Journal of Information Processing Vol.25 627–633 (Aug. 2017)

[DOI: 10.2197/ipsjjip.25.627]

Regular Paper

Transforming Graphs with the Same Graphic Sequence

Sergey Bereg[1,a) Hiro Ito[2,b)

Received: November 8, 2016, Accepted: February 9, 2017

Abstract: Let G and H be two graphs with the same vertex set V. It is well known that a graph G can be transformed into a graph H by a sequence of 2-switches if and only if every vertex of V has the same degree in both G and H. We study the problem of finding the minimum number of 2-switches for transforming G into H.

Keywords: graphs, degree sequence, 2-switch, graph transformation, approximation algorithm

1. Introduction

A graphic sequence is the sequence of numbers that are vertex degrees of a graph. Any degree sequence whose sum is even can be realized by a multigraph having loops [8]. In this paper we consider simple graphs (graphs without loops and multiple edges). Erdös and Gallai [6] found a characterization of graphic sequences.

Theorem 1 (Erdös and Gallai [6]) *A sequence of positive numbers $d_1 \geq d_2 \geq \ldots \geq d_n$ is graphic if and only if $d_1 + d_2 + \ldots + d_n$ is even and the inequalities*

$$\sum_{i=1}^{k} d_i \leq k(k-1) + \sum_{i=k+1}^{n} \min\{k, d_i\}$$

hold for every k.

Hakimi [8] and Havel [10] found another characterization of graphic sequences.

Theorem 2 (Hakimi [8], Havel [10]) *For $n \geq 1$, a sequence S of n nonnegative integers is graphic if and only if S' is graphic, where S' is the sequence of size $n-1$ obtained from S by deleting its largest element d and subtracting 1 from its d next largest elements. The only 1-element graphic sequence is $d_1 = 0$.*

The following transformation of a graph preserves the degree sequence.

Definition 3 *A 2-switch is the replacement of a pair of edges (a, b) and (c, d) in a simple graph by the edges (a, c) and (b, d), given that (a, c) and (b, d) did not appear in the graph originally,* see **Fig. 1**.

It is clear that the degrees of the vertices remain unchanged when a 2-switch is applied to a graph. The following theorem shows that two graphs with the same graphic sequence can be transformed one to the other using 2-switches.

Theorem 4 *If G and H are two simple graphs with vertex set V, then $d_G(v) = d_H(v)$ for every $v \in V$ if and only if there is a sequence of 2-switches that transforms G into H.*

The graphs G and H have the same set of vertices. It can also be viewed as two labelled graphs with the same set of labels. We assume that n is the number of vertices in G (and in H).

Probably the earliest reference of Theorem 4 is Berge [2] stating that the 2-switch graph on the set of graphs with fixed degree sequence is connected. It also can be found in West [13, p.45]. In the proof of Theorem 4 both G and H are reduced to a *canonical graph* with vertex set V. Each reduction uses at most $m-1$ transformations where m is the number of edges in G (see more details in Section 2). Thus, the smallest number of 2-switches transforming G to H is at most $2m-2$. Finding the minimum number of 2-switches transforming given G and H is of particular interest of this paper.

Let $G = (V, E_G)$ and $H = (V, E_H)$ be two simple graphs such that $d_G(v) = d_H(v)$ for every $v \in V$. We consider a new graph $F(G, H)$ or just F defined as (V, E_F) where $E_F = E_G \cup E_H - E_G \cap E_H$. We color the edges of F with two colors as follows. An edge e is colored (i) red if $e \in E_G - E_H$, and (ii) blue if $e \in E_H - E_G$. The number of red edges and the number of blue edges in F are equal. We denote it by $r(G, H)$.

A *red-blue alternating walk* in F (or an *alternating walk*, simply) is an even length cycle such that (i) the edges in the walk are pairwise distinct, and (ii) the edge colors in the walk alternate (red-blue-red-..-blue). Note that the vertices of a walk may not be pairwise distinct. The set of edges of F can be decomposed into red-blue alternating walks. Let $p(G, H)$ be the maximum number of walks in a decomposition of a F into red-blue alternating walks. Our main result is the following theorem.

Theorem 5 *Let $G = (V, E_G)$ and $H = (V, E_H)$ be two simple graphs such that $d_G(v) = d_H(v)$ for every $v \in V$. The small-*

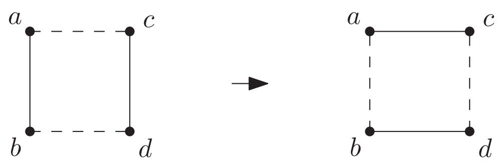

Fig. 1 2-switch.

1 Department of Computer Science, The University of Texas at Dallas, Richardson, TX 75080, USA.
2 Department of Computer and Network Engineering, School of Informatics and Engineering, The University of Electro-Communications, Tokyo 182–8585, Japan.
a) besp@utdallas.edu
b) itohiro@uec.ac.jp

est number of 2-switches for transforming G into H is equal to $r(G,H) - p(G,H)$.

