

Approximating SHORTEST CONNECTED GRAPH TRANSFORMATION for Trees^{*}

Nicolas Bousquet¹ and Alice Joffard²

¹ Univ. Grenoble Alpes, CNRS, Laboratoire G-SCOP, Grenoble-INP, Grenoble, France.
nicolas.bousquet@grenoble-inp.fr

² LIRIS, Université Claude Bernard, Lyon, France
alice.joffard@liris.cnrs.fr

Abstract. Let G, H be two connected graphs with the same degree sequence. The aim of this paper is to find a transformation from G to H via a sequence of flips maintaining connectivity. A flip of G is an operation consisting in replacing two existing edges uv, xy of G by ux and vy .

Taylor showed that there always exists a sequence of flips that transforms G into H maintaining connectivity. Bousquet and Mary proved that there exists a 4-approximation algorithm of a shortest transformation. In this paper, we show that there exists a 2.5-approximation algorithm running in polynomial time. We also discuss the tightness of the lower bound and show that, in order to drastically improve the approximation ratio, we need to improve the best known lower bounds.

1 Introduction

Sorting by reversals problem. The problem of sorting by reversals has been widely studied in the last twenty years in genomics. The reversal of a sequence of DNA is a common mutation of a genome, that can lead to major evolutionary events. It consists, given a DNA sequence that can be represented as a labelled path x_1, \dots, x_n on n vertices, in turning around a part of it. More formally, a reversal is a transformation that, given two integers $1 \leq i < j \leq n$, transforms the path x_1, \dots, x_n into $x_1, \dots, x_{i-1}, x_j, x_{j-1}, \dots, x_i, x_{j+1}, \dots, x_n$. It is easy to prove that, given two paths on the same vertex set (and with the same leaves), there exists a sequence of reversals that transforms the first into the second. Biologists want to find the minimum number of reversals needed to transform a genome (i.e. a path) into another in order to compute the evolutionary distance between different species.

An input of the SORTING BY REVERSALS problem consists of two paths P, P' with the same vertex set (and the same leaves) and an integer k . The output is positive if and only if there exists a sequence of at most k reversals that transforms P into P' . Caprara proved that the SORTING BY REVERSALS problem is NP-complete [4]. Kececioglu and Sankoff first proposed an algorithm that

^{*} This work was supported by ANR project GrR (ANR-18-CE40-0032).

computes a sequence of reversals of size at most twice the length of an optimal solution in polynomial time [10]. Then, Christie improved it into a $3/2$ -approximation algorithm [5]. The best polynomial time algorithm known so far is a 1.375-approximation due to Berman et al. [2].

A reversal can be equivalently defined as follows: given a path P and two edges ab and cd , a reversal consists in the deletion of the edges ab and cd and the addition of ac and bd that keeps the connectivity of the graph. Indeed, when we transform x_1, \dots, x_n into $x_1, \dots, x_{i-1}, x_j, x_{j-1}, \dots, x_i, x_{j+1}, \dots, x_n$, we have deleted the edges $x_{i-1}x_i$ and x_jx_{j+1} and have created the edges $x_{i-1}x_j$ and x_ix_{j+1} . In this paper, we study the generalization of the SORTING BY REVERSALS problem for trees and general graphs that has also been extensively studied in the last decades.

SHORTEST CONNECTED GRAPH TRANSFORMATION problem. Let $G = (V, E)$ be a graph where V denotes the set of vertices and E the set of edges. For basic definitions on graphs, the reader is referred to [6]. All along the paper, the graphs are loop-free but may admit multiple edges. A *tree* is a connected graph which does not contain any cycle (a multi-edge being considered as a cycle).

The *degree sequence* of a graph G is the sequence of the degrees of its vertices in non-increasing order. Given a non-increasing sequence of integers $S = \{d_1, \dots, d_n\}$, a graph $G = (V, E)$ whose vertices are labeled as $V = \{v_1, \dots, v_n\}$ realizes S if $d(v_i) = d_i$ for all $i \leq n$. Senior [12] gave necessary and sufficient conditions to guarantee that, given a sequence of integers $S = \{d_1, \dots, d_n\}$, there exists a connected multigraph realizing S . Hakimi [7] then proposed a polynomial time algorithm that outputs a connected (multi)graph realizing S if such a graph exists or returns no otherwise.

A *flip* σ (also called *swap* or *switch* in the literature) on two edges ab and cd consists in deleting the edges ab and cd and creating the edges ac and bd (or ad and bc)³. The flip operation that transforms the edges ab and cd into the edges ac and bd is denoted $(ab, cd) \rightarrow (ac, bd)$. When the target edges are not important we will simply say that we flip the edges ab and cd .

