Swapping Colored Tokens on Graphs

Katsuhisa Yamanaka[1], Takashi Horiyama[2], David Kirkpatrick[3], Yota Otachi[4], Toshiki Saitoh[5], Ryuhei Uehara[4], and Yushi Uno[6]

[1] Iwate University, Japan. yamanaka@cis.iwate-u.ac.jp
[2] Saitama University, Japan. horiyama@al.ics.saitama-u.ac.jp
[3] University of British Columbia, Canada. kirk@cs.ubc.ca
[4] Japan Advanced Institute of Science and Technology, Japan. otachi@jaist.ac.jp, uehara@jaist.ac.jp
[5] Kobe University, Japan. saitoh@eedept.kobe-u.ac.jp
[6] Osaka Prefecture University, Japan. uno@mi.s.osakafu-u.ac.jp

Abstract. We investigate the computational complexity of the following problem. We are given a graph in which each vertex has the current and target colors. Each pair of adjacent vertices can swap their current colors. Our goal is to perform as small numbers of swappings as possible so that the current and target colors agree at each vertex. When the colors are chosen from $\{1, 2, \dots, c\}$, we call this problem c-Colored Token Swapping since the current color of a vertex can be seen as a colored token placed on the vertex. We show that c-Colored Token Swapping is NP-complete for every constant $c \geq 3$ even if input graphs are restricted to connected planar bipartite graphs of maximum degree 3. We then show that 2-Colored Token Swapping can be solved in polynomial time for general graphs and in linear time for trees.

1 Introduction

Sorting problems are fundamental and important in computer science. In this paper, we consider a problem of sorting on graphs. Let $G = (V, E)$ be an undirected unweighted graph with vertex set V and edge set E. Suppose that each vertex in G has a color in $C = \{1, 2, \dots, c\}$. A token is placed on each vertex in G, and each token also has a color in C. Then, we wish to transform the current token-placement into the one such that a token of color i is placed on a vertex of color i for all vertices by swapping tokens on adjacent vertices in G. See Fig. 1 for an example. If there exists a color i such that the number of vertices of color i is not equal to the number of tokens of color i in the current token-placement, then we cannot transform the current token-placement into the target one. Thus, without loss of generality, we assume that the number of vertices of color i for each $i = 1, 2, \dots, c$ is equal to the number of tokens of the same color. As we will see in the next section, any token-placement can be transformed into the target one by $\mathrm{O}(n^2)$ token-swappings, where n is the number of vertices in G. We thus consider the problem of minimizing the number of token-swappings to obtain the target token-placement.

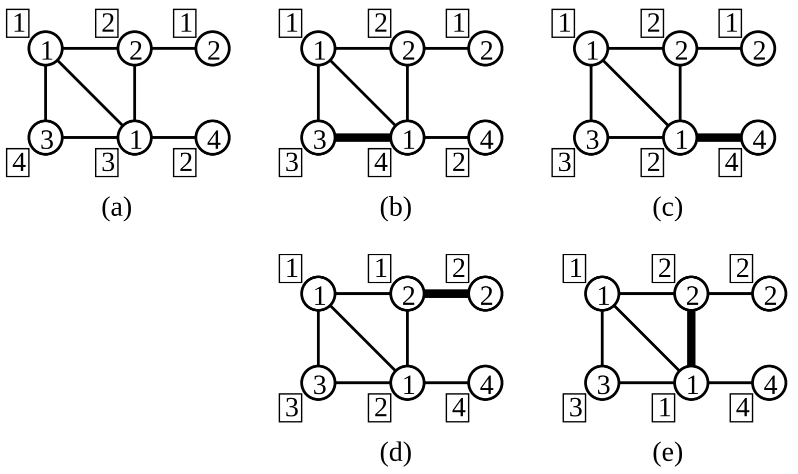

Fig. 1. An example of 4-Colored Token Swapping. Colors of vertices are written inside circles and tokens are drawn as rectangles with their colors. We swap the two tokens along each thick edge. (a) An initial token-placement. (b)–(d) Intermediate token-placements. (e) The target token-placement.

