T I C L E I N F O

Article history:
Received 11 June 2019
Received in revised form 30 August 2019
Accepted 29 October 2019
Available online xxxx
Communicated by D.-Z. Du

Keywords:
Conflict-free connection
Polynomial-time algorithm
Strong conflict-free connection
Complexity

A B S T R A C T

Let G be an(a) edge(vertex)-colored graph. A path P of G is called a *conflict-free path* if there is a color that is used on exactly one of the edges(vertices) of P. The graph G is called *conflict-free (vertex-)connected* if any two distinct vertices of G are connected by a conflict-free path, whereas the graph G is called *strongly conflict-free connected* if any two distinct vertices u, v of G are connected by a conflict-free path of length of a shortest path between u and v in G. For a connected graph G, the *(strong) conflict-free connection number* of G, denoted by $(scfc(G))$ $cfc(G)$, is defined as the smallest number of colors that are required to make G (strongly) conflict-free connected. In this paper, we first investigate the question: Given a connected graph G and a coloring $c : E(or\ V) \to \{1, 2, \cdots, k\}$ $(k \geq 1)$ of G, determine whether or not G is, respectively, conflict-free connected, conflict-free vertex-connected, strongly conflict-free connected under the coloring c. We solve this question by providing polynomial-time algorithms. We then show that the problem of deciding whether $scfc(G) \leq k$ $(k \geq 2)$ for a given graph G is NP-complete. Moreover, we prove that it is NP-complete to decide whether there is a k-edge-coloring $(k \geq 2)$ of G such that all pairs $(u, v) \in P$ $(P \subset V \times V)$ are strongly conflict-free connected.

1. Introduction

All graphs mentioned in this paper are simple, undirected and finite. We follow book [4] for undefined notation and terminology. Even et al. in [14] introduced the hypergraph version of this concept of conflict-free coloring. The coloring was motivated to solve the problem of assigning frequencies to different base stations in cellular networks. There are a number of base stations and clients in the network. Each base station is a vertex in the hypergraph which needs to be allocated to a frequency. Different frequencies stand for different colors in a vertex-colored hypergraph. Every client is moveable, so it can be in the range of lots of base stations. Thus each client is a set of many vertices, i.e., clients represent edges. For each client, in order to make connection with one of the base station in the range, there must be at least one base station with a unique frequency in the range for fear of interference. Unnecessarily many different frequencies can be expensive, so this situation may be converted to a conflict-free vertex-coloring problem of a hypergraph seeking for the minimum number of colors which is defined as the *conflict-free chromatic number* of the hypergraph. More information for the conflict-free coloring can be seen from the survey paper [28]. Later on, Czap et al. [12] introduced the concept of *conflict-free connection* of graphs on the basis of the earlier hypergraph version. An edge-colored graph G is called *conflict-free connected* if any two

☆ Supported by NSFC No. 11871034, 11531011 and NSFQH No. 2017-ZJ-790.
* Corresponding author.
E-mail addresses: jimengecho@163.com (M. Ji), lxl@nankai.edu.cn (X. Li), zhuxy@mail.nankai.edu.cn (X. Zhu).

https://doi.org/10.1016/j.tcs.2019.10.043
0304-3975/

of its vertices are connected by a path which contains a color used on exactly one of the edges of the path, where the path is called *a conflict-free path*. The minimum number of colors required to make G conflict-free connected is called the *conflict-free connection number* of G, denoted by *cfc*(G). Czap et al. [12] showed that $cfc(G) \leq \chi_r'(G)$ for a connected graph G, where $\chi_r'(G)$ is called *edge ranking number*. [A vertex (edge) coloring of a graph $G = (V, E)$ is a vertex (edge) *t-ranking* if, for any two vertices (edges) of a same color i, every path between them contains a vertex (edge) of color j larger than i. The vertex ranking number $\chi_r(G)$ (edge ranking number $\chi_r'(G)$) is the smallest value of t such that G has a vertex (edge) t-ranking.] Furthermore, Chang et al. [10] showed that if T is a cfc-critical tree, then $cfc(T) = \chi_r'(T)$. Lam [16] proved that the edge ranking problem of simple graphs is NP-complete. Llewellyn et al. [25] and Pothen [27] independently proved that finding an optimal vertex ranking of a graph is NP-hard. For more information on ranking number, we refer to [3,15]. As a natural counterpart of the conflict-free connection, Li et al. in [23] introduced the concept of *conflict-free vertex-connection* of graphs. An vertex-colored graph G is called *conflict-free connected* if any two of its vertices are connected by a path which contains a color used on exactly one of its vertices, where the path is called *a conflict-free path*. The minimum number of colors required to make G conflict-free vertex-connected is called the *conflict-free vertex-connection number* of G, denoted by *vcfc*(G).

A path in a vertex-colored graph is called a *conflict-free path* if it has at least one vertex with a unique color on the path. A vertex-colored graph is called *conflict-free vertex-connected* if there is a conflict-free path between every pair of distinct vertices of G. For a connected graph G, the minimum number of colors required to make G conflict-free vertex-connected is called the *conflict-free vertex-connection number* of G, denoted by *vcfc*(G). There have been many results on the conflict-free (vertex-)connection coloring due to its theoretical and practical significance, such as [8–10,12,13,23,24].

The conflict-free connection number of graphs without cut-edges has been obtained in [12,13]. Thus determining the value of $cfc(G)$ for graphs G with cut-edges becomes the main task. Trees are extremal such graphs for which every edge is a cut-edge. For a tree T we can build a hypergraph H as follows. The hypergraph $H_{EP}(T) = (\mathcal{V}, \mathcal{E})$ has $\mathcal{V}(H_{EP}) = E(T)$ and $\mathcal{E}(H_{EP}) = \{E(P) |\ P$ is a path of $T\}$. One can easily see that the conflict-free chromatic number of the hypergraph H is just the conflict-free connection number of T. For more results we refer to [8–10,13]. Nevertheless, most of them are about the graph structural characterizations. The graph structural analytic method may be more useful to handle graphs with certain characterizations such as the 2-edge-connected graph and some given graph classes. But a polynomial-time algorithm is applicable to all general graphs. However, very few results on this have been obtained for now. Thus we address the computational aspects of the (strong) conflict-free (vertex-)connection colorings in this paper:

Definition 1.1. An edge-colored graph G is called *strongly conflict-free connected* if any two distinct vertices u, v of G are connected by a conflict-free path of length of a shortest path between u and v in G, where the path is called a strong conflict-free path. For a connected graph G, the *strong conflict-free connection number* of G, denoted by $scfc(G)$, is defined as the smallest number of colors that are required to make G strongly conflict-free connected.

Combining the concepts of *rainbow connection number* (Chartrand et al. in [11]), *proper connection number* (Andrews et al. in [2] and Borozan et al. in [5]) and *monochromatic connection number* (Caro and Yuster in [6]), it is natural to ask such a question. (For details on the concepts, one can refer to some books [18,20] and survey papers [17,19,21,22].)

Problem 1. Given an integer $k \geq 1$ and a connected graph G, is it NP-hard or polynomial-time solvable to answer each of the following questions?

(a) Is $rc(G) \leq k$?
(b) Is $pc(G) \leq k$?
(c) Is $mc(G) \geq k$?
(d) Is $cfc(G) \leq k$? (Is $vcfc(G) \leq k$? for the vertex version)
(e) Is $scfc(G) \leq k$? (can be also referred to as the **k-strong conflict-free connectivity problem** in the following context).

For general graphs, Ananth et al. proved in [1] that Question (a) is NP-hard. Chakraborty et al. proved in [7] that Question (a) is NP-complete even if $k = 2$. The answers for Questions $(b), (c)$, (d) and (e) remain unknown. For a tree T, Question (a) is easy since $rc(T) = n - 1$, and Question (b) is also easy since $pc(T) = \Delta(T)$, where n is the order of T and $\Delta(T)$ is the maximum degree of T. However, the complexity for Question (d) is unknown even if G is a tree T.

Actually, Problem 1 is equivalent to the following statement:

Problem 2. Given an integer $k \geq 1$ and a connected graph G, determine whether there is a k-edge (or vertex)-coloring to make G

(a) rainbow connected.
(b) proper connected.
(c) monochromatically connected.

(d) conflict-free connected (or conflict-free vertex-connected).
(e) strongly conflict-free connected.

The following is a weaker version for Problem 1:

Problem 3. Given a connected graph G with n vertices and m edges and a coloring $c : E$ (or $V) \to \{1, 2, \cdots, k\}$ $(k \geq 1)$ of the graph, for each pair of distinct vertices (u, v) of G, determine whether there is a path P between u, v such that
(a) P is a rainbow path.
(b) P is a proper path.
(c) P is a monochromatic path.
(d) P is a conflict-free path (or vertex conflict-free path).
(e) P is a strong conflict-free path.

For general graphs, Chakraborty et al. proved in [7] that Question (a) is NP-complete. Recently, Ozeki [26] confirmed that Question (b) is polynomial-time solvable. It is not difficult to see that Question (c) can also be solved in polynomial-time, just by checking all subgraphs each being induced by the set of edges with a same color.

The paper is arranged as follows: Next section, we will provide two polynomial-time algorithms for Problem 3 (d) and Problem 3 (e). In section 3, we present the complexity result for the strong conflict-free connection problem by proving that it is NP-complete to answer Problem 4 for $k \geq 2$ and Problem 1(e) for $k \geq 2$.

2. Polynomial-time algorithms

Before presenting our main theorem for Question (d) in Problem 3, some auxiliary lemmas are needed.

Lemma 2.1. [12] *Let u, v be distinct vertices and $e = xy$ be an edge of a 2-connected graph G. Then there is a u-v-path in G containing the edge e.*

Let x be a vertex and Y be a set of vertices of a connected graph G. Then a family of k internally disjoint (x, Y)-paths whose terminal vertices are pairwise distinct is referred to as a k-fan from x to Y.

The following is a famous Fan Lemma.

Lemma 2.2. *Let G be a k-connected graph, x be a vertex of G, and $Y \subseteq V \backslash \{x\}$ be a set of at least k vertices of G. Then there exists a k-fan in G from x to Y.*

For a connected graph G, a vertex of G is called *a separating vertex* if its removal splits the graph into at least two nonempty connected components. We call the graph *nonseparable* if it is connected without separating vertices. A *block* of the graph is a subgraph which is nonseparable and maximal in this property. We can construct a bipartite graph $B(G)$ for every connected graph G as follows: let $V(B(G)) = (\mathcal{B}, S)$ where $\mathcal{B}$ represents the set of all blocks in G and S is the set of separating vertices. A block $B \in \mathcal{B}$ and a vertex $s \in S$ are adjacent if and only if $s \in B$ in G. It is clear that $B(G)$ is also a tree, and we call it the *block tree* of G.

Lemma 2.3. *For a connected graph G, let $u, v \in V(G)$, $st \in E(G)$. Then there is no u-v-path containing edge st if and only if there exists a vertex z such that neither u nor v is connected to s or t in the graph $G - z$.*

Proof. For sufficiency, suppose there exists a u-v-path containing st. Then obviously z must appear at least twice on this path, a contradiction.

For necessity, we claim that G is not 2-connected since otherwise Lemma 2.1 will lead to a contradiction.

Assume that $st \in B_1$, $u \in B_2$ and $v \in B_3$ where B_i $(i = 1, 2, 3)$ are blocks of G. Then $B_1 = B_2 = B_3$ cannot happen since otherwise a u-v-path containing st can be found according to Lemma 2.1, a contradiction. If $B_2 = B_3$, then the removal of any separating vertex on the path of $B(G)$ between B_1 and B_2 will leave neither u nor v connected to s or t. Consider the case that $B_2 \neq B_3$. We claim that B_1 is not on the path between B_2 and B_3 in $B(G)$, since otherwise a u-v-path can be chosen to go through st by applying Lemma 2.1 to B_1, also a contradiction. At last, we consider the deletion of the first separating vertex on the path of $B(G)$ from B_1 to B_2, and this will cause the disconnections we want. □

With a similar proof, one can get the corresponding lemma for the vertex version.

Lemma 2.4. *For a connected graph G, let $u, v, w \in V(G)$. Then there is no u-v-path containing vertex w if and only if there exists a vertex $z \neq w$ such that neither u nor v is connected to w in the graph $G - z$.*

The famous Depth-First Search (DFS) (see [4]) will be used in our algorithm. For a graph G with n vertices and m edges, the DFS starts from a root vertex x and goes as far as it can along a path, after that, it backtracks until finding a new path and then explores it. The algorithm stops when all vertices of G have been explored. As is well known, the time complexity for DFS is $\mathcal{O}(n+m)$.

Theorem 2.5. *There exists a polynomial-time algorithm to determine Question* (*d*) *in* Problem 3. *The complexity for the edge version is at most $\mathcal{O}(n^3m^2)$, and the complexity for the vertex version is at most $\mathcal{O}(n^4m)$.*

Proof of the edge version. Given $k \geq 1$ and a connected graph G with an edge-coloring $c : E(G) \to \{1, 2, \cdots, k\}$, let E_i ($i = 1, 2, \cdots, k$) be the edge set containing all edges colored with i. We present our algorithm below:

Algorithm 1: Determining whether an edge-colored graph is conflict-free connected.

Input: A given integer $k \geq 1$, a connected graph G with n vertices, m edges and an edge-coloring $c : E(G) \to \{1, 2, \cdots, k\}$.
Output: Whether G is conflict-free connected or not.
1: Check if there is an unselected pair of distinct vertices in G. If so, pick one pair (u, v), go to **2**; otherwise, go to **8**.
2: Set $i = 0$, go to **3**.
3: Check if $i \leq k-1$. If so, $i := i+1$, $G' := G - E_i$, go to **4**; otherwise, go to **9**.
4: For (u, v), determine if there is an unselected edge e in E_i. If so, pick $e = st$, set $G'' := G' + e$, go to **5**; otherwise, go to **3**.
5: Check if u, v and st are connected in G''. If so, go to **6**; otherwise, go to **4**.
6: For (u, v) and st, determine if there is an unselected vertex in G''. If so, pick one vertex z, go to **7**; otherwise, go to **1**.
7: Determine if neither u nor v is connected to s or t in $G'' - z$. If so, go to **4**; otherwise, go to **6**.
8: Return: G is conflict-free connected under coloring c.
9: Return: G is not conflict-free connected under coloring c.

Let us first prove the algorithm above is correct. If for a pair of distinct vertices (u, v), there is no conflict-free path between them, then for any edge e in G, there is no u-v-path in $G - E_{c(e)} + e$ containing e. Thus according to Lemma 2.3, for each e, there must be a vertex z (step **6**) such that neither u nor v is connected to s or t in $G'' - z = G - E_{c(e)} + e - z$. As a result, after traversing every edge in G, it comes to step **4**, then step **3** and finally step **9** obtaining the right result that G is not conflict-free connected.

If for u and v, there is a conflict-path between them, then there must exist an edge e such that for any vertex z in G, either u or v is connected to s or t in $G'' - z = G - E_{c(e)} + e - z$. Therefore, after repeating steps **7** and **6** for some times, the running process will come to step **1** and then examine the next pair of vertices. If all pairs of vertices have been examined, it will announce that G is conflict-free connected. This shows the correctness of our algorithm.

For a fixed pair of vertices u and v and a fixed edge $e = st$, to examine step **5**, we only need to apply the DFS algorithm appointing s as the root vertex. Then for any vertex z of G, again apply the DFS algorithm to step **7**. Consequently we get that the complexity is $\mathcal{O}((n+m)n + n + m) = \mathcal{O}(nm)$. Since there are $\mathcal{O}(n^2)$ pair of vertices and m edges in G, the overall complexity is at most $\mathcal{O}(n^3m^2)$.

Proof of the vertex version. With Fan Lemma and Lemma 2.4, it actually has analogous analysis with the edge version. The differences are as follows: (*i*) V_i $(1 \leq i \leq k)$ shall take the place of E_i $(1 \leq i \leq k)$, and (*ii*) we will pick a vertex this time instead of an edge in step **4**. Because of this, an m will be replaced by an n in the complexity for the edge version, so the time complexity for the vertex version is $\mathcal{O}(n^4m)$.

