Vol.2015-AL-153 No.14
2015/6/13

Swapping Labeled Tokens on Complete Split Graphs

Gaku Yasui[1,a)] Kouta Abe[1,b)] Katsuhisa Yamanaka[1,c)] Takashi Hirayama[1,d)]

Abstract: A token-swapping problem is a kind of generalization of sorting problems. Given a graph $G = (V, E)$ in which each vertex has a token, we wish to move tokens to their target vertices by repeatedly swapping two tokens on adjacent vertices. Recently, Yamanaka et al. have proposed a polynomial-time 2-approximation algorithm for trees and polynomial-time exact algorithm for bipartite complete graphs. In this paper, we give a polynomial-time exact algorithm for complete split graphs.

1. Introduction

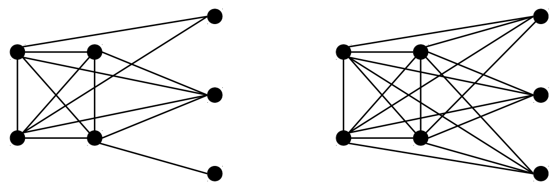
Sorting problems are fundamental and important in computer science. In this paper, we consider a problem of sorting on graphs. Given a simple connected graph $G = (V, E)$ in which each vertex has a labeled token, we wish to move each token to its target vertex by swapping the two tokens on adjacent vertices. We call this a token-swapping problem. The token-swapping problem can be solved in $O(n^2)$ token-swaps for any connected graph [1]. Thus, our objective is to minimize the number of token-swaps.

Some results of the token-swapping problem have been known for several graph classes. For paths, cycles, and complete graphs, the problem can be exactly solved in polynomial time [2]. For square of paths, Heath and Vergara [3] have proposed a polynomial-time 2-approximation algorithm. Recently, Yamanaka et al. [1] have proposed a polynomial-time 2-approximation algorithm for trees and a polynomial-time exact algorithm for bipartite complete graphs.

2. Preliminaries

2.1 Graph notations

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 edge set of G, respectively. We always denote $n = |V|$. For a vertex v in G, let $N(v)$ be the set of all neighbors of v (which does not include v itself), that is, $N(v) = \{w \in V(G) \mid (v, w) \in E(G)\}$. Let $N[v] = N(v) \cup \{v\}$.



(a) (b)

Fig. 1 (a) A split graph and (b) a complete split graph.

A graph is a *split graph* if its vertex set is partitioned into a clique and an independent set. A split graph is a *complete split graph* in which each vertex of its independent set is adjacent to all vertices of its clique. See Fig. 1 for examples.

2.2 Token-swapping problem

Suppose that the vertices in a graph $G = (V, E)$ have distinct labels $v_1, v_2, \ldots, v_n$. Let $L = \{\ell_1, \ell_2, \ldots, \ell_n\}$ be a set of n labeled tokens. Then, a *token-placement* f of G is a mapping $f\colon V \to L$ such that $f(v_i) \neq f(v_j)$ holds for every two distinct vertices $v_i, v_j \in V$; imagine that tokens are placed on the vertices 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 $(v_i, v_j) \in E$ such that $f'(v_i) = f(v_j)$ and $f'(v_j) = f(v_i)$; and

(b) $f'(v_k) = f(v_k)$ for all vertices $v_k \in V \setminus \{v_i, v_j\}$.

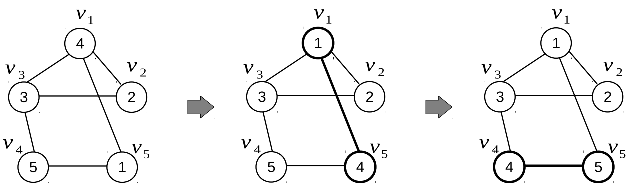
In other words, the token-placement f' is obtained from f by *swapping* the tokens on two vertices v_i and v_j such that $(v_i, v_j) \in E$. For two token-placements f and f' of G, a sequence $\mathcal{S} = \langle f_1, f_2, \ldots, f_h \rangle$ of token-placements is called a *swapping sequence* between f and f' if the following three conditions (1)–(3) hold:

(1) $f_1 = f$ and $f_h = f'$;

(2) f_k is a token-placement of G for each $k = 2, 3, \ldots, h-1$; and

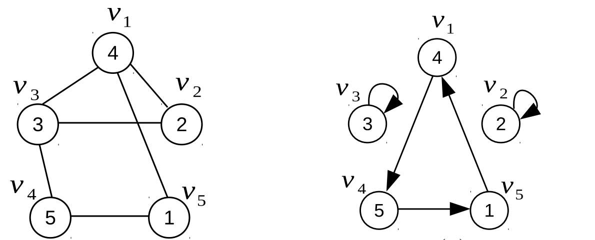
(3) f_{k-1} and f_k are adjacent for every $k = 2, 3, \ldots, h$.