If vertices have distinct colors and tokens also have distinct colors, then the problem is called Token Swapping [11]. This has been investigated for several graph classes. Token Swapping can be solved in polynomial time for paths [7, 8], cycles [7], stars [10], complete graphs [1, 7], and complete bipartite graphs [11]. Heath and Vergara [6] gave a polynomial-time 2-approximation algorithm for squares of paths, where the square of a path is the graph obtained from the path by adding a new edge between two vertices with distance exactly two in the path. For squares of paths, some upper bounds of the minimum number of token-swappings are known [3, 4, 6]. Yamanaka et al. [11] gave a polynomial-time 2-approximation algorithm for trees. Token Swapping is solved for only restricted graph classes. However no hardness result is known, even if input graphs are general graphs, to the best of our knowledge.

The c-Colored Token Swapping problem is a generalization of Token Swapping. We investigate c-Colored Token Swapping and clarify its computational complexity in the sense that we found the boundary of easy and hard cases with respect to the number of colors. For $c = 2$, the problem can be solved in polynomial time for general graphs and in linear time for trees. However, the problem for $c = 3$ is hard even if input graphs are quite restricted. We show that the problem is NP-complete for connected planar bipartite graphs with maximum degree 3.

2 Preliminaries

In this paper, we assume without loss of generality that graphs are simple and connected. Let $G = (V, E)$ be an undirected unweighted graph with vertex set V and edge set E. We sometimes denote by $V(G)$ and $E(G)$ the vertex set and the edge set of G, respectively. We always denote $|V|$ by n. For a vertex v in G, let $N(v)$ be the set of all neighbors of v. Each vertex of a graph $G = (V, E)$ has a color in $C = \{1, 2, \ldots, c\}$. We denote by $c(v)$ the color of a vertex $v \in V$. A token is placed on each vertex in G, and each token also has a color in C. For a vertex v, we denote by $f(v)$ the color of the token placed on v. Then, we call the function $f : V \to C$ a *token-placement* of G. Two token-placements f and f' of G are said to be *adjacent* if the following two conditions (a) and (b) hold:

(a) there exists exactly one edge $(u, v) \in E$ such that $f'(u) = f(v)$ and $f'(v) = f(u)$; and

(b) $f'(w) = f(w)$ for all vertices $w \in V \setminus \{u, v\}$.

In other words, the token-placement f' is obtained from f by *swapping* the tokens on the two adjacent vertices u and v. Note that swapping two tokens of the same color gives the same token-placement. Thus, to eliminate redundancy, we assume that tokens of the same color are never swapped. For two token-placements f and f' of G, a sequence $\mathcal{S} = \langle f_0, f_1, \ldots, f_h \rangle$ of token-placements is a *swapping sequence* between f and f' if the following three conditions (1)–(3) hold:

(1) $f_0 = f$ and $f_h = f'$;

(2) f_k is a token-placement of G for each $k = 0, 1, \ldots, h$; and

(3) f_{k-1} and f_k are adjacent for every $k = 1, 2, \ldots, h$.

The *length* of a swapping sequence $\mathcal{S}$ is defined to be the number of token-placements in $\mathcal{S}$, and denoted by $\mathrm{len}(\mathcal{S})$. For two token-placements f and f' of G, we denote by $\mathrm{OPT}(f, f')$ the minimum length of a swapping sequence between f and f'. As we will prove in Lemma 1, there always exists a swapping sequence between any two token-placements f and f' if the number of vertices of color i for each $i = 1, 2, \ldots, c$ is equal to the number of tokens of the same color. For the two token-placement f and f', $\mathrm{OPT}(f, f')$ is well-defined.

Given two token-placements f and f' of a graph G and a nonnegative integer ℓ, the c-Colored Token Swapping problem is to determine whether or not $\mathrm{OPT}(f, f') \le \ell$ holds. From now on, we always denote by f and f' the *initial* and *target* token-placements of G, respectively, and we may assume without loss of generality that f' is a token-placement of G such that $f'(v) = c(v)$ for all vertices $v \in V$.

We show that the length of any swapping sequence need never exceed n^2. This claim is derived by slightly modifying the proof of Theorem 1 in [11].