Let $S = \{d_1, \dots, d_n\}$ be a non-increasing sequence and let G and H be two graphs on n vertices v_1, \dots, v_n realizing S . The graph G can be transformed into H if there is a sequence $(\sigma_1, \dots, \sigma_k)$ of flips that transforms G into H . Note that since flips do not modify the degree sequence, all the intermediate graphs also realize S . Let $\mathcal{G}(S)$ be the graph whose vertices are the loop-free multigraphs realizing S and where two vertices G and H of $\mathcal{G}(S)$ are adjacent if G can be transformed into H via a single flip. Since the flip operation is reversible, the graph $\mathcal{G}(S)$ is an undirected graph called the *reconfiguration graph* of S . Note that there exists a sequence of flips between any pair of graphs realizing S if and only if the graph $\mathcal{G}(S)$ is connected. Hakimi [8] proved that, for any non-increasing sequence S , if the graph $\mathcal{G}(S)$ is not empty then it is connected.

³ In the case of multigraphs, we simply decrease by one the multiplicities of ab and cd and increase by one the ones of ac and bd .

One can wonder if the reconfiguration graph is still connected when we restrict to graphs with stronger properties. For a graph property Π , let us denote by $\mathcal{G}(S, \Pi)$ the subgraph of $\mathcal{G}(S)$ induced by the graphs realizing S that have the property Π . If we respectively denote by \mathcal{C} and \mathcal{S} the property of being connected and simple, Taylor proved in [13] that $\mathcal{G}(S, \mathcal{C})$, $\mathcal{G}(S, \mathcal{S})$ and $\mathcal{G}(S, \mathcal{C} \wedge \mathcal{S})$ are connected (where \wedge stands for “and”). Let G, H be two graphs of $\mathcal{G}(S, \Pi)$. A sequence of flips *transforms* G into H in $\mathcal{G}(S, \Pi)$ if the sequence of flips transforms G into H and all the intermediate graphs also have the property Π . In other words, a sequence of flips that transforms G into H in $\mathcal{G}(S, \Pi)$ is a path between G and H in $\mathcal{G}(S, \Pi)$. Since [13] ensures that $\mathcal{G}(S, \Pi)$ is connected, one can ask what is the minimum length of such a transformation between G and H . This problem is known to be NP-hard, see e.g. [4]. In this paper we will study the following problem:

SHORTEST CONNECTED GRAPH TRANSFORMATION

Input: Two connected multigraphs G, H with the same degree sequence.

Output: The minimum number of flips needed to transform G into H in $\mathcal{G}(S, \mathcal{C})$.

Note that SHORTEST CONNECTED GRAPH TRANSFORMATION is a generalization of SORTING BY REVERSALS since, when the degree sequence consists of $n - 2$ vertices of degree 2 and two vertices of degree 1, we simply want to find a sequence of reversals of minimum length between two paths. Bousquet and Mary [3] proposed a 4-approximation algorithm for SHORTEST CONNECTED GRAPH TRANSFORMATION. Our main result is the following:

Theorem 1. SHORTEST CONNECTED GRAPH TRANSFORMATION *admits a 2.5-approximation algorithm.*

Section 3 is devoted to the proof of Theorem 1. In order to prove it, we will mainly focus on the SHORTEST TREE TRANSFORMATION problem which is the same as SHORTEST CONNECTED GRAPH TRANSFORMATION except that the input consists of trees with the same degree sequence. Informally speaking, it is due to the fact that if an edge of the symmetric difference appears in some cycle, then we can reduce the size of the symmetric difference in one flip, as observed in [3].

When we desire to give some explicit bound on the quality of a solution, we need to compare it with the length of an optimal transformation. When we do not want to keep connectivity, Will [14] gives an explicit formula of the number of steps in a minimum transformation. When we want to keep connectivity, no such formula is known. Our $5/2$ -approximation algorithm is obtained by comparing it to the formula of Will (which is a lower bound when we want to keep connectivity). In Section 4, we discuss the tightness of this lower bound. We exhibit two graphs G and H such that the length of a shortest transformation between G and H is at least 1.5 times larger than the bound given by [14], and even twice longer under some assumptions on the set of possible flips. In order to prove this result, we generalize some notions introduced for sorting by reversals in [5] to general graphs.

This example ensures that if we want to find an approximation algorithm with a ratio better than 1.5, we might have to improve the algorithm, but overall, we need to improve the lower bound. The formal point and the two graphs G and H can be found in Section 4.

Related Works

Mass spectrometry. Mass spectrometry is a technique used by chemists in order to obtain the formula of a molecule. It provides the mass-to-charge (m/z) ratio spectrum of the molecule from which we can deduce how many atoms of each element the molecule has. With this formula, we would like to find out the nature of the molecule, i.e. the bonds between the different atoms. But the existence of structural isomers points out that there could exist several solutions for this problem. Thus, we would like to find all of them. Since the valence of each atom is known, this problem actually consists in finding all the connected loop-free multigraphs whose degree sequence is the sequence of the valences of those atoms. The reconfiguration problem we are studying here can be a tool for an enumeration algorithm consisting in visiting the reconfiguration graph.