1 Iwate University, Japan
a) yasui@kono.cis.iwate-u.ac.jp
b) k-abe@kono.cis.iwate-u.ac.jp
c) yamanaka@cis.iwate-u.ac.jp
d) hirayama@cis.iwate-u.ac.jp



(a) f_0 (b) (c) f_t

Fig. 2 (a) A given graph and a token-placement f_0. (b) The token-placement obtained by swapping two tokens placed on v_1 and v_5. (c) The token-placement obtained by swapping two tokens placed on v_4 and v_5. This is the target token-placement.



(a) (b)

Fig. 3 (a) A token-placement of a graph, and (b) its conflict graph.

The *length* of a swapping sequence $\mathcal{S}$, denoted by len($\mathcal{S}$), is defined to be the number of token-placements in $\mathcal{S}$ minus one, that is, len($\mathcal{S}$) indicates the number of token-swaps in $\mathcal{S}$. We call a given token-placement an *initial token-placement*, denoted by f_0. The *target token-placement*, denoted by f_t, is the token-placement such that $f_t(v_i) = \ell_i$ for all $i = 1, 2, \ldots, n$. A vertex v_i is a *target vertex* of a token ℓ_j if $\ell_j = f_t(v_i)$ holds. Token-Swapping is the problem of finding the minimum length of a swapping sequence between a given initial token-placement f_0 and a target token-placement f_t. See Fig. 2 for an example. For a graph G and an initial token-placement f_0, $\mathrm{OPT}_G(f_0) = \min\{\mathrm{len}(\mathcal{S}) \mid \mathcal{S}$ is a swapping sequence between f_0 and $f_t\}$.

2.3 Conflict graph

We introduce a digraph $D = (V_D, E_D)$ for a token-placement f of a graph G, called the *conflict graph*, as follows:

- $V_D = V(G)$; and
- there is an arc (v_i, v_j) from v_i to v_j if and only if $f(v_i) = f_t(v_j)$.

Therefore, each token $f(v_i)$ on a vertex $v_i \in V_D$ needs to be moved to the vertex $v_j \in V_D$ such that $(v_i, v_j) \in E_D$. (See Fig. 3 for an example.) A vertex v_i with $f(v_i) = f_t(v_i)$ has a self-loop.

The following lemma holds.

Lemma 1. *[1] Let D be the conflict graph for a token-placement f of a graph G. Then, every component in D is a directed cycle.*

For a token-placement f of a graph, let $C(f)$ be the set of cycles of the conflict graph. Then, we define $V(C(f)) = \bigcup_{C \in C(f)} V(C)$.

Algorithm 1 find-swapping-sequence(G, f_0)

```
G is a complete split graph and f_0 is an initial token-placement
  of G. Let f be the current token-placement of G, and set f = f_0.
for all v ∈ V_Q do
    while f(v) ≠ f_t(v) do
        Swap f(v) with the token on the target vertex of f(v), and
        update f to the token-placement obtained by the token-swap.
    end while
end for
for all v ∈ V_I do
    if f(v) ≠ f_t(v) then
        Swap f(v) with the token on any vertex u in V_Q, and let
        f be the obtained token-placement.
        while f(u) ≠ f_t(u) do
            Swap f(u) with the token on the target vertex of f(u),
            and update f to the token-placement obtained by the
            token-swap.
        end while
    end if
end for
```

3. Upper and lower bounds

In this section, we consider the Token-Swapping problem for complete split graphs. We first give an algorithm that constructs a swapping sequence, and then we estimate the length of the swapping sequence. Next we show that the length of the swapping sequence is optimal.

Let G be a complete split graph. Let V_Q and V_I be sets of vertices of the clique and the independent set of G. It is easily observed that the clique and the independent set of G can be founded in polynomial time. Let f be a token-placement of G, and let D be a conflict graph for f of G. We define $C(f)$ as the set of cycles of D for f. We similarly define $C_Q(f)$ as the set of cycles of D each of which is consisting of only vertices of V_Q, and define $C_I(f)$ as the set of cycles of D each of which is consisting of only vertices of V_I. Denote $C_{QI}(f) = C(f) \setminus (C_Q(f) \cup C_I(f))$. That is, $C_{QI}(f)$ is the set of cycles each of which contains at least one vertex of V_Q and at least one vertex of V_I. Let $C^1(f)$ be the set of cycles with length one. Let $C'_Q(f)$ be the set of cycles in $C_Q(f)$ with length two or more, and let $C'_I(f)$ be the set of cycles in $C_I(f)$ with length two or more.

Now we give an algorithm that finds a swapping sequence between an initial token-placement f_0 and the target token-placement f_t. Our algorithm is shown in **Algorithm 1**. Our algorithm first moves tokens on vertices in V_Q to their target vertices, then moves tokens on vertices in V_I. The details are as follows.