Lemma 1. *For any pair of token-placements f and f' of a graph G, $OPT(f, f') \le n^2$.*

Proof. Let T be any spanning tree of a graph $G = (V, E)$. Choose an arbitrary leaf v of T. Then, we move a nearest token of color $c(v)$ in T from the current position u to its target position v. Note that there is no token of color $c(v)$ placed on a vertex of the path in T from u to v except u. Let $(p_1, p_2, \ldots, p_q)$ be a unique path in T from $p_1 = u$ to $p_q = v$. Then, we swap the tokens on p_k and p_{k+1} for each $k = 1, 2, \ldots, q-1$ in this order, and obtain the token-placement f of G such that $f(v) = c(v)$. We then delete the vertex v from G and T, and repeat the process until we obtain f'.

Each vertex obtains a token of the same color via a swapping sub-sequence of length in n. Therefore, the swapping sequence $\mathcal{S}$ above between f and f' satisfies $\mathrm{len}(\mathcal{S}) \leq n^2$. Since $\mathrm{OPT}(f, f') \leq \mathrm{len}(\mathcal{S})$, we have $\mathrm{OPT}(f, f') \leq n^2$. $\square$

From Lemma 1, any token-placement for an input graph can be transformed into the target one by $\mathrm{O}(n^2)$ token-swappings, and a swapping sequence of length $\mathrm{O}(n^2)$ can be computed in polynomimal time.

3 Hardness results

In this section, we show that c-Colored Token Swapping problem is NP-complete for any constant $c \geq 3$ by constructing a polynomial-time reduction from Planar 3DM [2]. To define Planar 3DM, we first introduce the following well-known NP-complete problem.

Problem: 3-Dimensional Matching (3DM) [5, SP1]
Instance: Set $T \subseteq X \times Y \times Z$, where X, Y, and Z are disjoint sets having the same number m of elements.
Question: Does T contain a matching, i.e., a subset $T' \subseteq T$ such that $|T'| = m$ and it contains all elements of X, Y, and Z?

Planar 3DM is a restricted version of 3DM in which the following bipartite graph G is planar. The graph G has the vertex set $V(G) = T \cup X \cup Y \cup Z$ with a bipartition $(T, X \cup Y \cup Z)$. Two vertices $t \in T$ and $w \in X \cup Y \cup Z$ are adjacent in G if and only if $w \in t$. Planar 3DM is NP-complete even if G is a connected graph of maximum degree 3 [2].

Theorem 1. 3-Colored Token Swapping *is NP-complete even for connected planar bipartite graphs of maximum degree 3.*

Proof. By Lemma 1, there is a polynomial-length swapping sequence for any initial token-placement, and thus 3-Colored Token Swapping is in NP.

Now we present a reduction from Planar 3DM. Let $(X, Y, Z; T)$ be an instance of Planar 3DM and $m = |X| = |Y| = |Z|$. As mentioned above, we construct a bipartite graph $G = (T, X \cup Y \cup Z; E)$ from $(X, Y, Z; T)$. We set $c(x) = 1$ and $f(x) = 2$ for every $x \in X$, set $c(y) = 2$ and $f(y) = 3$ for every $y \in Y$, set $c(z) = 3$ and $f(z) = 1$ for every $z \in Z$, and set $c(t) = 1$ and $f(t) = 1$ for every $t \in T$. See Fig. 2. From the assumptions, G is a planar bipartite graph of maximum degree 3. The reduction can be done in polynomial time. We prove that the instance $(X, Y, Z; T)$ is a yes-instance if and only if $\mathrm{OPT}(f, f') \leq 3m$.

To show the only-if part, assume that there exists a subset T' of T such that $|T'| = m$ and T' contains all elements of X, Y, and Z. Since the elements of

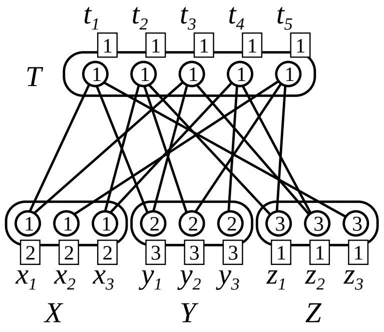

Fig. 2. The graph constructed from an instance ($X = \{x_1, x_2, x_3\}$, $Y = \{y_1, y_2, y_3\}$, $Z = \{z_1, z_2, z_3\}$, $T = \{t_1 = (x_1, y_1, z_3)$, $t_2 = (x_3, y_2, z_1)$, $t_3 = (x_1, y_1, z_2)$, $t_4 = (x_3, y_3, z_2)$, $t_5 = (x_2, y_2, z_1)\}$).

