Theoretical Computer Science 729 (2018) 1–10

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Swapping colored tokens on graphs[☆]

Katsuhisa Yamanaka[a,*], Takashi Horiyama[b], J. Mark Keil[c], David Kirkpatrick[d], Yota Otachi[e], Toshiki Saitoh[f], Ryuhei Uehara[g], Yushi Uno[h]

[a] Iwate University, Japan
[b] Saitama University, Japan
[c] University of Saskatchewan, Canada
[d] University of British Columbia, Canada
[e] Kumamoto University, Japan
[f] Kyushu Institute of Technology, Japan
[g] Japan Advanced Institute of Science and Technology, Japan
[h] Osaka Prefecture University, Japan

A R T I C L E   I N F O

Article history:
Received 17 November 2016
Received in revised form 2 March 2018
Accepted 12 March 2018
Available online 19 March 2018
Communicated by A. Marchetti-Spaccamela

Keywords:
Computational complexity
NP-completeness
Fixed-parameter algorithm
Token swapping
Colored token swapping

A B S T R A C T

We investigate the computational complexity of the following problem. We are given a graph in which each vertex has an initial and a target color. Each pair of adjacent vertices can swap their current colors. Our goal is to perform the minimum number of swaps so that the current and target colors agree at each vertex. When the colors are chosen from $\{1, 2, \ldots, c\}$, 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 $c = 3$ 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. Besides, we show that, the problem for complete graphs is fixed-parameter tractable when parameterized by the number of colors, while it is known to be NP-complete when the number of colors is unbounded.

1. Introduction

Sorting problems are fundamental and important in computer science. Let us consider the problem of sorting a given permutation by applying the minimum number of swaps of two elements. If we are allowed to swap only adjacent elements (that is, we can apply only adjacent transpositions), then the minimum number of swaps is equal to the number of inversions of a permutation [12,15]. If we are allowed to swap any two elements, then the minimum number of swaps is equal to the number of elements of a permutation minus the number of cycles of the permutation [2,12]. Now, if we are

[☆] A preliminary version appeared in the proceedings of the 14th International Symposium on Algorithms and Data Structures (WADS 2015), vol. 9214 of Lecture Notes in Computer Science, pp. 619–628, 2015.

* Corresponding author.
E-mail addresses: yamanaka@cis.iwate-u.ac.jp (K. Yamanaka), horiyama@al.ics.saitama-u.ac.jp (T. Horiyama), keil@cs.usask.ca (J.M. Keil), kirk@cs.ubc.ca (D. Kirkpatrick), otachi@cs.kumamoto-u.ac.jp (Y. Otachi), toshikis@ces.kyutech.ac.jp (T. Saitoh), uehara@jaist.ac.jp (R. Uehara), uno@mi.s.osakafu-u.ac.jp (Y. Uno).

https://doi.org/10.1016/j.tcs.2018.03.016

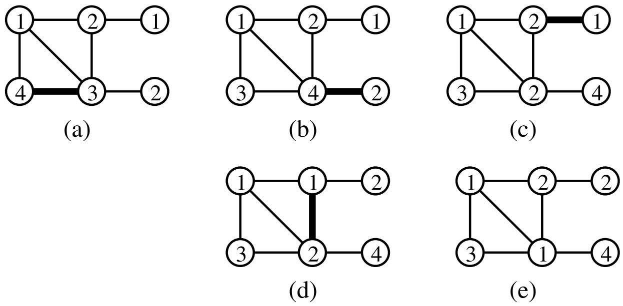

Fig. 1. An instance of 4-Colored Token Swapping. Tokens on vertices are written inside circles. We swap the two tokens along each thick edge. (a) The initial token-placement. (b)–(d) Intermediate token-placements. (e) The target token-placement.

given a set of "allowed swaps", can we exactly estimate the minimum number of swaps? We formalize this question as a problem of swapping tokens 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 token, and each token has a color in $C = \{1, 2, \dots, c\}$. Given two token-placements, we wish to transform one to the other by applying the fewest number of token swaps on adjacent vertices. We call the problem c-Colored Token Swapping if c is a constant, otherwise, we simply call it Colored Token Swapping (a formal definition can be found in the next section). See Fig. 1 for an example.