2. Preliminaries

We call the coloring of graph F (defined in the previous section) *even* since, for any vertex v, equal number of red and blue edges are incident to v. It implies that the number of red edges and the number of blue edges in F are equal (it is denoted by $r(G,H)$). It also implies that the edges of F can be decomposed into edge-disjoint cycles such that each cycle is a red-blue alternating walk. We denote by $p(G,H)$ the smallest number of cycles in such a decomposition.

We also consider the complete graph K_n with the set of vertices V. Clearly, F is the subgraph of K_n. We color the edges of K_n: the common edges of F and K_n are colored red and blue as before and the other edges are colored black and white as follows. An edge e is colored

- black if $e \in E_H \cap E_G$,
- white if $e \notin E_G \cup E_H$.

The proof Theorem 4 uses a *canonical graph* C with vertex set V defined inductively as follows. Let $v_1, v_2, \ldots, v_n$ be the vertices of V sorted such that their degrees form a non-increasing sequence $d(v_1) \geq d(v_2) \geq \ldots d(v_n)$. Consider the sequence

$$d(v_2) - 1, d(v_3) - 1, \ldots, d(v_{k+1}) - 1, d(v_{k+2}), \ldots, d(v_n)$$

where $k = d(v_1)$. By Theorem 2 it is a graphic sequence. Let C' be a *canonical graph* corresponding to it. Then C is obtained from C' by adding a new vertex v_1 and edges $(v_1, v_2), (v_1, v_3), \ldots, (v_1, v_{k+1})$.

The main argument in the proof of Theorem 4 is as follows. Consider two sets $S = \{v_2, v_3, \ldots, v_{k+1}\}$ and $N(v_1)$, the set of neighbors of v_1. If $S = N(v_1)$, then the theorem holds by induction hypothesis. If $S \neq N(v_1)$, then any edge connecting v_1 and a vertex $z \notin S$ can be flipped to an edge connecting v_1 and a vertex $x \in S - N(v_1)$ using a 2-switch. By repeating this step we spend at most k transformations for the induction step. With every 2-switch we insert a new edge of the canonical graph C. In the last 2-switch we add two edges of C. So, the total number of 2-switches is at most $m - 1$.

This gives an upper bound of $2m - 2$ for the number of 2-switches to transform the graph G to the graph H. Theorem 5 implies that, if $m > 0$, then at most $(m - 1)$ 2-switches suffice since $r(G,H) \leq m$ and $p(G,H) \geq 1$.

3. Main Result

Our main result (Theorem 5) characterizes the 2-switch distance between two graphs, i.e., the smallest number of 2-switches for transforming one graph into the other. Let $\psi(G,H) = r(G,H) - p(G,H)$ for two graphs satisfying the condition of Theorem 5. Then, Theorem 5 states that the 2-switch distance between G and H is equal to $\psi(G,H)$. We prove that $\psi(G,H)$ is a lower bound (Lemma 6) and an upper bound (Lemma 7) for the 2-switch distance between G and H.

Lemma 6 (Lower Bound) *Let G' be the graph obtained by a 2-switch from G. Then*

$$\psi(G',H) \geq \psi(G,H) - 1. \tag{1}$$

Proof: Consider any 2-switch of edges ab and cd by the edges ac and bd. The edges ab and cd are each colored red or black. The edges ac and bd are each colored blue or white. Let C' be a partition of $F(G',H)$ into $p(G',H)$ alternating walks.

Case 1. Both edges (a,b) and (c,d) are red.

Suppose that the colors of (a,c) and (b,d) are blue, see **Fig. 2** (a). Then $r(G,H) = r(G',H) + 2$. The walks of C' and $abdca$ form a partition of the set of edges of $F(G,H)$ into alternating walks. Therefore $p(G,H) \geq p(G',H) + 1$. The bound Eq. (1) follows.

Suppose that the colors of (a,c) and (b,d) are blue and white respectively, see Fig. 2 (b). Then $r(G,H) = r(G',H) + 1$. One of the walks of C' contains (b,d). We replace (b,d) with $bacd$ to obtain the set of alternating walks for $F(G,H)$. Thus $p(G,H) \geq p(G',H)$. The bound Eq. (1) follows.

Suppose that the colors of (a,c) and (b,d) are white, see Fig. 2 (c) and (d). Then $r(G,H) = r(G',H)$. If the red edges (a,c) and (b,d) belong to different walks C_1 and C_2 of C', then $C_1 - \{(a,c)\}$ and $C_2 - \{(b,d)\}$ can be combined in one alternating walk with (a,b) and (c,d) in $F(G,H)$, see Fig. 2 (c). Thus $p(G,H) \geq p(G',H) - 1$. The bound Eq. (1) follows.