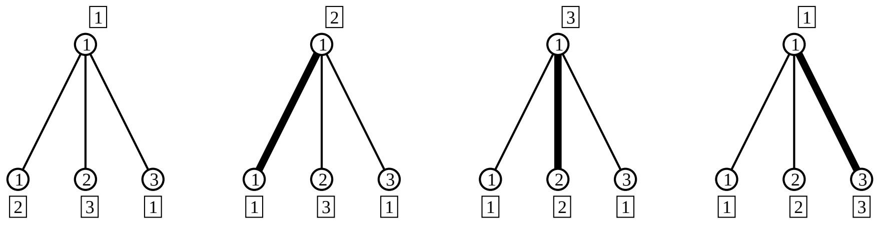

Fig. 3. A swapping sequence to resolve the token-placement of a triple.

T' are pairwise disjoint, we can cover the graph G' with m disjoint stars of four vertices, where each star is induced by an element t of T' and its three elements. To locally move the tokens on the target place in such a star, we need only three swappings. See Fig. 3. This implies that a swapping sequence of length $3m$ exists.

To show the if part, assume that there is a swapping sequence $\mathcal{S}$ from f to f' with at most $3m$ token-swappings. Let $T' \subseteq T$ be the set of vertices such that the tokens on them are moved in $\mathcal{S}$. Let G' be the subgraph of G induced by $T' \cup X \cup Y \cup Z$. Let $w \in X \cup Y \cup Z$. Since $c(w) \neq f(w)$ and $N(w) \subseteq T$, the sequence $\mathcal{S}$ swaps the tokens on w and on a neighbor $t \in T'$ of w at least once. This implies that w has degree at least 1 in G'. Since each $t \in T'$ has degree at most 3 in G', we can conclude that $|T'| \geq \frac{1}{3} |X \cup Y \cup Z| = m$. In $\mathcal{S}$, the token placed on a vertex in $X \cup Y$ in the initial token-placement is moved at least twice, while the token placed on a vertex in $Z \cup T'$ is moved at least once. As a token-swapping moves two tokens at the same time,

$$\mathrm{len}(\mathcal{S}) \geq \frac{1}{2}(2\,|X| + 2\,|Y| + |Z| + |T'|) \geq 3m.$$

From the assumption that $\mathrm{len}(\mathcal{S}) \leq 3m$, it follows that $|T'| = m$, and hence each $w \in X \cup Y \cup Z$ has degree exactly 1 in G'. Therefore, G' consists of m disjoint stars centered at the vertices of T' which form a solution of Planar 3DM. $\square$

The proof above can be extended for any constant number of colors. It is known that we can assume that G has a degree-2 vertex [2]. We add a path $(p_4, p_5, \ldots, p_c)$ to G, and connect p_4 to a degree-2 vertex in G. We set $c(p_i) = i$ and $f(p_i) = i$. The proof still works for the new graph, and hence we obtain the following corollary.

Corollary 1. *For every constant $c \geq 3$, c-Colored Token Swapping is NP-complete even for connected planar bipartite graphs of maximum degree 3.*

Note that the degree bound in the corollary above is tight. If a graph has maximum degree 2, then we can solve c-Colored Token Swapping in polynomial time for every constant c as follows. A graph of maximum degree 2 consists of disjoint paths and cycles. Observe that a shortest swapping sequence does not swap tokens of the same color. This immediately gives a unique matching between tokens and target vertices for a path component. For a cycle component, observe that each color class has at most n candidates for such a matching restricted to the color class. This is because after we guess the target of a token in a color class, the targets of the other tokens in the color class can be uniquely determined. In total, there are at most n^c matchings between tokens and target vertices. By guessing such a matching, we can reduce c-Colored Token Swapping to Token Swapping. Now we can apply Jerrum's $\mathrm{O}(n^2)$-time algorithms for solving Token Swapping on paths and cycles [7]. Therefore, we can solve c-Colored Token Swapping in $\mathrm{O}(n^{c+2})$ time for graphs of maximum degree 2.

4 Polynomial-time algorithms

In this section, we give some positive results. We first show that 2-Colored Token Swapping for general graphs can be solved in polynomial time. We next show that 2-Colored Token Swapping problem for trees can be solved in linear time without constructing a swapping sequence.

4.1 General graphs

Let $C = \{1, 2\}$ be the color set. Let $G = (V, E)$ be a graph, and let f and f' be an initial token-placement and the target token-placement. We construct a weighted complete bipartite graph $G_B = (X, Y, E_B, w)$, as follows. The vertex sets X, Y and the edge set E_B are defined as follows:

$$\begin{aligned} X &= \{x_v \mid v \in V \text{ and } f(v) = 1\} \\ Y &= \{y_v \mid v \in V \text{ and } c(v) = 1\} \\ E_B &= \{(x, y) \mid x \in X \text{ and } y \in Y\}. \end{aligned}$$

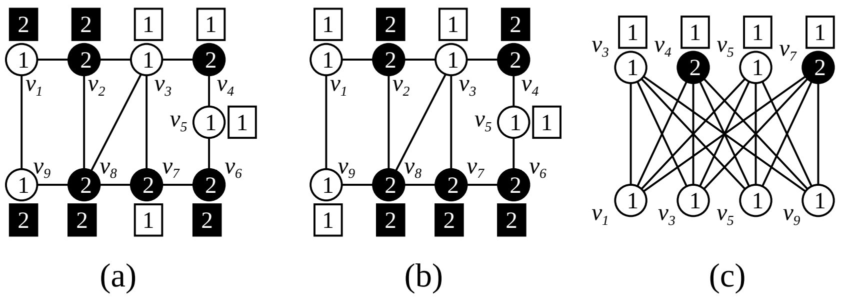

Fig. 4. (a) An initial token-placement. (b) The target token-placement. (c) The weighted complete bipartite graph constructed from (a) and (b) (the weight of each edge is omitted).

Intuitively, X is the copies of vertices in V having tokens of color 1, and Y is the copies of vertices in V of color 1. The weight function w is a mapping from E_B to positive integers. For $x \in X$ and $y \in Y$, the weight $w(e)$ of the edge $e = (x, y)$ is defined as the length of a shortest path from x to y in G. Fig. 4 gives an example of an initial token-placement, the target token-placement, and the associated weighted complete bipartite graph.

We bound $\mathrm{OPT}(f, f')$ from below, as follows. Let $\mathcal{S}$ be a swapping sequence between f and f'. The swapping sequence gives a perfect matching of G_B, as follows. For each token of color 1, we choose an edge (x, y) of G_B if the token is placed on $x \in X$ in f and on $y \in Y$ in f'. The obtained set is a perfect matching of G_B. A token corresponding to an edge e in the matching needs $w(e)$ token-swappings, and two tokens of color 1 are never swapped in $\mathcal{S}$. Therefore, for a minimum weight matching M of G_B, we have the following lower bound:

$$\mathrm{OPT}(f, f') \geq \sum_{e \in M} w(e).$$

Now we describe our algorithm. First we find a minimum weight perfect matching M of G_B. We choose an edge e in M. Let $P_e = \langle p_1, p_2, \ldots, p_q \rangle$ of G be a shortest path corresponding to e. We have the following lemma.