Flips and reconfiguration. The SHORTEST CONNECTED GRAPH TRANSFORMATION problem belongs to the class of reconfiguration problems that received a considerable attention in the last few years. Reconfiguration problems consist, given two solutions of the same problem, in transforming the first solution into the second via a sequence of "elementary" transformations (such as flips) maintaining some properties all along. For more information on reconfiguration problems, the reader is referred for instance to [11].

2 Preliminaries

2.1 Symmetric Difference

Unless specified otherwise, we consider unoriented loop-free multigraphs. Let $G = (V(G), E(G))$ be a graph where $V(G)$ is the set of vertices of G and $E(G)$ is its set of edges. The *intersection* of two graphs G and H on the same set of vertices V is the graph $G \cap H$ with vertex set V , and such that $e \in E(G \cap H)$, with multiplicity m , if the minimum multiplicity of e in both graphs is m . Their *union*, $G \cup H$, has vertex set V , and $e \in E(G \cup H)$, with multiplicity m , if and only if the maximum multiplicity of e in G and H is m . Finally, the *difference* $G - H$ has vertex set V and $e \in E(G - H)$ with multiplicity m if and only if the difference between its multiplicities in G and H is $m > 0$. The *symmetric difference* of G and H is $\Delta(G, H) = (G - H) \cup (H - G)$. We denote by $\delta(G, H)$ the number of edges of $\Delta(G, H)$.

Let G, H be two graphs with the same degree sequence. An edge e of G is *good* if it is in $G \cap H$ and is *bad* otherwise. Note that since G and H have the same degree sequence, the graph $\Delta(G, H)$ has even degree on each vertex and

the number of edges of G incident to v is equal to the number of edges of H incident to v .

Each flip removes at most 4 edges of the symmetric difference. Therefore, the length of a transformation from G to H is at least $\delta(G, H)/4$. In fact, it is possible to obtain a slightly better bound on the length of the transformation. A cycle C in $\Delta(G, H)$ is *alternating* if edges of G and H alternate in C . Since the number of edges of G incident to v is equal to the number of edge of H incident to v in $\Delta(G, H)$, the graph $\Delta(G, H)$ can be partitioned into a collection of alternating cycles. We denote by $mnc(G, H)$ the maximal number of cycles in a partition \mathcal{C} of $\Delta(G, H)$ into alternating cycles. Will [14] proved the following:

Theorem 2 (Will [14]). *Let G, H be two graphs with the same degree sequence. A shortest sequence of flips that transforms G into H (that does not necessarily maintain the connectivity of the intermediate graphs) has length exactly $\frac{\delta(G, H)}{2} - mnc(G, H)$.*

Note that Theorem 2 indeed provides a lower bound for a transformation of SHORTEST CONNECTED GRAPH TRANSFORMATION .

2.2 Basic Facts Concerning Flips

Let $G = (V, E)$ be an unoriented graph and $v \in V(G)$. The set $N_G(v)$ of *neighbours* of v in G is the set of vertices u such that $uv \in E(G)$. Let D be a directed graph and $v \in V(D)$. The set $N_D^-(v)$ of *in-neighbours* of v in D is the set of vertices u such that uv is an arc of D , and the set $N_D^+(v)$ of *out-neighbours* of v in D is the set of vertices u such that vu is an arc of D . When G and D are obvious from the context we will simply write $N(v)$, $N^-(v)$, $N^+(v)$.

The *inverse* σ^{-1} of a flip σ is the flip such that $\sigma \circ \sigma^{-1} = id$, i.e. applying σ and then σ^{-1} leaves the initial graph. The *opposite* $-\sigma$ of a flip σ is the unique other flip that can be applied to the two edges of σ . If we consider a flip $\sigma = (ab, cd) \rightarrow (ac, bd)$, then $\sigma^{-1} = (ac, bd) \rightarrow (ab, cd)$ and $-\sigma = (ab, cd) \rightarrow (ad, bc)$. Note that $-\sigma$ is a flip deleting the same edges as σ while σ^{-1} cancels the flip σ . When we transform a graph G into another graph H , we can flip the edges of G or the edges of H . Indeed, applying the sequence of flips $(\sigma_1, \dots, \sigma_i)$ to transform G into a graph K , and the sequence of flips (τ_1, \dots, τ_j) to transform H into K is equivalent to applying the sequence $(\sigma_1, \dots, \sigma_i, \tau_j^{-1}, \dots, \tau_1^{-1})$ to transform G into H .