Besides, for a picked pair of vertices u and v, if $c(u) = c(v)$, then the vertex set $V_{c(u)}$ is not needed to consider in step **3** since $c(u)$ can never be the unique color on any u-v-path; if $c(u) \neq c(v)$, any vertex of $(V_{c(u)} \backslash u)$ (or $(V_{c(v)} \backslash v)$) is not needed to add back after removing $(V_{c(u)} \backslash u)$ (or $(V_{c(v)} \backslash v)$) from G (like in step **4**) because the unique color has already exists on u (or v). This saves some operations compared to the algorithm for the edge version. Thus the complexity for the vertex version is at most $\mathcal{O}(n^4m)$. □

For Question (*e*) in Problem 3, we also get a polynomial-time algorithm in which the Breadth-First Search (BFS) is used. For a graph G with n vertices and m edges, the BFS starts from a root vertex x and explores all the neighbors of the vertices at the present level before moving to the next depth level. The algorithm stops when all vertices of G have been explored. As is well known, the time complexity for BFS is $\mathcal{O}(n+m)$.

Before presenting our algorithm, we give a necessary definition.

Definition 2.6. For a vertex u in a connected graph G, it is obvious that any edge $e = st$ must have $|d_G(u,s) - d_G(u,t)| \leq 1$. So, e is called a *vertical edge* of u if $|d_G(u,s) - d_G(u,t)| = 1$ and a *horizontal edge* of u otherwise.

Theorem 2.7. *There exists a polynomial-time algorithm to determine Question* (*e*) *in* Problem 3. *The complexity is at most $\mathcal{O}(n^2m^2)$.*

Proof. Given $k \geq 1$ and a connected graph G with an edge-coloring $c : E(G) \to \{1, 2, \cdots, k\}$, let E_i ($i = 1, 2, \cdots, k$) be the edge set containing all edges colored with i. We present our algorithm below:

Algorithm 2: Determining whether an edge-colored graph is strongly conflict-free connected.

Input: A given integer $k \geq 1$, a connected graph G with n vertices, m edges and an edge-coloring $c: E(G) \to \{1, 2, \cdots, k\}$.
Output: Whether G is strongly conflict-free connected or not.
1: Check if there is an unselected pair of distinct vertices in G. If so, pick one pair (u, v), go to **2**; otherwise, go to **6**.
2: Set $i = 0$, go to **3**.
3: Check if $i \leq k-1$. If so, $i := i+1$, $G' := G - E_i$, go to **4**; otherwise, go to **7**.
4: For (u, v), determine if there is an unselected vertical edge $e = st$ with $d_G(u, s) < d_G(u, t) \leq d_G(u, v)$ in E_i. If so, set $G'' := G' + e$, go to **5**; otherwise, go to **3**.
5: Check if $d_G(u, s) = d_{G''}(u, s)$ and $d_{G''}(v, t) = d_G(u, v) - d_G(u, t)$. If so, go to **1**; otherwise, go to **4**.
6: Return: G is strongly conflict-free connected under coloring c.
7: Return: G is not strongly conflict-free connected under coloring c.

We will prove that the algorithm above is correct. If for a pair of distinct vertices u and v, there is no strong conflict-free path between them, then for any vertical edge $e = st$ with $d_G(u, s) < d_G(u, t) \leq d_G(u, v)$ in G, any u-v-path in $G - E_{c(e)} + e$ containing e has length greater than $d_G(u, v)$. Hence there must be $d_G(u, s) \neq d_{G''}(u, s)$ or $d_{G''}(v, t) \neq d_G(u, v) - d_G(u, t)$ in step **5**. As a result, after traversing every vertical edge $e = st$ with $d_G(u, s) < d_G(u, t) \leq d_G(u, v)$ in G, it comes to step **4**, then step **3** and finally step **7** obtaining the right result that G is not strongly conflict-free connected.

If for a pair of vertices u and v, there is a strong conflict-free path between them, then there must exist a vertical edge $e = st$ with $d_G(u, s) < d_G(u, t) \leq d_G(u, v)$ in G such that we can obtain a u-v-path in $G - E_{c(e)} + e$ containing e whose length is equal to $d_G(u, v)$. Then there must be $d_G(u, s) = d_{G''}(u, s)$ and $d_{G''}(v, t) = d_G(u, v) - d_G(u, t)$. Therefore, the running process will come to step **1** after step **5** and then examine the next pair of vertices. If all pairs of vertices have been examined, it will announce that G is strongly conflict-free connected. This shows the correctness of our algorithm.

For a fixed pair of vertices u and v, firstly we need to apply the BFS algorithm to G designating u as the root to acquire all vertical edge $e = st$ with $d_G(u, s) < d_G(u, t) \leq d_G(u, v)$ in G. Then for any fixed edge $e = st$, we only need to apply the BFS algorithm a few more times to G' to examine step **5**. Consequently we get that the complexity is $\mathcal{O}(n+m+m(n+m)) = \mathcal{O}(m^2)$. Since there are $\mathcal{O}(n^2)$ pairs of vertices in G, the overall complexity is at most $\mathcal{O}(n^2m^2)$. □

If one wants to determine whether an edge-colored graph is k-subset strongly conflict-free connected, one only needs to examine all pairs of vertices in P instead of those in $V \times V$ in Algorithm 2. Then we immediately have the following theorem:

Theorem 2.8. *There exists a polynomial-time algorithm to determine whether an edge-colored graph is k-subset strongly conflict-free connected.*

3. Hardness results on strong conflict-free connectivity problem

In this section, we will first prove the main result that Problem 1(e) is NP-complete in subsection 3.1, and then we derive the result that Problem 4 for $k \geq 2$ is NP-complete in subsection 3.2. Before proving the main result, we first define the following three useful problems.