Lemma 2. *Suppose that the two tokens on endpoints of P_e have different colors. The two tokens can be swapped by $w(e)$ token-swappings such that the color of the token on each internal vertex does not change.*

Proof. Without loss of generality, we assume that $f(p_1) = 2$ and $f(p_q) = 1$ hold. We first choose the minimum i such that $f(p_i) = 1$ holds. We next move the token on p_i to p_1 by $i - 1$ token-swappings. We repeat the same process to the subpath $\langle p_i, p_{i+1}, \ldots, p_q \rangle$. Finally, we obtain the desired token-placement. Recall that there are only two colors on graphs, and so the above "color shift" operation works. Since each edge of P_e is used by one token-swapping, the total number of token-swapping is $w(e) = q - 1$. $\square$

This lemma permits to move the two tokens on the two endpoints p_1 and p_q of P_e to their target positions in $w(e)$ token-swappings. Let g be the token-placement obtained after the token-swappings. We can observe that $f(v) = g(v)$ for every $v \in V \setminus \{p_1, p_q\}$ and $g(v) = c(v)$ for $v \in \{p_1, p_q\}$. Then we remove e from the matching M. We repeat the same process until M becomes empty. Our algorithm always exchanges tokens on two vertices using a shortest path between the vertices. Hence, the length of the swapping sequence constructed by our algorithm is equal to the lower bound.

Now we estimate the running time of our algorithm. The algorithm first constructs the weighted complete bipartite graph. This can be done using Floyd-Warshall algorithm in $\mathrm{O}(n^3)$ time. Then, our algorithm constructs a minimum weight perfect matching. This can be done in $\mathrm{O}(n^3)$ time [9, p.252]. Finally, for each of the $\mathrm{O}(n)$ paths in the matching, our algorithm moves the tokens on the endpoints of the path in linear time. We have the following theorem.

Theorem 2. 2-Colored Token Swapping *is solvable in $O(n^3)$ time. Furthermore, a swapping sequence of the minimum length can be constructed in the same running time.*

4.2 Trees

In this subsection, we show that 2-Colored Token Swapping for trees can be solved in linear time without constructing a swapping sequence.

Let T be an input tree, and let f and f' be an initial token-placement and the target token-placement of T. Let $e = (x, y)$ be an edge of T. Removal of e disconnects T into the two subtrees. We denote by $T(x)$ the subtree containing x and denote by $T(y)$ the subtree containing y.

Now we define the value $\mathrm{diff}(e)$ for each edge e of T. Intuitively, $\mathrm{diff}(e)$ is the number of tokens of color 1 which we wish to move from $T(x)$ to $T(y)$ along e. More formally, we give the definition of $\mathrm{diff}(e)$, as follows. Let n^1_t be the number of tokens of color 1 in $T(x)$, and let n^1_v be the number of vertices of color 1 in $T(x)$. Then, we define $\mathrm{diff}(e) = \left|n^1_t - n^1_v\right|$. (Note that, even if we count tokens and vertices of color 1 in $T(y)$ instead of $T(x)$, the value $\mathrm{diff}(e)$ takes the same value.) See Fig. 5 for an example. For each edge e of T, we need to move at least $\mathrm{diff}(e)$ tokens of color 1 from a subtree to the other one along e. Therefore, $\mathrm{OPT}(f, f')$ is lower bounded by the sum $D = \sum_{e \in E(T)} \mathrm{diff}(e)$.

To give an upper bound of $\mathrm{OPT}(f, f')$, we next show that there exists a swapping sequence of length D. The following lemma is key to construct the swapping sequence.

Lemma 3. *If $D \neq 0$, then there exists an edge e such that the token-swapping on e decreases D by one.*

Proof. We first give an orientation of edges of T. For each edge $e = (x, y)$, we orient e from x to y if the number of tokens of color 1 in $T(x)$ is greater than the number of vertices of color 1 in $T(x)$. Intuitively, the direction of an edge means that we need to move one or more tokens of color 1 from $T(x)$ to $T(y)$. If

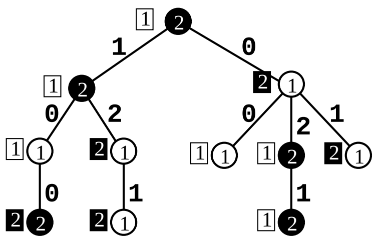

Fig. 5. An example of diff(e). For each edge e, the value diff(e) is written beside e.

the two numbers are equal, we remove e from T. Let T' be the obtained directed tree. For an edge $e = (x, y)$ oriented from x to y, if x has a token of color 1 and y has a token of color 2, swapping the two token decreases D by one. We call such an edge a *desired* edge. We now show that there exists a desired edge in T'. Observe that if no vertex u with $f(u) = 1$ is incident to a directed edge in T', then indeed T' has no edge and $D = 0$. Let u be a vertex with $f(u) = 1$ that has at least one incident edge in T'. If u has no out-going edge, then the number of the color-1 tokens in T exceeds the number of the color-1 vertices in T. Thus we can choose an edge (u, v) oriented from u to v. If $f(v) = 2$ holds, the edge is desired. Now we assume that $f(v) = 1$ holds. We apply the same process for v, then an edge (v, w) oriented from v to w can be found. Since trees have no cycle, by repeating the process, we always find a desired edge. □

From Lemma 3, we can find a desired edge, and we swap the two tokens on the endpoints of the edge. Since a token-swapping on a desired edge decreases D by one, by repeatedly swapping on desired edges, we obtain the swapping sequence of length D. Note that $D = 0$ if and only if $c(v) = f(v)$ for every $v \in V(T)$. Hence, we have $\mathrm{OPT}(f, f') \leq D$.

Therefore $\mathrm{OPT}(f, f') = D$ holds, and so we can solve 2-Colored Token Swapping by calculating D. The value diff(e) for every edge e, and thus the value D, can be calculated in a bottom-up manner in linear time in total. We have the following theorem.

Theorem 3. 2-Colored Token Swapping *is solvable in linear time for trees.*

5 Conclusions

We have investigated computational complexity of c-Colored Token Swapping. We first showed the NP-completeness for 3-Colored Token Swapping by a reduction from Planar 3DM, even for connected planar bipartite graphs of maximum degree 3. We next showed that 2-Colored Token Swapping can be solved in $\mathrm{O}(n^3)$ time for general graphs and in linear time for trees.

We showed that c-Colored Token Swapping for every constant c can be solved in polynomial time for graphs of maximum degree 2 (disjoint paths and cycles). If c is not a constant, can we solve c-Colored Token Swapping for such graphs in polynomial time? For Token Swapping on cycles, Jerrum [7] proposed an $O(n^2)$-time algorithm. As mentioned in [7], the proof of the correctness of the algorithm needs complex discussions.

A standard dynamic programming technique based on our algorithm for trees may give a polynomial-time (but not necessarily fixed-parameter tractable) algorithm on graphs of bounded treewidth for c-Colored Token Swapping. Designing a fixed-parameter tractable algorithm parameterized by treewidth would be interesting future work.

References

1. A. Cayley. Note on the theory of permutations. *Philosophical Magazine*, 34:527–529, 1849.
2. M.E. Dyer and A.M. Frieze. Planar 3DM is NP-complete. *Journal of Algorithms*, 7:174–184, 1986.
3. X. Feng, Z. Meng, and I.H. Sudborough. Improved upper bound for sorting by short swaps. In *Proc. IEEE Symposium on Parallel Architectures, Algorithms and Networks*, pages 98–103, 2004.
4. X. Feng, I.H. Sudborough, and E. Lu. A fast algorithm for sorting by short swap. In *Proc. IASTED International Conference Computational and Systems Biology*, pages 62–67, 2006.
5. M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, 1979.
6. L.S. Heath and J.P.C. Vergara. Sorting by short swaps. *Journal of Computational Biology*, 10(5):775–789, 2003.
7. M.R. Jerrum. The complexity of finding minimum-length generator sequence. *Theoretical Computer Science*, 36:265–289, 1985.
8. D.E. Knuth. *The art of computer programming*, volume 3. Addison-Wesley, 2nd edition, 1998.
9. B. Korte and J. Vygen. *Combinatorial Optimization: Theory and Algorithms.* Springer, 2nd edition, 2005.
10. I. Pak. Reduced decompositions of permutations in terms of star transpositions, generalized catalan numbers and k-ary trees. *Discrete Mathematics*, 204:329–335, 1999.
11. K. Yamanaka, E.D. Demaine, T. Ito, J. Kawahara, M. Kiyomi, Y. Okamoto, T. Saitoh, A. Suzuki, K. Uchizawa, and T. Uno. Swapping labeled tokens on graphs. *Theoretical Computer Science*, 2015. accepted.