If the red edges (a,c) and (b,d) are connected in one alternating walk C, then $p(G,H) \geq p(G',H) + 1$, see Fig. 2 (d). The bound Eq. (1) follows.

Case 2. The edge (a,b) is red and the edge (c,d) is black.

Suppose that the colors of (a,c) and (b,d) are blue, see **Fig. 3** (a). Then $r(G,H) = r(G',H) + 1$. By replacing the edge (c,d) in an alternating walk from C' by $cabd$ we bound $p(G,H) \geq p(G',H)$. The bound Eq. (1) follows.

Suppose that the colors of (a,c) and (b,d) are blue and white respectively, see Fig. 3 (b). Then $r(G,H) = r(G',H)$. If the edges (b,d) and (c,d) are in two alternating walks C_1 and C_2 of C', then they can be combined in one alternating walk $C_1 \cup C_2 \cup \{(a,b),(a,c)\} - \{(b,d),(c,d)\}$, see Fig. 3 (c). If the edges (b,d) and (c,d) are in a same alternating walk, then they can be replaced by (a,b) and (a,c), see Fig. 3 (d). In both cases $p(G,H) \geq p(G',H) - 1$. The bound Eq. (1) follows.

Suppose that the colors of (a,c) and (b,d) are white, see Fig. 3 (e). Then $r(G,H) = r(G',H) - 1$. To bound $p(G,H)$ we check the walks of C' containing the edges (a,c), (c,d) and (b,d). If there are three walks, then they can be combined in one walk for G, see Fig. 3 (e). The number of walks can be two or one, see Fig. 3 (f). In all cases $p(G,H) \geq p(G',H) - 2$ and the bound Eq. (1) follows.

Case 3. The edges (a,b) and (c,d) are black.

Suppose that the colors of (a,c) and (b,d) are blue, see **Fig. 4** (a). Then $r(G,H) = r(G',H)$ and $p(G,H) \geq p(G',H) - 1$ by an argument similar to Case 1 where (a,c) and (b,d) are white. The bound Eq. (1) follows.

Suppose that the colors of (a,c) and (b,d) are blue and white respectively, see Fig. 4 (b). Then $r(G,H) = r(G',H) - 1$ and $p(G,H) \geq p(G',H) - 2$ by an argument similar to Case 2 where (a,c) and (b,d) are white. The bound Eq. (1) follows.

Suppose that the colors of (a,c) and (b,d) are white, see Fig. 4 (c). Then $r(G,H) = r(G',H) - 2$. If $abcda$ is a walk of C', then $p(G,H) \geq p(G',H) - 1$. If the edges of $abcda$ par-

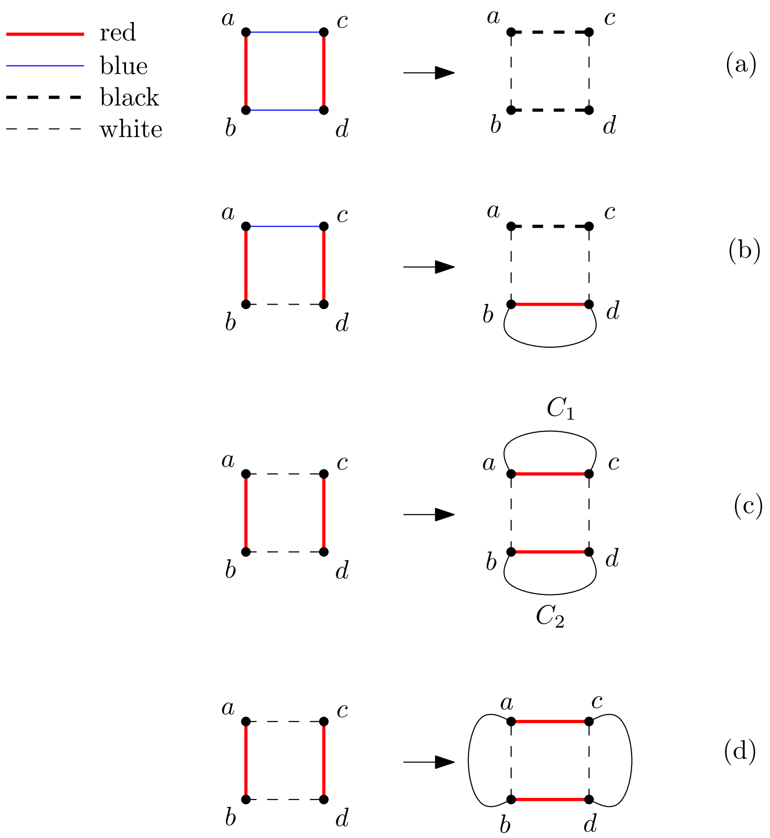

Fig. 2 Case 1 of the lower bound. Red and blue edges are shown as solid lines, bold and thin respectively. Black and white edges are shown as dashed lines, bold and thin respectively. The edges (a, b) and (c, d) are red and the edges (a, c) and (b, d) are (a) both blue, and (b) blue and white, and (c), (d) both white.