Problem 4 *(k-subset strong conflict-free connectivity problem).* Given a graph G and a set $P \subset V \times V$, deciding whether there is an edge-coloring of G with k colors such that all pairs $(u, v) \in P$ are strongly conflict-free connected.

Problem 5 *(Partial 2-edge-coloring problem).* Given a graph $G = (V, E)$ and a partial 2-edge-coloring $\hat{c}: \hat{E} \to \{0, 1\}$ for $\hat{E} \subset E$, deciding whether $\hat{c}$ can be extended to a complete 2-edge-coloring $c: E \to \{0, 1\}$ that makes G strongly conflict-free connected.

Problem 6 *(k-vertex-coloring problem).* Given a graph $G = (V, E)$ and a fixed integer k, deciding whether there is a k-vertex-coloring for G such that each color class is an independent set.

3.1. k-strong conflict-free connectivity problem

Theorem 3.1 is our main result. In the following we first prove Theorem 3.1 for $k = 2$ and then for $k \geq 3$.

Theorem 3.1. *For $k \geq 2$,* Problem 1 *(e) is NP-complete.*

At first we deal with the case $k = 2$, Chakraborty et al. obtained the following result (Theorem 1.1 in [7]).

Lemma 3.2. [7] *Given a graph G, deciding if $rc(G) = 2$ is NP-complete. In particular, computing $rc(G)$ is NP-hard.*

Then we can easily get the following result by the definitions of the rainbow connection and strong conflict-free connection.

Lemma 3.3. *Given a graph $G = (V, E)$, $rc(G) = 2$ if and only if $diam(G) = 2$ and $scfc(G) = 2$.*

Lemma 3.4. *For $k = 2$, Problem 1 (e) is NP-complete.*

Proof. It is NP-complete to decide whether the rainbow connection number of a connected graph is 2 by Lemma 3.2. Therefore, deciding whether $scfc(G) = 2$ and $diam(G) = 2$ is NP-complete by Lemma 3.3. Since it is easy to see that deciding if $diam(G) = 2$ can be done in polynomial-time, then deciding if $scfc(G) = 2$ must be NP-complete. □

Lemma 3.5. *For $k \geq 3$, Problem 6 $\preceq$ Problem 4.*

Proof. Now we polynomially construct a graph $G' = (V', E')$: for a given connected graph $G = (V, E)$, let $V' = V \cup \{x\}$, $E' = \{vx : v \in V\}$, and $P = \{(u, v) : uv \in E\}$. It remains to prove that the graph $G = (V, E)$ is vertex-colorable with $k \geq 3$ colors if and only if the graph $G' = (V', E')$ can be k-edge-colored such that there is a strong conflict-free path between every pair $(u, v) \in P$.

Assume that G can be vertex-colored with k colors. We prove that there is an assignment of k colors to the edges of the graph G' that enables a strong conflict-free path between every pair $(u, v) \in P$. We define a bijection between V and E': $v \in V \rightarrow vx \in E'$. If i is a color assigned to a vertex $v \in V$, then we assign the color i to the edge $xv \in E'$. For any pair $(u, v) \in P$, xu and xv have different colors since $uv \in E$. Thus, the unique path $u - x - v$ is a strong conflict-free path between u and v. The other direction can be also easily verified according to the bijection above. □

Now we are at the point to give the proof of our main result Theorem 3.1.

Proof of Theorem 3.1. For $k = 2$, it holds by Lemma 3.4. Then for $k \geq 3$, considering Theorem 2.7 and Lemma 3.5, to prove Theorem 3.1, we only need to reduce the instances obtained from the proof of Lemma 3.5 to some instances of Problem 1(e). Let $G = (V, E)$ be a star graph with $\hat{V} = \{v_1, v_2, \cdots, v_n\}$ being the leaf vertex set and a being the central vertex. The vertices of any pair $(v_i, v_j) \in P$ are both leaf vertices of G. Then we construct a graph G' according to G as follows: for every vertex $v_i \in \hat{V}$, we introduce two new vertices x_{v_i} and x'_{v_i}, and for every pair of leaf vertices $(u, v) \in (\hat{V} \times \hat{V}) \setminus P$ we introduce two new vertices $x_{(u,v)}, x'_{(u,v)}$. Then we have:

$$V' = V \cup V_1 \cup V_2 \text{ where}$$
$$V_1 = \{x_{v_i} : i \in \{1, \cdots, n\}\} \cup \{x_{(v_i,v_j)} : (v_i, v_j) \in (\hat{V} \times \hat{V}) \setminus P\}$$
$$V_2 = \{x'_{v_i} : i \in \{1, \cdots, n\}\} \cup \{x'_{(v_i,v_j)} : (v_i, v_j) \in (\hat{V} \times \hat{V}) \setminus P\}$$
$$E' = E \cup E_1 \cup E_2 \cup E_3 \cup E_4 \text{ where}$$
$$E_1 = \{v_i x_{v_i} : v_i \in \hat{V}, x_{v_i} \in V_1\}$$
$$E_2 = \{v_i x_{(v_i,v_j)}, v_j x_{(v_i,v_j)} : (v_i, v_j) \in (\hat{V} \times \hat{V}) \setminus P\}$$
$$E_3 = \{xx' : x \in V_1, x' \in V_2\}$$
$$E_4 = \{ax' : x' \in V_2\}$$

Now we need to prove that G' is k-strongly conflict-free connected if and only if G is k-subset strongly conflict-free connected.

First, there is a two-length path $v_i - a - v_j$ in G for any pair $(v_i, v_j) \in P$, and this path also occurred in G' which is the unique path of length two in G' between v_i and v_j. It implies that if the graph G' is strongly conflict-free colored with k colors, then G has an edge-coloring with k colors such that every pair in P is strongly conflict-free connected.

Second, assume that there is a k-edge-coloring c of G using colors from $\{1, 2, \cdots, k\}$ such that all pairs in P are strongly conflict-free connected. Then we extend this edge-coloring c of G to a k-edge-coloring c' of G': E retains the coloring c; assign the color 3 to $uv \in E_1$; assign the colors 1 and 2 to $v_i x_{(v_i,v_j)}$ and $v_j x_{(v_i,v_j)} \in E_2$ respectively. Since the subgraph $H = (V_1 \cup V_2, E_3)$ is a complete bipartite graph, we choose a perfect matching M of size $|V_1|$, and assign the edges in M with color 1 and assign the edges in $E_3 \setminus M$ with color 2. We then assign to the edges $ax' \in E_4$ color 3. It is easy to verify that this coloring makes G' strongly conflict-free connected. Since the graph G' is bipartite, the k-strong conflict-free connectivity problem is NP-complete even for bipartite graphs. □

Please cite this article in press as: M. Ji et al., (Strong) conflict-free connectivity: Algorithm and complexity, Theoret. Comput. Sci. (2019), https://doi.org/10.1016/j.tcs.2019.10.043

3.2. k-subset strong conflict-free connectivity problem

Now we will show the following Theorem 3.6 in this subsection.

Theorem 3.6. *For $k \geq 2$,* Problem 4 *is NP-complete.*

In the following process, we first show Problem 5 can be reduced to Problem 4 and then 3-SAT can reduced to Problem 5.

Lemma 3.7. *For $k = 2$,* Problem 5 $\preceq$ Problem 4.

Proof. Given such a partial coloring $\hat{c}$ for $\hat{E} \subset E$, let $\hat{E} = \hat{E}_1 \cup \hat{E}_2$, where $\hat{E}_1$ contains all edges in $\hat{E}$ colored with 0 and $\hat{E}_2 = \hat{E} \setminus \hat{E}_1$. We then extend the original graph $G = (V, E)$ to a graph $G' = (V', E')$, and define a set P of pairs of vertices of V' such that the answer for Problem 5 with G and $\hat{c}$ as parameters is yes if and only if the answer for Problem 4 with G' and P as parameters is yes.

Let $[n]$ $(n = |V|)$ be an arbitrary linear ordering of the vertices and $l(v)$ $(v \in V)$ be the number related to v in the ordering. Let θ: $E \to V$ be a mapping that maps an edge $e = uv$ to u if $l(u) > l(v)$, and to v otherwise. On the contrary, let ε: $E \to V$ be a mapping that maps $e = uv$ to u if $l(u) < l(v)$, and to v otherwise. Let $r = \lceil \frac{n}{2} \rceil$ if $\lceil \frac{n}{2} \rceil$ is odd, and $r = \lceil \frac{n}{2} \rceil + 1$ otherwise. We polynomially construct G' as follows: the vertex set is

$$V' = V \cup V_1 \cup V_2 \cup V_3 \text{ where}$$
$$V_1 = \{b_1, c, b_2\}$$
$$V_2 = \{c_e : \text{ for } \forall e \in (\hat{E}_1 \cup \hat{E}_2)\}$$
$$V_3 = \{t_1^e, t_2^e, \cdots, t_r^e : \text{for } \forall e \in (\hat{E}_1 \cup \hat{E}_2)\}$$

and the edge set is

$$E' = E \cup E_1 \cup E_2 \cup E_3 \text{ where}$$
$$E_1 = \{b_1 c, b_2 c\}$$
$$E_2 = \{b_i t_1^e, t_1^e t_2^e, \cdots, t_{r-1}^e t_r^e, t_r^e c_e : i \in \{1, 2\}, e \in \hat{E}_i\}$$
$$E_3 = \{c_e \varepsilon(e) : e \in (\hat{E}_1 \cup \hat{E}_2)\}$$

Now we define the set P of pairs of vertices of V':

$$P = \{b_1, b_2\} \cup \{\{u, v\} : u, v \in V, u \neq v\} \cup \{\{c, t_1^e\}, \{b_i, t_2^e\}, \{t_1^e, t_3^e\}, \{t_2^e, t_4^e\}, \cdots,$$
$$\{t_{r-2}^e t_r^e\}, \{t_{r-1}^e, c_e\}, \{t_r^e, \varepsilon(e)\} : i \in \{1, 2\}, e \in \hat{E}_i\} \cup \{\{c_e, \theta(e)\} : e \in (\hat{E}_1 \cup \hat{E}_2)\}$$