The first for-statement constructs the following token-placement. Let C be any cycle of D including a vertex $v \in V_Q$. Then a token $f(v)$ can be moved to its target vertex by one token-swap, since v is adjacent to all vertices of G. Thus, by swapping the token $f(v)$ on v with the token on the target vertex of $f(v)$, we obtain a cycle with one less length and a cycle with one length. By repeating such token-swaps, we obtain a token-placement f' such that $f'(v) = f_t(v)$ for all $v \in V(C)$ and $f'(v) = f(v)$ for all $v \notin V(C)$. We perform the above process for $v \in V_Q$ such that $f(v) \neq f_t(v)$ holds. Let g be the obtained token-placement. Then we have the following lemma.

Lemma 2. *For the token-placement g, we have*

- $g(v) = f_t(v)$ *if* $v \in V_Q \cup \bigcup_{C \in C_{QI}(f_0)} V(C) \cap V_I$
- $g(v) = f_0(v)$ *otherwise.*

Now, we estimate the number of token-swaps to construct g. Let $s_1, s_2, \ldots, s_p$ be lengths of cycles in $C'_Q(f_0) \cup C_{QI}(f_0)$, where $p = \left|C'_Q(f_0)\right| + |C_{QI}(f_0)|$. Since the tokens on vertices of any cycle with length s_i in $C'_Q(f_0) \cup C_{QI}(f_0)$ can be moved to their target vertices by $(s_i - 1)$ token-swaps, the number of token-swaps to construct g is:

$$
\begin{aligned}
&(s_1 - 1) + (s_2 - 1) + \cdots + (s_p - 1) \\
&= (s_1 + s_2 + \cdots + s_p) - p \\
&= |V(C'_Q(f_0))| + |V(C_{QI}(f_0))| \\
&\quad -(|C'_Q(f_0)| + |C_{QI}(f_0)|) \\
&= |V(C'_Q(f_0))| + |V(C_{QI}(f_0))| + |C^1(f_0)| \\
&\quad -(|C'_Q(f_0)| + |C_{QI}(f_0)| + |C^1(f_0)|) \\
&= |V(C(f_0))| - |V(C'_I(f_0))| \\
&\quad -(|C'_Q(f_0)| + |C_{QI}(f_0)| + |C^1(f_0)|) \\
&= |V(C(f_0))| - |V(C'_I(f_0))| \\
&\quad -(|C(f_0)| - |C'_I(f_0)|). \qquad (1)
\end{aligned}
$$

The second for-statement in **Algorithm 1** moves tokens on vertices of cycles in $C'_I(f_0)$ to their target vertices. Because vertices in the cycles contained in V_I are independent set, tokens on the vertices cannot moved to their target vertices by one token-swap. For a token on a vertex v of $C \in C'_I(f_0)$, we swap the token and a token on a vertex $v' \in V_Q$. Then, we obtain a cycle with one more length. Since the cycle contains a vertex in V_Q and at least two vertices in V_I, the above method for cycles in $C_Q \cup C_{QI}$ works, and hence all tokens on vertices of the cycle can be moved to their target vertices. The number of token-swaps is $t + 1$, where t is the length of C. Let $t_1, t_2, \ldots, t_{|C'_I(f_0)|}$ be lengths of cycles in $C'_I(f_0)$. The number of token-swaps in the second for-statement is

$$
\begin{aligned}
&(t_1 + 1) + (t_2 + 1) + \cdots + (t_{|C'_I(f_0)|} + 1) \\
&\quad = (t_1 + t_2 + \cdots + t_{|C'_I(f_0)|}) + |C'_I(f_0)| \\
&\quad = |V(C'_I(f_0))| + |C'_I(f_0)|. \qquad (2)
\end{aligned}
$$

Taking the sum of Equations (1) and (2), we obtain the number of token-swaps of **Algorithm 1**.

$$
\begin{aligned}
&|V(C(f_0))| - |V(C'_I(f_0))| - (|C(f_0)| - |C'_I(f_0)|) + \\
&|V(C'_I(f_0))| + |C'_I(f_0)| = n - |C(f_0)| + 2|C'_I(f_0)|
\end{aligned}
$$

By the above analysis, we obtain an upper bound as in the following lemma.

Lemma 3. $\mathrm{OPT}_G(f_0) \leq n - |C(f_0)| + 2\left|C'_I(f_0)\right|$.

To show that the upper bound is optimal, we show a lower bound as in the following lemma.

Lemma 4. $\mathrm{OPT}_G(f_0) \geq n - |C(f_0)| + 2\left|C'_I(f_0)\right|$.