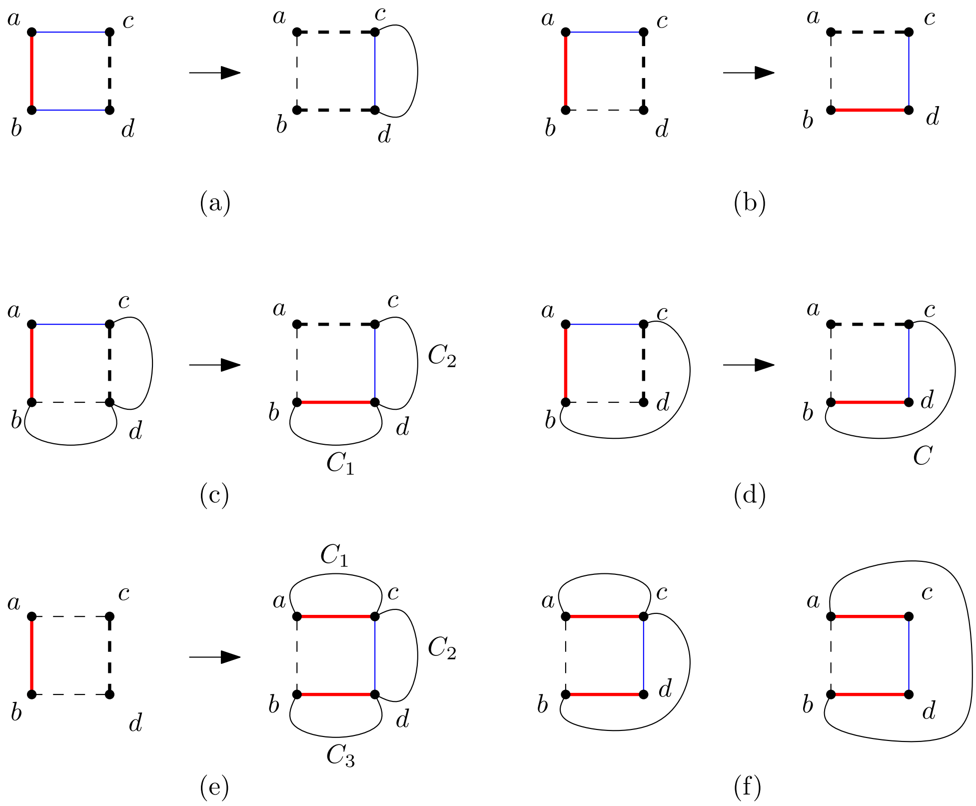

Fig. 3 Case 2 of the lower bound.

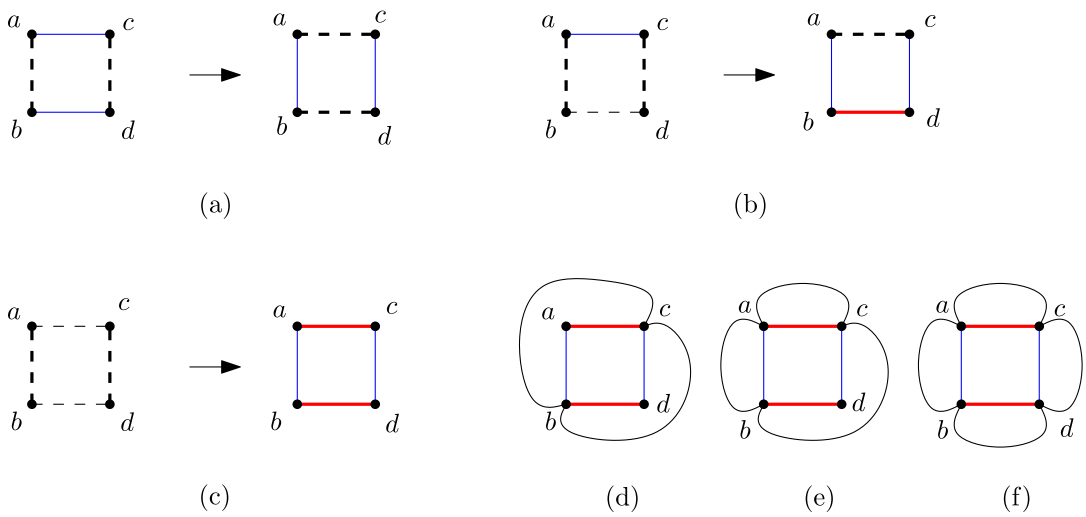

Fig. 4 Case 3 of the lower bound.

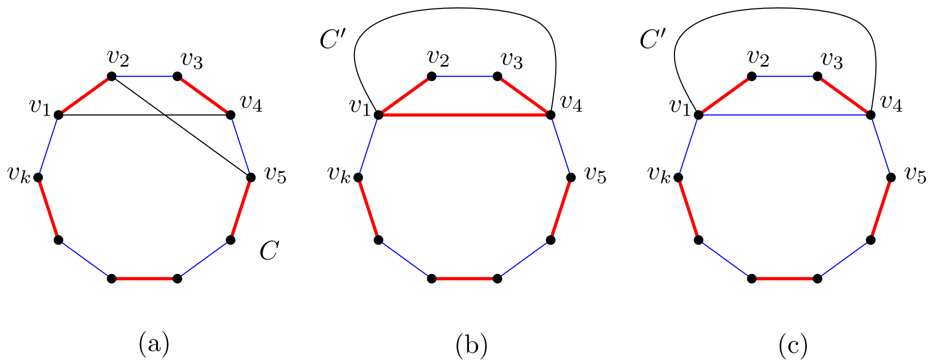

Fig. 5 Lemma 7. (a) $v_1 \neq v_4$ or $v_2 \neq v_5$. (b) (v_1, v_4) is red. (c) (v_1, v_4) is blue.

ticipate in two walks of C', then $p(G, H) \geq p(G', H) - 1$, see Fig. 4 (d). If the edges of *abcda* participate in three cycles of C', then $p(G, H) \geq p(G', H) - 2$, see Fig. 4 (e). If the edges of *abcda* participate in four cycles of C', then $p(G, H) \geq p(G', H) - 3$, see Fig. 4 (f). In all cases $p(G, H) \geq p(G', H) - 3$ and the bound Eq. (1) follows. □

Lemma 7 (Upper Bound) *Let $G = (V, E_G)$ and $H = (V, E_H)$ be two simple graphs such that $d_G(v) = d_H(v)$ for every $v \in V$. There exists a 2-switch in G or H that decreases the value of $\psi(G, H)$ by exactly one.*

Proof: The graph $F(G, H)$ can be partitioned into $p(G, H)$ alternating walks. From all partitions of $F(G, H)$ into $p(G, H)$ alternating walks, we select a partition C such that its shortest walk $C = v_1 v_2 \ldots v_k$ has the least length. We prove Lemma 7 by induction on k.

Suppose that $|C| = 4$. We apply a 2-switch in G replacing edges (v_1, v_2) and (v_3, v_4) with (v_2, v_3) and (v_1, v_4). Let G' be the new graph. Then $r(G', H) = r(G, H) - 2$ and $p(G', H) = p(G, H) - 1$. Thus, $\psi(G', H) = \psi(G, H) - 1$.

Now suppose that $|C| \geq 6$. Then $v_1 \neq v_4$ or $v_2 \neq v_5$ since the edges (v_1, v_2) and (v_4, v_5) have different colors, see **Fig. 5** (a). Without loss of generality we assume that $v_1 \neq v_4$. We consider 4 cases depending on the color of (v_1, v_4).

Suppose that (v_1, v_4) is red. Let C' be a walk in C containing (v_1, v_4). The edges of $C \cup C'$ can be partitioned into two walks so that one walk is $v_1 v_4 v_5 \ldots v_k$, see Fig. 5 (b). This walk is shorter than C. This is a contradiction.

If (v_1, v_4) is blue, then again $C \cup C'$ can be partitioned into two walks so that one walk is 4-cycle $v_1 v_2 v_3 v_4$, see Fig. 5 (c). This is a contradiction again.

If (v_1, v_4) is white, then apply a 2-switch to G replacing edges (v_1, v_2) and (v_3, v_4) with (v_1, v_4) and (v_2, v_3). Let G' be the new graph. Then $r(G', H) = r(G, H) - 1$ and $p(G', H) = p(G, H)$, see **Fig. 6** (a). Then $\psi(G', H) = \psi(G, H) - 1$ by induction hypothesis.

If (v_1, v_4) is black, then apply a 2-switch to H replacing edges (v_1, v_2) and (v_3, v_4) with (v_1, v_4) and (v_2, v_3). Let H' be the new graph. Then $r(G, H') = r(G, H) - 1$ and $p(G, H') = p(G, H)$, see Fig. 6 (b). Then $\psi(G, H') = \psi(G, H) - 1$ by induction hypothesis. The lemma follows. □

Theorem 5 simply follows from the upper and lower bounds.

4. Computing Distance $\psi(G, H)$

A plausible approach to compute a smallest sequence of 2-switches is as follows. Find a largest size decomposition of F into alternating walks and then find a sequence of $\psi(G, H)$ 2-switches transforming G to H. We show that the second step can be done in polynomial time for any (not necessary the largest) decomposition of F. Unfortunately the first step is a NP-hard problem. We develop an approximation algorithm for computing $d(G, H)$.

4.1 Computing 2-switches

We design an algorithm for computing 2-switches based on

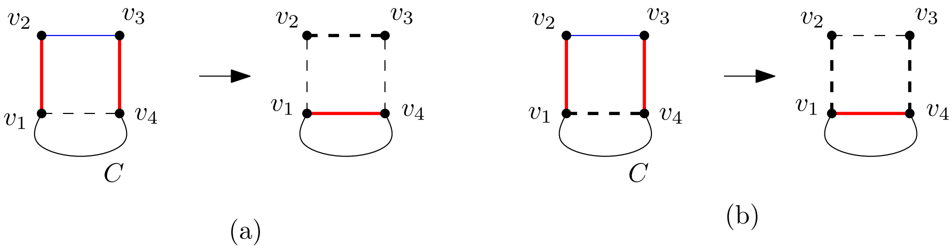

Fig. 6 An illustration of the proof of Lemma 7. (a) Graph $F(G, H)$ with white edge (v_1, v_4) and graph $F(G', H)$. (b) Graph $F(G, H)$ with black edge (v_1, v_4) and graph $F(G, H')$.

a given decomposition C of $F(G, H)$ into red-blue alternating walks. We define $\psi(G, H, C) = r(G, H) - |C|$.

Lemma 8 *Let $G = (V, E_G)$ and $H = (V, E_H)$ be two simple graphs such that $d_G(v) = d_H(v)$ for every $v \in V$. Let C be a decomposition of the edges of $F(G, H)$ into red-blue alternating walks. Then a sequence of $\psi(G, H, C)$ 2-switches transforming G into H can be computed in $O(|V| \cdot |C|)$ time.*

Proof: The algorithm follows the steps of the proof of Lemma 7. However we cannot use the proof directly since C is not assumed to have the largest size. The algorithm repeats the following step until $C = \emptyset$.

Find a shortest walk $C = v_1 v_2 \ldots v_k$ in C. If $|C| = 4$ then apply a 2-switch to the edges of C producing the graph G'. Then $C - C$ is the decomposition of $F(G', H)$ into $|C| - 1$ alternating walks. Since $r(G', H) = r(G, H) - 2$, we have $\psi(G', H, C) = \psi(G, H, C - C) - 1$. The algorithm continues with $C - C$.

Suppose that $|C| \geq 6$. As in the proof of Lemma 7, $v_1 \neq v_4$ or $v_2 \neq v_5$. We consider only the case $v_1 \neq v_4$ since the case $v_2 \neq v_5$ is similar.

Suppose that (v_1, v_4) is red. Find a walk C' in C containing (v_1, v_4). There are two paths $p_1 = v_1 v_4$ and $p_2 = v_1 v_2 v_3 v_4$ from v_1 to v_4. Replace p_1 in C' by p_2 and replace p_2 in C by p_1. The length of walk C decreases by 2. Repeat the main step for C.

If (v_1, v_4) is blue, then apply a 2-switch to $v_1 v_2 v_3 v_4$ producing the graph G'. Two walks C and C' are transformed into one walk. Let C' be the new set of walks. Then $r(G', H) = r(G, H) - 2$ and $\psi(G', H, C') = \psi(G, H, C) - 1$.

If (v_1, v_4) is white, then apply a 2-switch in G replacing edges (v_1, v_2) and (v_3, v_4) with (v_1, v_4) and (v_2, v_3). The walk C is changed as in Fig. 6 (a) producing new set of walks C'. Then $r(G', H) = r(G, H) - 1$ and $|C'| = |C|$. Thus, $\psi(G', H, C') = \psi(G, H, C) - 1$.

If (v_1, v_4) is black, then apply a 2-switch in H replacing edges (v_1, v_4) and (v_2, v_3) with (v_1, v_2) and (v_3, v_4). Again $\psi(G', H, C') = \psi(G, H, C) - 1$, see Fig. 6 (b).

The number of 2-switches computed by the algorithm is equal to $\psi(G, H, C)$. We analyze the implementation details and the running time.

We store the walks in a list. With every walk we store its size. With every edge (u, v) of $G \cup H$, we store its color and a pointer to the corresponding walk if the color is red or blue. Note that a found 2-switch can be applied to either G or H. We store 2-switches in two lists accordingly. When the algorithm finishes we concatenate two lists into one.

One iteration of the algorithm takes a constant time. After each iteration, either a new 2-switch is found or the size of C is reduced. Since the maximum size of C is $|V|$, the lemma follows. □

To compute $\psi(G, H)$, it remains to compute the largest C by Lemma 8. Unfortunately this problem is NP-hard.

4.2 NP-hardness

The decision problem of computing distance $\psi(G, H)$ is as follows.

2-switches problem. Given an integer k and graphs $G = (V, E_G)$ and $H = (V, E_H)$ such that $d_G(v) = d_H(v)$ for every $v \in V$, determine whether G can be transformed to H with at most k 2-switches, i.e., $\psi(G, H) \leq k$.

Caprara [4] proved that the problem of computing the maximum-cardinality decomposition of a balanced graph into alternating walks is NP-hard. By Theorem 5 the 2-switches problem is NP-complete.

4.3 Approximation

The main idea of our approximation algorithm is a reduction to the *maximum independent set* problem where one wants to find a maximum-cardinality independent set of a graph. The problem is known to be NP-hard [7]. Furthermore, for some positive constant $\varepsilon > 0$, finding an approximation of the size of maximum independent set within a factor of n^ε is NP-hard [1]. For some classes of graphs, there exist approximation algorithms with a constant factor [3], [9], [11], [14].

A *k-claw* in an undirected graph is an induced subgraph $K_{1,k}$. A graph is *k-claw free* if no vertex has k distinct independent neighbors. k-claw free graphs admit a polynomial time approximation algorithms.

Theorem 9 (Refs. [9], [11], [14]) *Let $\varepsilon > 0$ and $k \geq 3$ be constants. Let I^* be a maximum independent set in a k-claw free graph G. An independent set I of size bounded by*

$$|I^*| \leq \left(\frac{k-1}{2} + \varepsilon\right)|I|$$

can be computed in polynomial time.

We define a graph F_4 using alternating walks of length four in F: a vertex of F_4 corresponds to an alternating walk of length four in F, two vertices are connected by an edge if the corresponding walks have at least one common edge in F.

Lemma 10 *The graph F_4 is 5-claw free.*

Proof: Suppose that F_4 has a 5-claw with center v. Let C be the corresponding 4-walk in F. There are two neighbors u and w of

v in the 5-claw such that their 4-walks share the same edge of C (otherwise v has at most 4 neighbors in F_4). Then (u, w) is an edge of F_4 which contradicts the definition of 5-claw. □

Theorem 11 *Let $\varepsilon > 0$ be a positive constant and let $G = (V, E_G)$ and $H = (V, E_H)$ be graphs such that $d_G(v) = d_H(v)$ for every $v \in V$. There is a polynomial time algorithm for computing a $(3/2+\varepsilon)$-approximation of the 2-switch distance between G and H.*

Proof: We assume that $\varepsilon < 1/2$.

Algorithm ApproxDistance (ε, G, H)
Given $\varepsilon > 0$ and two graphs G and H, find a sequence of at most $(3/2 + \varepsilon)d(G, H)$ 2-switches transforming G into H.

---

(1) Construct F_4 which is a 5-claw free graph. Set

$$\delta = \frac{2}{1 - 2\varepsilon} - 2. \tag{2}$$

Find an independent set I of size at least $|I^*|/(2 + \delta)$ using Theorem 9 where I^* is a maximum independent set in F_4. Let C_4 be the set of alternating walks corresponding to I.

(2) Remove the walks of C_4 from F and decompose the remaining edges into alternating walks. This decomposition $\mathcal{D}$ can be done arbitrarily. Let $C'_4 = \mathcal{D} \cup C_4$ be the set of all walks, i.e. C'_4 is a decomposition of $F(G, H)$.

(3) Find a list of 2-switches according to C'_4 using the algorithm from Lemma 8. The number of 2-switches is $r(G, H) - |C'_4|$.

Set $r = r(G, H)$. Let P be a maximum walk decomposition of E_F and set $p = |P|$. Let p' be the number of walks produced by our algorithm. By Theorem 5, it suffices to prove that

$$r - p' \leq \left(\frac{3}{2} + \varepsilon\right)(r - p). \tag{3}$$

By Lemma 10, F_4 is 5-claw free graph. By Theorem 9 the independent set of F_4 computed in Step 1 is a $(2+\delta)$-approximation. Set $p'_4 = |C_4|$ and let p^*_4 be the size of largest set of independent 4-walks in F. Let p_4 be the number of 4-walks in P. Then

$$(2 + \delta)p'_4 \geq p^*_4 \geq p_4. \tag{4}$$

There are two groups of walks in P: p_4 walks of length four and $p - p_4$ walks of length at least six. Then

$$r \geq 2p_4 + 3(p - p_4) = 3p - p_4.$$

By Eq. (4)

$$r \geq 3p - (2 + \delta)p'_4 \geq 3p - (2 + \delta)p'. \tag{5}$$

By Eq. (2), $\delta > 0$ (since $\varepsilon < 1/2$) and

$$\varepsilon = \frac{1}{2} - \frac{1}{2 + \delta}.$$

It can be verified that

$$(2 + \delta)(1 + 2\varepsilon) - 2 = (2 + \delta)\left(\frac{3}{2} + \varepsilon\right) - 3. \tag{6}$$

Since every walk in P contains at least two red edges, we have

$$r \geq 2p. \tag{7}$$

Multiplying left hand sides and right hand sides of Eqs. (6) and (7) (and dividing by two), we obtain

$$\left((2 + \delta)\left(\frac{1}{2} + \varepsilon\right) - 1\right) r \geq \left((2 + \delta)\left(\frac{3}{2} + \varepsilon\right) - 3\right) p.$$