Now, if there is a strong conflict-free coloring with two colors $\pi_c = (E_1, E_2)$ of G which extends $\pi_{\hat{c}} = (\hat{E}_1, \hat{E}_2)$, then we color G' as follows. Every edge $e \in E$ retains the coloring c: the edge is colored with 0 if it is in E_1 and otherwise it is colored with 1. The edges $b_1 c, \varepsilon(e) c_e$ for $e \in \hat{E}_2$ are all colored with 0, $b_2 c$ and $c_e \varepsilon(e)$ for $e \in \hat{E}_1$ are all colored with 1. Moreover, the edges $b_1 t_1^e, t_1^e t_2^e, \cdots, t_{r-1}^e t_r^e, t_r^e c_e$ $(e \in \hat{E}_1)$ are assigned to the colors 1 and 0 alternately and the edges $b_2 t_1^e, t_1^e t_2^e, \cdots, t_{r-1}^e t_r^e, t_r^e c_e$ $(e \in \hat{E}_2)$ are assigned to the colors 0 and 1 alternately. One can see that this coloring indeed makes each pair in P strongly conflict-free connected.

On the other direction, we can see that P contains all vertex pairs of G and for each of such pair u and v, all the shortest paths between u and v in G' are completely contained in G. Thus any 2-edge-coloring of G' that makes the pairs in P strongly conflict-free connected clearly contains a strong conflict-free coloring of G. Also, such a coloring would have to color cb_1 and cb_2 differently. It would also have to give every $b_1 t_1^e (e \in \hat{E}_1)(b_2 t_1^e (e \in \hat{E}_2))$ a color different from that of $cb_1(cb_2)$. By further reasoning, we can see that the colors used on $b_1 t_1^e, t_1^e t_2^e, \cdots, t_{r-1}^e t_r^e, t_r^e c_e$ $(e \in \hat{E}_1)$ and $b_2 t_1^e, t_1^e t_2^e, \cdots, t_{r-1}^e t_r^e, t_r^e c_e$ $(e \in \hat{E}_2)$ are both alternately. As a result, $c_e \varepsilon(e) (e \in \hat{E}_1)$ must be in a color different from that of cb_1, and $c_e \varepsilon(e) (e \in \hat{E}_2)$ is in a color different from that of cb_2. Finally, every $e \in \hat{E}_i$ must be assigned with the color identical to that of cb_i to make $\theta(e)$ and c_e strongly conflict-free connected. Without loss of generality, we suppose that the edge cb_1 is colored with 0. It is clear that this coloring of G' conforms to the original partial coloring $\hat{c}$. This implies that $\hat{c}$ can be extended to a complete 2-edge-coloring c: $E \to \{0, 1\}$ that makes G strongly conflict-free connected. $\square$

Lemma 3.8. *3-SAT* $\preceq$ Problem 5.

Proof. Let $\phi := \bigwedge_{i=1}^{l} c_i$ be a 3-conjunctive normal form formula over variables $\{x_1, x_2, \cdots, x_n\}$. Then we polynomially construct a graph $G' = (V', E')$ as follows:

$$V' = \{c_i : i \in [l]\} \cup \{x_i : i \in [n]\} \cup \{a\}$$
$$E' = \{x_i c_j : x_i \in c_j\} \cup \{x_i a : i \in [n]\} \cup \{c_i c_j : i, j \in [l]\} \cup \{x_i x_j : i, j \in [n]\}$$

Now we define the partial 2-edge-coloring c': edges $\{c_i c_j : i, j \in [l]\}$ and $\{x_i x_j : i, j \in [n]\}$ are assigned the color 0; the edge $x_i c_j \in E'$ is assigned the color 0 if x_i is positive in c_j and color 1 otherwise. Thus only the edges in $\{x_i a : i \in [n]\}$ are left uncolored.

Without loss of generality, assume that all variables in ϕ appear both as positive and as negative, so it only remains to prove that there is an extension c of c' that enables a strong conflict-free path between a and each c_i $(i \in [l])$ if and only if ϕ is satisfiable since there will always be a strong conflict-free path between any other pair of vertices of V' whatever the extension is. Let $c(x_i a) = x_i$ $(i \in [n])$. One can verify that this relationship does hold. In fact, in a successful extension c of c', the color vector formed by $c(x_i a)$ $(i \in [n])$ can be seen as a solution vector of ϕ, and vice versa. □

Proof of Theorem 3.6. First, Theorem 2.8 implies that Problem 4 belongs to NP. Then, for $k = 2$, we reduce Problem 5 to Problem 4 by Lemma 3.7, and then reduce 3-SAT to Problem 5 by Lemma 3.8. Clearly, Problem 4 is NP-complete for $k = 2$. For $k \geq 3$, it also holds by Lemma 3.5. The proof of Theorem 3.6 is completed. □

4. Concluding remarks

In this paper, we mainly study the computational complexity of (strong) conflict-free (vertex-)connection numbers of graphs. We show that there exist polynomial-time algorithms to check whether a given (vertex)edge-colored graph is (strongly) conflict-free (vertex-)connected. We then show that given a connected graph G and an integer $k \geq 2$, it is NP-complete to determine whether there is a k-edge-coloring to make G strongly conflict-free connected. For further study, we propose the following open problem: For a tree T and an integer $k \geq 2$, is it NP-complete to determine whether there is a k-edge-coloring to make T conflict-free connected?

Declaration of competing interest

There is no conflict of interest.

Acknowledgement

The authors are grateful to the reviewers for the valuable suggestions and comments, which helped to improve the presentation of the paper.