Proof. Let f be a token-placement of G, and let $p_G(f) = n - |C(f)| + 2\left|C'_I(f)\right|$. Note that $p_G(f_t) = 0$ holds. We first show that any token-swap decreases $p_G(f)$ by at most one for any token-placement f. That is, we show that $p_G(f') \geq p_G(f) - 1$, where f' is a token-placement adjacent to f and is obtained by swapping two tokens on the edge (u, v).

Case 1: (u, v) is an edge of the clique of G

In this case, the token-swap on (u, v) never change the value of $|C'_I(f)|$. If (u, v) is an underlying edge of D, then $|C(f)|$ is increased by one, and hence $p_G(f') = p_G(f) - 1$. Now, we assume that (u, v) is not an underlying edge of D. If u and v are included in the same cycle, then the token-swap on (u, v) divides the cycle with the two cycles, and hence $p_G(f') = p_G(f) - 1$. Otherwise, suppose that u and v are included in distinct cycles. Then, token-swap on (u, v) decreases $|C(f)|$ by one, since it unifies two cycles of D. Hence $p_G(f') = p_G(f) + 1$ holds.

Case 2: (u, v) is an edge between a vertex of the clique and a vertex of the independent set of G

Without loss of generality, suppose u is a vertex of the clique of G and v is a vertex of the independent set of G. If (u, v) is an underlying edge of a cycle of D, then the token-swap on (u, v) increases $|C(f)|$ by one, and $|C'_I(f)|$ remains the same. Thus, we have $p_G(f') = p_G(f) - 1$. Otherwise, let C_u and C_v be the cycles including u and v, respectively. First consider the case that u and v is included in the same cycle of D, that is C_u and C_v are the same cycle. The token-swap on (u, v) divides C_u into the two cycles, say C'_u and C'_v. We assume that C'_u contains u and C'_v contains v. If $C'_u \in C'_I(f')$ holds, we have $p_G(f') = p_G(f) + 1$. Note that $C'_u \in C_{QI}(f')$ holds, since u is a vertex of the clique. Otherwise, $|C(f)|$ is increased by one, and hence we have $p_G(f') = p_G(f) - 1$. Now we suppose $C_u \neq C_v$. We analyze the following two subcases.

Case (A): $C_v \in C'_I(f_0)$

$|C(f)|$ is decreased by one, and $|C'_I(f)|$ is decreased by one. We therefore obtain $p_G(f') = p_G(f) - 1$.

Case (B): $C_v \notin C'_I(f_0)$

$|C(f)|$ is decreased by one, and hence $p_G(f') = p_G(f) + 1$.

By the above case analysis, we obtain the following inequation;

$$p_G(f') \geq p_G(f) - 1. \qquad (3)$$

Thus, any token-swap decreases $p_G(f)$ by at most one for any token-placement f of G.

Vol.2015-AL-153 No.14
2015/6/13

From Equation (3), for any swapping sequence $\mathcal{S} = \langle f_1, f_2, \ldots, f_h \rangle$ between f_0 and f_t of G, $p_G(f_{i+1}) \geq p_G(f_i) - 1$ holds for $i = 1, 2, \ldots, h-1$. By taking a sum of all the inequations for $i = 1, 2, \ldots, h-1$, we have the following inequations.

$$
\begin{aligned}
p_G(f_t) &\geq p_G(f_0) - \mathrm{len}(\mathcal{S}) \\
\mathrm{len}(\mathcal{S}) &\geq p_G(f_0) - p_G(f_t) \\
\mathrm{len}(\mathcal{S}) &\geq n - |C(f_0)| + 2\left|C'_I(f_0)\right|
\end{aligned}
$$

Thus, we obtain the lower bound $\mathrm{OPT}_G(f) \geq p_G(f)$. $\square$

Immediately from Lemmas 2 and 3, we have the following theorem.

Theorem 5. *For any token-placement f_0 on a complete split graph G,* $\mathrm{OPT}_G(f_0) = n - |C(f_0)| + 2\left|C'_I(f_0)\right|$.

4. Conclusion

We have designed a polynomial-time algorithm that find an exactly optimal solution of token-swapping problem for a complete split graph. Our future works include to design an algorithm for split graphs.

Acknowledgments This work is partially supported by MEXT/JSPS KAKENHI, including the ELC project (Grant Numbers 24106007 and 25330001).

References

[1] Yamanaka, K., Demaine, E., Ito, T., Kawahara, J., Kiyomi, M., Okamoto, Y., Saitoh, T., Suzuki, A., Uchizawa, K. and Uno, T.: Swapping Labeled Tokens on Graphs, *Theoretical Computer Science, accepted*.

[2] Jerrum, M.: The Complexity Of Finding Minimum-length Generator Sequences, *Theoretical Computer Science*, Vol. 36, pp. 265–289 (1985).

[3] Heath, L. and Vergara, J.: Sorting by Short Swaps, *Journal of Computational Biology*, Vol. 10, pp. 775–789 (2003).