Let $G = (V, E)$ be a connected graph and let H be a graph with the degree sequence of G . A flip is *good* if it flips bad edges and creates at least one good edge. It is *bad* otherwise. A *connected flip* is a flip such that its resulting graph is connected. Otherwise, it is *disconnected*. A *path* from $a \in V$ to $b \in V$ is a sequence of vertices (v_1, \dots, v_k) such that $a = v_1$, $b = v_k$, for every integer $i \in [k - 1]$, $v_i v_{i+1} \in E(G)$ and there is no repetition of vertices. Similarly, a path from e to f with $e, f \in E(G)$ is a path from an endpoint of e to an endpoint of f that does not contain the other endpoint of e and of f . A path between x and y (vertices or edges) is a path from x to y or a path from y to x . The *content* of a path is its set of vertices. We say that an edge e *belongs to* (or *is on*) a path P if both endpoints

of e appear consecutively in P . The *intersection* $P_1 \cap P_2$ of two paths P_1 and P_2 is the intersection of their contents. The vertices of a sequence (v_1, \dots, v_k) are *aligned* in G if there exists a path P which is the concatenation of $k - 1$ paths $P_1 P_2 \dots P_{k-1}$ where P_i is a path from v_i to v_{i+1} for $i \in [k - 1]$. Note that we might have $v_i = v_{i+1}$ and then $P_i = v_i$.

Note that, for every connected graph G , if $ab, cd \in E(G)$, $ab \neq cd$, then (a, b, c, d) , (a, b, d, c) , (b, a, c, d) , or (b, a, d, c) are aligned. Moreover, if G is a tree, exactly one of them is aligned. Let G be a connected graph and $a, b, c, d \in V(G)$ such that (a, b, c, d) are aligned. The *in-area* of the two edges ab and cd is the connected component of $G \setminus \{ab, cd\}$ containing the vertices b and c . The other components are called *out-areas*. The following lemma links the connectivity of a flip and the alignment of its vertices:

Lemma 1. (\star) *Let G be a connected graph and $ab, cd \in E(G)$ where a, b, c and d are pairwise distinct vertices of G . If (a, b, c, d) or (b, a, d, c) are aligned in G , then the flip $(ab, cd) \rightarrow (ac, bd)$ is connected. If G is a tree, then it is also a necessary condition.*

The proofs of all the statements marked with a \star are not included in this extended abstract. Lemma 1 ensures that, for trees, exactly one of the two flips σ and $-\sigma$ is connected.

Let e and f be two vertex-disjoint edges of a tree T , and let σ_2 be a flip in T that does not flip e nor f . The flip σ_2 *depends on* e and f if applying the connected flip on e and f changes the connectivity of σ_2 . By abuse of notation, for any two flips σ_1 and σ_2 on pairwise disjoint edges, σ_2 *depends on* σ_1 if σ_2 depends on the edges of σ_1 . The flip σ_1 *sees* σ_2 if exactly one of the edges of σ_2 is on the path linking the two edges of σ_1 in G .

The following lemma links the dependency of two flips and the position of their edges in a tree:

Lemma 2. (\star) *Let T be a tree and σ_1 and σ_2 be two flips on T , whose edges are pairwise distinct. The three following points are equivalent:*

1. σ_2 depends on σ_1 ,
2. σ_1 depends on σ_2 ,
3. σ_2 sees σ_1 and σ_1 sees σ_2 .

We now give two consequences of applying a connected flip.

Lemma 3. (\star) *Let T be a tree and σ_1 and σ_2 be two flips on T with pairwise disjoint edges, where σ_1 is connected. Let T' be the tree obtained after applying σ_1 to T . The flip σ_1^{-1} sees σ_2 in T' if and only if σ_1 sees σ_2 in T . And σ_2 sees σ_1^{-1} in T' if and only if σ_2 sees σ_1 in T .*

Lemma 4. (\star) *Let T be a tree and σ_1 , σ_2 and σ_3 be three flips on T whose edges are pairwise disjoint and such that σ_1 sees σ_2 , σ_2 sees σ_3 , and σ_2 is connected. Let T' be the tree obtained by applying the flip σ_2 to T . The flip σ_1 sees σ_3 in T if and only if σ_1 does not see σ_3 in T' .*

3 Upper Bound

Let us recall a result of [3].

Lemma 5. *Let G, H be two connected graphs with the same degree sequence. There exists a sequence of at most two flips that decreases $\delta(G, H)$ by at least 2. Moreover, if there is an alternating C_4 in $\Delta(G, H)$, it can be removed in at most 2 steps, without modifying the rest of the graph.*

Lemma 5 immediately implies the following:

Corollary 1. SHORTEST CONNECTED GRAPH TRANSFORMATION admits a polynomial time 4-approximation algorithm.

The goal of the rest of this section is to improve the approximation ratio. The crucial lemma is the following:

Lemma 6. *Let G, H be two trees with the same degree sequence. There exists a sequence of at most 3 flips that decreases $\delta(G, H)$ by at least 4. Moreover, this sequence only flips bad edges.*

Proof. Let G' be the graph whose vertices are the connected components of $G \cap H$ and where two vertices S_1 and S_2 of G' are incident if there exists an edge in G between a vertex of S_1 and a vertex of S_2 . In other words, G' is obtained from G by contracting every connected component of $G \cap H$ into a single vertex. Note that the edges of G' are the edges of $G - H$. Moreover, as G is a tree, G' also is. We can similarly define H' . Note that G' and H' have the same degree sequence.

Let S_1 be a leaf of G' and S_2 be its parent in G' . Let us show that S_2 is not a leaf of G' . Indeed, otherwise G' would be reduced to a single edge. In particular, $E(G - H)$ would contain only one edge. Since the degree sequence of $G - H$ and $H - G$ are the same, the edge of $H - G$ would have to be the same, a contradiction. Thus, we can assume that S_2 is not a leaf. Let $u_1 u_2$ be the edge of $G - H$ between $u_1 \in S_1$ and $u_2 \in S_2$. Since $G - H$ and $H - G$ have the same degree sequence and S_1 is a leaf of G' , there exists a unique vertex v_1 such that $u_1 v_1 \in E(H - G)$. Moreover there exists a vertex v_2 such that $u_2 v_2 \in E(H - G)$.

Let us first assume that $v_1 = v_2$. Then there exists a vertex w distinct from u_1 and u_2 such that $v_1 w \in E(G - H)$ since v_1 has degree at least 2 in $H - G$. Since S_1 is a leaf of G' , $w \notin S_1$ and either (u_1, u_2, v_1, w) or (u_1, u_2, w, v_1) are aligned in G . If (u_1, u_2, v_1, w) are aligned then the flip $(u_1 u_2, v_1 w) \rightarrow (u_1 v_1, u_2 w)$ in G is connected and creates the edge $u_1 v_1$. If (u_1, u_2, w, v_1) are aligned then $(u_1 u_2, v_1 w) \rightarrow (u_1 w, u_2 v_1)$ is connected and creates the edge $u_2 v_1 = u_2 v_2$. In both cases, we reduce the size of the symmetric difference by at least 2 in one flip, and we can conclude with Lemma 5.

From now on, we assume that $v_1 \neq v_2$. We focus on the alignment of u_1, v_1, u_2 and v_2 in H . Since S_1 is a leaf of G' , it is also a leaf of H' . Thus, v_1 is on the path from u_1 to u_2 and either (u_1, v_1, u_2, v_2) or (u_1, v_1, v_2, u_2) are aligned. If (u_1, v_1, u_2, v_2) are aligned then Lemma 1 ensures that $(u_1 v_1, u_2 v_2) \rightarrow (u_1 u_2, v_1 v_2)$

is connected in H and reduces the size of the symmetric difference by at least 2. We can conclude with Lemma 5. Thus, we can assume that (u_1, v_1, v_2, u_2) are aligned in H (see Figure 1 for an illustration).

Let us first remark that if u_2 has degree at least 2 in $H - G$ (or equivalently in $G - H$), then we are done. Indeed, if there exists $w \neq v_2$ such that $u_2w \in E(H - G)$ then, since (u_1, v_1, v_2, u_2) are aligned, (u_1, v_1, u_2, w) have to be aligned. Indeed, v_2u_2 is the only edge of $H - G$ on the path from v_1 to u_2 incident to u_2 . Thus the flip $(u_1v_1, u_2w) \rightarrow (u_1u_2, v_1w)$ is connected in H . Since it reduces $\delta(G, H)$ by at least 2, we can conclude with Lemma 5.

From now on, we will assume that u_2 has degree 1 in $H - G$. Let H_3 (resp. H_4) be the connected component of v_1 and v_2 (resp. u_2) in $H \setminus \{u_1v_1, u_2v_2\}$, which exists since (u_1, v_1, v_2, u_2) are aligned. Note that the third component of $H \setminus \{u_1v_1, u_2v_2\}$ is reduced to S_1 . By definition, H_3 is the in-area of u_1v_1 and u_2v_2 .

We now show that there exists an edge $u_3u_4 \in E(G - H)$, with $u_3 \in H_3$, $u_4 \in H_4$, and such that the connected component S_4 of $G \cap H$ containing u_4 is not a leaf of G' . Indeed, since G is connected, there exists a path P from v_1 to u_2 in G . Since u_1u_2 is the only edge of $G - H$ that has an endpoint in S_1 , this path does not contain any vertex of S_1 . Thus, it necessarily contains an edge u_3u_4 between a vertex u_3 of H_3 and a vertex u_4 of H_4 . Since H_3 and H_4 are anticomplete in $G \cap H$, $u_3u_4 \in E(G - H)$. Moreover, the connected component S_4 of $G \cap H$ containing u_4 is not a leaf of G' , as it is either S_2 which is not a leaf, or P has to leave S_4 at some point with an edge of $G - H$ since P ends in $u_2 \in S_2$.

Since u_3 and u_4 have the same degree in $G - H$ and $H - G$, there exist v_3, v_4 such that $u_3v_3, u_4v_4 \in E(H - G)$. Moreover, since S_4 is not a leaf of G' (and thus of H'), there exists an edge of $H - G$ between a vertex $u_5 \in S_4$ and a vertex $v_5 \in V \setminus S_4$ where $u_5v_5 \neq u_4v_4$.

Let us prove that u_3, v_3, u_4 and v_4 are pairwise distinct. By definition, we have $u_3 \neq v_3, u_4 \neq v_4$ and $u_3 \neq u_4$. Moreover, since $u_3u_4 \in E(G - H)$, $u_3 \neq v_4$ and $u_4 \neq v_3$. Thus, the only vertices that can be identical are v_3 and v_4 . If $v_3 = v_4$, since $u_3 \in H_3, u_4 \in H_4$, and v_2u_2 is the only edge of $H - G$ from H_3 to H_4 , then either $v_3 = v_4 = v_2$ or $v_3 = v_4 = u_2$. In the first case, $u_4 = u_2$ since v_2u_2 is the only edge of $H - G$ from H_3 to H_4 . Thus, u_2 is the endpoint of both u_1u_2 and u_2u_3 in $G - H$. In the second case, u_2 is the endpoint of both u_2u_3 and u_2u_4 in $H - G$. Thus, in both cases, u_2 has degree at least 2 in $H - G$, a contradiction.

We now focus on the alignment of u_3, u_4, v_3 and v_4 in H . If (v_3, u_3, v_4, u_4) or (u_3, v_3, u_4, v_4) are aligned, then the flip $(u_3v_3, u_4v_4) \rightarrow (u_3u_4, v_3v_4)$ is connected in H and reduces the size of the symmetric difference by at least 2, since $u_3u_4 \in E(G - H)$. Note that the flip is well-defined since all the vertices are distinct. Thus, we can conclude with Lemma 5. Therefore, we can assume that (u_3, v_3, v_4, u_4) or (v_3, u_3, u_4, v_4) are aligned in H .

We give, in each case, a sequence of three flips that decreases the size of the symmetric difference by at least 4. Due to space restriction, the proof that

those flips can be applied and maintain the connectivity are not included in this extended abstract.

Case 1. (u_3, v_3, v_4, u_4) are aligned. (See Figure 1 for an illustration).

We successively apply the flips $\sigma_1 : (u_2v_2, u_5v_5) \rightarrow (u_2v_5, u_5v_2)$, $\sigma_2 : (u_3x, u_4v_4) \rightarrow (u_3u_4, xv_4)$ where $x = u_5$ if $u_3 = v_2$ and $v_3 = u_2$, and $x = v_3$ otherwise, and $\sigma_3 : (u_1v_1, u_2v_5) \rightarrow (u_1u_2, v_1v_5)$ in H . Since $u_1u_2, u_3u_4 \in E(G - H)$, this sequence of flips indeed reduces $\delta(G, H)$ by at least 4.

Case 2. (v_3, u_3, u_4, v_4) are aligned.

We apply $\sigma_1 : (u_2v_2, u_4v_4) \rightarrow (u_2v_4, u_4v_2)$, $\sigma_2 : (u_3v_3, u_4v_2) \rightarrow (u_3u_4, v_2v_3)$ then $\sigma_3 : (u_1v_1, u_2v_4) \rightarrow (u_1u_2, v_1v_4)$ to H . Again, $u_1u_2, u_3u_4 \in E(G - H)$ and it reduces $\delta(G, H)$ by at least 4.

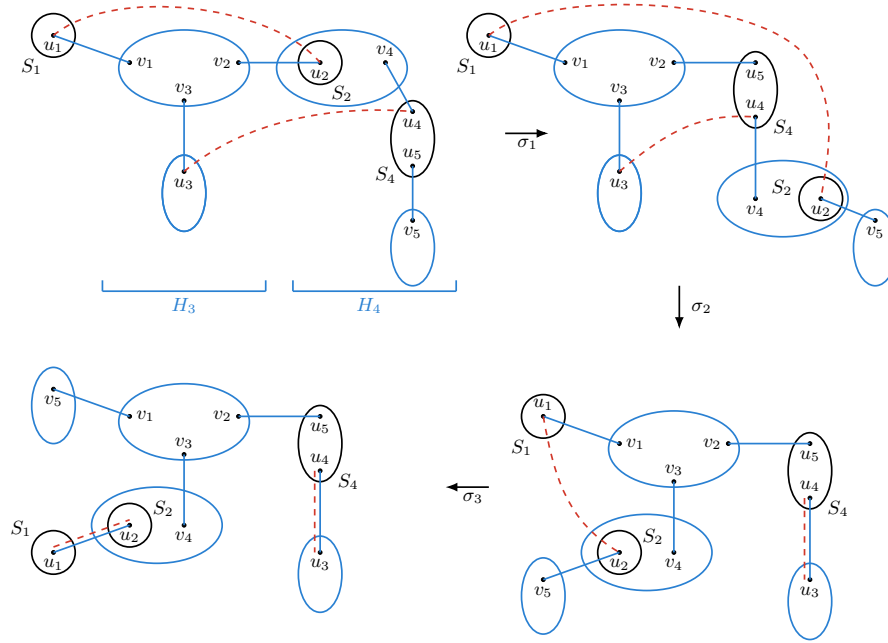


Fig. 1: The three flips $\sigma_1 : (u_2v_2, u_5v_5) \rightarrow (u_2v_5, u_5v_2)$, $\sigma_2 : (u_3v_3, u_4v_4) \rightarrow (u_3u_4, v_3v_4)$ and $\sigma_3 : (u_1v_1, u_2v_5) \rightarrow (u_1u_2, v_1v_5)$ applied to the graph H where (u_3, v_3, v_4, u_4) are aligned. The blue full edges are in $E(H - G)$ and the red dashed edges are in $E(G - H)$.

Therefore, in all the cases, we have found a sequence of three flips whose edges are in the symmetric difference and that reduce $\delta(G, H)$ by at least 4. Moreover, the proof immediately provides a polynomial time algorithm to find such a sequence. \square

Note that Lemma 6 allows to obtain a 3-approximation algorithm for SHORTEST CONNECTED GRAPH TRANSFORMATION. Indeed, as shown in the proof of Lemma 1 in [3], as long as there exists an edge of the symmetric difference in a cycle of G , one can reduce the size of the symmetric difference by 2 in one step. Afterwards, we can assume that the remaining graphs $G - H$ and $H - G$ are trees. By Lemma 6, in three flips, the symmetric difference of the optimal solution decreases by at most 12 while our algorithm decreases it by at least 4. (Note that free to try all the flips, finding these flips is indeed polynomial). But we can actually improve the approximation ratio. The idea consists in treating differently short cycles. A *short* cycle is a C_4 , a *long* cycle is a cycle of length at least 6. We now give the main result of this section.

Theorem 3. SHORTEST CONNECTED GRAPH TRANSFORMATION admits a $5/2$ -approximation algorithm running in polynomial time. It becomes a $9/4$ -approximation algorithm if $\Delta(G, H)$ does not contain any short cycle.

Proof. Let \mathcal{C} be an optimal partition of $\Delta(G, H)$ into alternating cycles, i.e. a partition with $mnc(G, H)$ cycles. Let c be the number of short cycles in \mathcal{C} . Berge and Ito [1] provide a polynomial time algorithm to find a partition of $\Delta(G, H)$ into alternating cycles having at least $\frac{c}{2}$ short cycles. Lemma 5 ensures that we can remove their $2c$ edges from the symmetric difference in at most c flips. If an edge of the symmetric difference is in a cycle of G or H , then in one step we can reduce the symmetric difference by 2 [3]. Otherwise, by Lemma 6, we can remove the remaining $\delta(G, H) - 2c$ edges using at most $\frac{3(\delta(G, H) - 2c)}{4}$ flips in polynomial time. Therefore, we can transform G into H with at most $c + \frac{3(\delta(G, H) - 2c)}{4}$ flips.

Let us now provide a lower bound on the length of a shortest transformation from G to H . By definition, \mathcal{C} contains c short cycles. Theorem 2 ensures that we need at least c steps to remove the short cycles, plus $\ell - 1$ flips to remove each cycle of length 2ℓ . Therefore, we need at least $\frac{\delta(G, H) - 4c}{3}$ flips to remove the $\delta(G, H) - 4c$ remaining edges from the symmetric difference.

The ratio between the upper bound and the lower bound is

$$f(c) := \frac{c + \frac{3\delta(G, H) - 6c}{4}}{c + \frac{\delta(G, H) - 4c}{3}} = \frac{3(3\delta(G, H) - 2c)}{4(\delta(G, H) - c)}.$$

The function f being increasing and since the number of short cycles in \mathcal{C} cannot exceed $\frac{\delta(G, H)}{4}$, we have $f(c) \leq f(\frac{\delta(G, H)}{4}) = \frac{5}{2}$. It gives a $\frac{5}{2}$ -approximation in polynomial time. Moreover, when there is no alternating short cycle in $\Delta(G, H)$, $c = 0$. Since $f(0) = \frac{9}{4}$, we obtain a $\frac{9}{4}$ -approximation. \square

4 Discussion on the Tightness of the Lower Bound

In this section, we discuss the quality of the lower bound of Theorem 2. We first prove that if we only flip bad edges of the same cycle of the symmetric difference then the length of a shortest transformation can be almost twice longer

than the one given by the lower bound of Theorem 2. In order to prove it, we generalize several techniques and results of Christie [5], proved for the SORTING BY REVERSALS problem.

Note that the result of Hannenhalli and Pevzner [9] actually proves that in the case of paths, when the symmetric difference only contains vertex-disjoint short cycles, it is not necessarily optimal to only flip edges of the same cycle. However, studying this restriction gives us a better understanding of the general problem.

We also prove that, if we only flip bad edges (which are not necessarily in the same cycle of the symmetric difference), then the length of a shortest transformation can be almost $3/2$ times longer than the one given by the lower bound. Note that all the existing approximation algorithms for SORTING BY REVERSALS and SHORTEST CONNECTED GRAPH TRANSFORMATION only flip bad edges. But again no formal proof guarantees that there always exists a shortest transformation where we only flip bad edges.

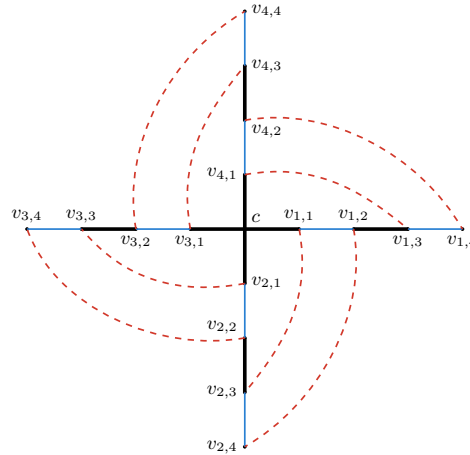


Fig. 2: The graphs G_4 and H_4 . The black thick edges are in $E(G_4 \cap H_4)$, the blue thin edges are in $E(G_4 - H_4)$ and the red dashed edges are in $E(H_4 - G_4)$.

Both results (whose proofs are not included in this extended abstract) are obtained with the same graphs G_k and H_k represented in Figure 2 for $k = 4$. For any $k \geq 2$, let $G_k = (V_k, E(G_k))$ and $H_k = (V_k, E(H_k))$ be the graphs with $V_k = \{v_{i,j}, 1 \leq i \leq k, 1 \leq j \leq 4\} \cup \{c\}$, $E(G_k) = \bigcup_{i \in [k]} \{cv_{i,1}, v_{i,1}v_{i,2}, v_{i,2}v_{i,3}, v_{i,3}v_{i,4}\}$, and $E(H_k) = \bigcup_{i \in [k]} \{cv_{i,1}, v_{i,1}v_{i+1,3}, v_{i,2}v_{i,3}, v_{i,2}v_{i+1,4}\}$, where the additions are defined modulo k . One can easily check that, in this construction, both G_k and H_k are the subdivisions of a star where each branch has 4 vertices. Note that $\Delta(G, H)$ is the disjoint union of k short cycles. Moreover, the partition of $\Delta(G, H)$ into alternating cycles is unique.

The proof of Lemma 5 ensures that there is a transformation from G_k to H_k in at most $2k$ steps where we only flip bad edges in the same cycle of the symmetric difference. So our first result is tight with our assumptions. We conjecture that the length of a shortest transformation from G_k to H_k is at least $2k - 1$ without any assumption on the set of possible flips.

References

1. S. Breg and H. Ito. Transforming graphs with the same graphic sequence. *Journal of Information Processing*, 25:627–633, 2017.
2. P. Berman, S. Hannenhalli, and M. Karpinski. 1.375-Approximation Algorithm for Sorting by Reversals. In R. Möhring and R. Raman, editors, *Algorithms — ESA 2002*, page 200–210, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.
3. N. Bousquet and A. Mary. Reconfiguration of graphs with connectivity constraints. In *Approximation and Online Algorithms - WAOA 2018*, pages 295–309, 2018.
4. A. Caprara. Sorting by reversals is difficult. In *Proceedings of the First Annual International Conference on Computational Molecular Biology, RECOMB '97*, pages 75–83. ACM, 1997.
5. D. A. Christie. A $3/2$ -approximation algorithm for sorting by reversals. In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '98*, pages 244–252, 1998.
6. R. Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, Heidelberg, third edition, 2005.
7. S. L. Hakimi. On realizability of a set of integers as degrees of the vertices of a linear graph. i. *Journal of the Society for Industrial and Applied Mathematics*, 10(3):496–506, 1962.
8. S. L. Hakimi. On realizability of a set of integers as degrees of the vertices of a linear graph ii. uniqueness. *Journal of the Society for Industrial and Applied Mathematics*, 11(1):135–147, 1963.
9. S. Hannenhalli and P. A. Pevzner. Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals. *Journal of the ACM (JACM)*, 46(1):1–27, 1999.
10. J. D. Kececioglu and D. Sankoff. Exact and approximation algorithms for sorting by reversals, with application to genome rearrangement. *Algorithmica*, 13(1/2):180–210, 1995.
11. N. Nishimura. Introduction to reconfiguration. *preprint*, 2017.
12. J. Senior. Partitions and their representative graphs. *American Journal of Mathematics*, 73(3):663–689, 1951.
13. R. Taylor. Constrained switchings in graphs. In K. L. McAvaney, editor, *Combinatorial Mathematics VIII*, pages 314–336, Berlin, Heidelberg, 1981. Springer Berlin Heidelberg.
14. T. G. Will. Switching distance between graphs with the same degrees. *SIAM Journal on Discrete Mathematics*, 12(3):298–306, 1999.