It can be written as

$$3p - r \geq (2 + \delta)\left(\left(\frac{3}{2} + \varepsilon\right) p - \left(\frac{1}{2} + \varepsilon\right) r\right). \tag{8}$$

By Eq. (5) we have $(2 + \delta)p' \geq 3p - r$. Combining it with Eq. (8) we get

$$p' \geq \left(\frac{3}{2} + \varepsilon\right) p - \left(\frac{1}{2} + \varepsilon\right) r$$

which implies Eq. (3). □

5. Conclusions

It is well known that, for any two graphs with the same degree sequence, one graph can be transformed into the other graph by a sequence of 2-switches. We studied the problem of finding the minimum number of 2-switches for transforming one graph into the other. Our main result is Theorem 5 that provides a formula for the number of 2-switches. Since the problem of computing the cycle decomposition is NP-hard, we design an approximation algorithm with the approximation factor close to 1.5. An interesting open problem is to improve the approximation guarantee. One can explore a recent development in an approximation of the k-set packing [5], [12].

Acknowledgments We thank CREST, JST, Grant Number JPMJCR1402 and JSPS KAKENHI Grant Numbers 15K11985 through which this work was partially supported.

References

[1] Arora, S., Lund, C., Motwani, R., Sudan, M. and Szegedy, M.: Proof verification and the hardness of approximation problems, *J. ACM*, Vol.45, No.3, pp.501–555 (1998).
[2] Berge, C.: *Graphes et hypergraphes*, Number 37 in Monographies Universitaires de Mathématiques, Dunod, Paris (1970).
[3] Berman, P.: A $d/2$ approximation for maximum weight independent set in d-claw free graphs, *Proc. 7th Scand. Workshop Algorithm Theory*, pp.214–219 (2000).
[4] Caprara, A.: Sorting permutations by reversals and Eulerian cycle decompositions, *SIAM Journal on Discrete Mathematics*, Vol.12, No.1, pp.91–110 (1999).
[5] Cygan, M.: Improved approximation for 3-dimensional matching via bounded pathwidth local search, *Proc. 54th Annu. IEEE Sympos. Found. Comput. Sci., FOCS*, pp.509–518 (2013).
[6] Erdös, P. and Gallai, T.: Graphs with prescribed degrees of vertices, *Mat. Lapok*, Vol.11, pp.264–274 (1960).
[7] Garey, M.R. and Johnson, D.S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman, New York, NY (1979).
[8] Hakimi, S.L.: On the realizability of a set of integers as degrees of the vertices of a graph, *SIAM Journal on Applied Mathematics*, Vol.10, pp.496–506 (1962).
[9] Halldórsson, M.M.: Approximating discrete collections via local improvements, *Proc. 6th ACM-SIAM Sympos. Discrete Algorithms*, pp.160–169 (1995).
[10] Havel, V.: A remark on the existence of finite graphs, *Časopis pro Pěstováni Matematiky*, Vol.80, pp.477–480, (1955). [in Czech].
[11] Hurkens, C.A.J. and Schrijver, A.: On the size of systems of sets every t of which have an SDR, with an application to the worst-case ratio of heuristics for packing problems, *SIAM Journal on Discrete Mathematics*, Vol.2, No.1, pp.68–72 (1989).
[12] Sviridenko, M. and Ward, J.: Large neighborhood local search for the maximum set packing problem, *Proc. 40th Internat. Colloq. Automata Lang. Prog., Part I*, pp.792–803 (2013).

[13] West, D.: *An Introduction to Graph Theory*, Prentice Hall (1995).
[14] Yu, G. and Goldschmidt, O.: Local optimality and its applications on independent sets for k-claw free graphs, *Journal of Combinatorial Optimization*, Vol.1, No.2, pp.151–164 (1997).

Sergey Bereg received M.S. in Computer Science from Ural State University in 1985 and Ph.D. in Computer Science from Minsk Institute of Mathematics in 1992. He is currently an Associate Professor at the University of Texas at Dallas. His research interests are in the foundations of computer science, in particular Computational Geometry, Computational Biology and Coding Theory. Dr. Bereg is a member of ACM.

Hiro Ito received B.E., M.E., and Ph.D. degrees in the Department of Applied Mathematics and Physics from the Faculty of Engineering, Kyoto University in 1985, 1987, and 1995, respectively. 1987-1996, 1996-2001, and 2001-2012, he was a member of NTT Laboratories, Toyohashi University of Technology, and Kyoto University, respectively. Since 2012, he has been a Full Professor in School of Informatics and Engineering at The University of Electro-Communications (UEC). He has been engaged in research on discrete algorithms mainly on graphs and networks, discrete mathematics, recreational mathematics, and algorithms for big data. Dr. Ito is a member of IEICE, the Operations Research Society of Japan, IPSJ, and the European Association for Theoretical Computer Science.