References

[1] P. Ananth, M. Nasré, New hardness results in rainbow connectivity, arXiv:1104.2074 [cs.CC], 2011.
[2] E. Andrews, E. Laforge, C. Lumduanhom, P. Zhang, On proper-path colorings in graphs, J. Comb. Math. Comb. Comput. 97 (2016) 189–207.
[3] H.L. Bodlaender, J.S. Deogun, K. Jansen, T. Kloks, D. Kratsch, H. Müller, Zs. Tuza, Rankings of graphs, SIAM J. Discrete Math. 11 (1) (1998) 168–181.
[4] J.A. Bondy, U.S.R. Murty, Graph Theory, GTM, vol. 244, Springer, 2008.
[5] V. Borozan, S. Fujita, A. Gerek, C. Magnant, Y. Manoussakis, L. Montero, Zs. Tuza, Proper connection of graphs, Discrete Math. 312 (2012) 2550–2560.
[6] Y. Caro, R. Yuster, Colorful monochromatic connectivity, Discrete Math. 311 (2011) 1786–1792.
[7] S. Chakraborty, E. Fischer, A. Matsliah, R. Yuster, Hardness and algorithms for rainbow connection, J. Comb. Optim. 21 (3) (2011) 330–347.
[8] H. Chang, T.D. Doan, Z. Huang, S. Jendrol', X. Li, I. Schiermeyer, Graphs with conflict-free connection number two, Graphs Comb. 34 (6) (2018) 1553–1563.
[9] H. Chang, Z. Huang, X. Li, Y. Mao, H. Zhao, On conflict-free connection of graphs, Discrete Appl. Math. 255 (2019) 167–182.
[10] H. Chang, M. Ji, X. Li, J. Zhang, Conflict-free connection of trees, J. Comb. Optim. (2018), https://doi.org/10.1007/s10878-018-0363-x.
[11] G. Chartrand, G.L. Johns, K.A. McKeon, P. Zhang, Rainbow connection in graphs, Math. Bohem. 133 (1) (2008) 85–98.
[12] J. Czap, S. Jendrol', J. Valiska, Conflict-free connection of graphs, Discuss. Math., Graph Theory 38 (4) (2018) 911–920.
[13] B. Deng, W. Li, X. Li, Y. Mao, H. Zhao, Conflict-free connection numbers of line graphs, in: Proc. COCOA 2017, Shanghai, China, Lect. Notes Comput. Sci. 10627 (2017) 141–151.
[14] G. Even, Z. Lotker, D. Ron, S. Smorodinsky, Conflict-free coloring of simple geometric regions with applications to frequency assignment in cellular networks, SIAM J. Comput. 33 (2003) 94–136.
[15] M. Katchalski, W. McCuaig, S. Seager, Ordered colorings, Discrete Math. 142 (1995) 141–154.
[16] T.W. Lam, F.L. Yue, Edge ranking of graphs is hard, Discrete Appl. Math. 85 (1998) 71–86.
[17] X. Li, C. Magnant, Properly colored notions of connectivity - a dynamic survey, Theory Appl. Graphs 0 (1) (2015), Art. 2.
[18] X. Li, C. Magnant, Z.M. Qin, Properly Colored Connectivity of Graphs, Springer Briefs in Math., Springer, Switsland, 2018.
[19] X. Li, Y. Shi, Y. Sun, Rainbow connections of graphs: a survey, Graphs Comb. 29 (2013) 1–38.
[20] X. Li, Y. Sun, Rainbow Connections of Graphs, Springer Briefs in Math, Springer, New York, 2012.
[21] X. Li, Y. Sun, An updated survey on rainbow connections of graphs – a dynamic survey, Theory Appl. Graphs 0 (1) (2017) 1–67, Article 3.
[22] X. Li, D. Wu, A survey on monochromatic connections of graphs, Theory Appl. Graphs 0 (1) (2018) 19, Art. 4.
[23] X. Li, Y. Zhang, X. Zhu, Y. Mao, H. Zhao, S. Jendrol', Conflict-free vertex-connections of graphs, Discuss. Math., Graph Theory, https://doi.org/10.7151/dmgt.2116.

Please cite this article in press as: M. Ji et al., (Strong) conflict-free connectivity: Algorithm and complexity, Theoret. Comput. Sci. (2019), https://doi.org/10.1016/j.tcs.2019.10.043

[24] Z. Li, B. Wu, On the maximum value of conflict-free vertex-connection number of graphs, Discrete Math. Algorithms Appl. 10 (05) (2018) 1850059, https://doi.org/10.1142/S1793830918500593.
[25] D.C. Llewellyn, C. Tovey, M. Trick, Local optimization on graphs, Discrete Appl. Math. 23 (1989) 157–178.
[26] K. Ozeki, A set-(g,f)-factor in graphs and its application, in: 2017 Workshop on Colored Notions of Connectivity in Graphs, Nankai University, Tianjin, China, May 29–31 2017.
[27] A. Pothen, The Complexity of Optima1 Elimination Trees, Technical Report CS-8X-13, The Pennsylvania State University, 1988.
[28] S. Smorodinsky, Conflict-free coloring and its applications, in: I. Bárány, K.J. Böröczky, G.F. Tóth, J. Pach (Eds.), Geometry — Intuitive, Discrete, and Convex, in: Bolyai Society Mathematical Studies, vol. 24, Springer, Berlin, Heidelberg, 2013.