In this paper, we study the computational complexity of c-Colored Token Swapping. We consider the case where c is a fixed constant and show the following results. If $c = 2$, then the problem can be solved in polynomial time (Theorem 6). On the other hand, 3-Colored Token Swapping is NP-complete even for planar bipartite graphs of maximum degree 3 (Theorem 2). We also show that c-Colored Token Swapping is $\mathcal{O}(n^{c+2})$-time solvable for graphs of maximum degree at most 2 (Theorem 4), 2-Colored Token Swapping is linear-time solvable for trees (Theorem 8), and c-Colored Token Swapping is fixed-parameter tractable for complete graphs if c is the parameter (Theorem 15).

If the tokens have distinct colors, then the problem is called Token Swapping [20]. This variant has been investigated for several graph classes. Token Swapping can be solved in polynomial time for paths [12,15], cycles [12], stars [18], complete graphs [2,12], and complete bipartite graphs [20]. Heath and Vergara [10] gave a polynomial-time 2-approximation algorithm for squares of paths (see also [6,7]). Yamanaka et al. [20] gave a polynomial-time 2-approximation algorithm for trees.

Recently, Miltzow et al. [17] gave computational complexity results for Token Swapping. First, they showed the NP-completeness of Token Swapping. Second, they proposed an exact exponential algorithm for Token Swapping on general graphs and showed that Token Swapping cannot be solved in $2^{o(n)}$ time unless the Exponential Time Hypothesis (ETH) fails, where n is the number of vertices. Third, they proposed polynomial-time 4-approximation algorithms for Token Swapping on general graphs and showed the APX-hardness of Token Swapping. Independently, Kawahara et al. [14] proved the NP-completeness of Token Swapping even if an input graph is bipartite and has only vertices of degree at most 3. More recently, Bonnet et al. [1] gave some results for Token Swapping. They showed the parameterized hardness: Token Swapping is W[1]-hard, parameterized by the number of swaps and Token Swapping cannot be solved in $f(k)(n+m)^{o(k/\log k)}$ unless the ETH fails, where n is the number of vertices, m is the number of edges and k is the number of swaps. Token Swapping on trees is one of the attractive open problems. They gave an interesting result for the open problem: Token Swapping is NP-hard even when both the treewidth and the diameter are constant, and cannot be solved in $2^{o(n)}$ time unless the ETH fails.

Miltzow et al. [17] and Bonnet et al. [1] also gave important results on Colored Token Swapping. Here, we mention only the results related to our contributions. Miltzow et al. [17] gave the NP-completeness of Colored Token Swapping using $\Omega(n)$ colors. On the other hand, in this paper, we show the NP-completeness of c-Colored Token Swapping even if the number of colors is only 3. Bonnet et al. [1] showed the NP-completeness of Colored Token Swapping on complete graphs using $\Omega(n)$ colors. To complement their result, we show the fixed-parameter tractability of c-Colored Token Swapping on complete graphs, parameterized by the number of colors.

2. Preliminaries

The graphs considered in this paper are finite, simple, and undirected. 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.

We formalize our problem as a problem reconfiguring an initial coloring of vertices to the target one by repeatedly swapping the two colors on adjacent vertices as follows. Let $C = \{1, 2, \dots, c\}$ be a set of colors. In this paper, we assume that c is a constant unless otherwise noted. A *token-placement* of G is a surjective function $f : V \to C$. For a vertex v, $f(v)$ represents the color of the token placed on v. Note that we assume that each color in C appears at least once. Two distinct token-placements f and f' of G are *adjacent* if the following two conditions (a) and (b) hold: