The token swapping problem (TSP) and its colored version are reconfiguration problems on graphs. This paper is concerned with the complexity of the TSP and two new variants; namely parallel TSP and parallel colored TSP. For a given graph where each vertex has a unique token on it, the TSP requires to find a shortest way to modify a token placement into another by swapping tokens on adjacent vertices. In the colored version, vertices and tokens are colored and the goal is to relocate tokens so that each vertex has a token of the same color. Their parallel versions allow simultaneous swaps on non-incident edges in one step. We investigate the time complexity of several restricted cases of those problems and show when those problems become tractable and remain intractable.

Keywords

Approximation Algorithm Polynomial Time Bipartite Graph Complete Graph Initial Configuration

These keywords were added by machine and not by the authors. This process is experimental and the keywords may be updated as the learning algorithm improves.

1 Introduction

Yamanaka et al. [14] have introduced a kind of reconfiguration problem on graphs, called the *token swapping problem (TSP)* [1]. Suppose that we have a simple graph where each vertex is assigned a token. Each token is labeled with its unique goal vertex, which may be different from where the token is currently placed. We want to relocate every misplaced token to its goal vertex. What we can do is to swap the two tokens on the ends of an arbitrary edge. The problem is to decide how many swaps are needed to realize the goal token placement. The upper half of Fig. 1 illustrates a problem instance and a solution. The graph has 4 vertices 1, 2, 3, 4 and 4 edges $\{1,2\},\{1,3\},\{2,4\},\{3,4\}$. Each token i is initially put on the vertex $5-i$. By swapping the tokens on the edges $\{3,4\},\{1,3\},\{2,4\},\{3,4\}$ in this order, we can match the indices of the tokens and vertices.

Yamanaka et al. have presented several positive results on the TSP in addition to classical results which can be seen as special cases of the TSP [7]. Namely, graph classes for which the TSP can be solved in polynomial-time are paths, cycles, complete graphs and complete bipartite graphs. They showed that the TSP for general graphs belongs to NP. The NP-hardness is recently shown in the preliminary version [9] of this paper and by Miltzow et al. [11] and Bonnet et al. [2] independently. On the other hand, some polynomial-time approximation algorithms are known for different classes of graphs including the general case [6, 11, 14]. For more backgrounds of the problem, the reader is referred to [14, 15].

A variant of the TSP is the *c-colored token swapping problem (c-CTSP)*. Tokens and vertices in the *c*-CTSP are colored by one of the *c* admissible colors. The *c*-CTSP is to decide how many swaps are required to relocate the tokens so that each vertex has a token of the same color. Yamanaka et al. [15] have investigated the *c*-CTSP and shown that the 3-CTSP is NP-complete while the 2-CTSP is solvable in polynomial time. This problem and a further generalization are also studied in [2].

This paper is concerned with the TSP and variants of it. First, we give a proof of the NP-hardness of the TSP.

- The TSP is NP-complete even when graphs are restricted to bipartite graphs where every vertex has degree at most 3 (Theorem 1).

The result is tight with respect to the maximum vertex degree as the problem is in P if an input graph is a path or a cycle. In addition, we present two polynomial-time solvable subcases of the TSP. One is of lollipop graphs, which are combinations of a complete graph and a path. The other is the class of graphs which are combinations of a star and a path.

Variants of the TSP we will consider in this paper are the parallel versions of the TSP and *c*-CTSP. While in the TSP just one pair of tokens is swapped at once, the *parallel token swapping problem (PTSP)* allows us to swap token pairs on unadjacent edges simultaneously. We call a set of compatible swaps a *parallel swap*. The PTSP is to estimate how many parallel swaps are needed to achieve a goal token configuration. Figure 1 compares optimal solutions for the same instance of the TSP and the PTSP, where two parallel swaps are enough to relocate all the tokens to the goal vertices. Our main results concerning those problems include the following.

Open image in new window

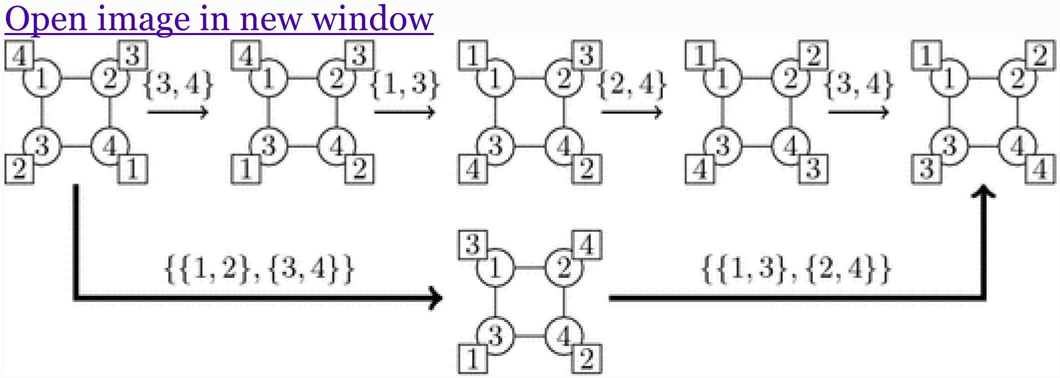

Fig. 1.

Vertices and tokens are shown by circles and squares, respectively. Optimal solutions for the TSP and the PTSP are shown by small and big arrows, respectively.

- The PTSP is NP-complete even to decide whether an instance admits a solution consisting of 3 parallel swaps (Theorem 4).
- One can decide in polynomial time whether an instance of the PTSP admits a solution consisting of 2 parallel swaps (Theorem 6).
- A polynomial-time algorithm that approximately solves the PTSP on paths is presented. It gives a parallel swap sequence whose length is at most one larger than that of an optimal solution (Theorem 7).
- The parallel 2-CTSP is NP-complete (Theorem 9).

The last result contrasts the fact that the 2-CTSP is solvable in polynomial-time [15].

One may consider the TSP and PTSP as special cases of the *minimum generator sequence problem (MGSP)* [4]. The MGSP is to determine whether one can obtain a permutation f on a finite set X by multiplying at most k permutations from a finite permutation set Π, where all of X, f, k and Π are input. The problem is known to be PSPACE-complete if k is specified in binary notation [7], while it becomes NP-complete if k is given in unary representation [4]. In the TSP and PTSP, permutation sets Π are restricted to the ones that have a graph representation. However, this does not necessarily mean that the NP-hardness of the PTSP implies the hardness of the MGSP, since the description size of all the admissible parallel swaps on a graph is exponential in the graph size.

2 Time Complexity of the Token Swapping Problem

We denote by $G=(V,E)$ an undirected graph whose vertex set is V and edge set is E. More precisely, elements of E are subsets of V consisting of exactly two distinct elements. A *configuration* f (on G) is a permutation on V, i.e., bijection from V to V. By $[u]_f$ we denote the orbit $\{f^i(u) \mid i \in \mathbb{N}\}$ of $u \in V$ under f. We call each element of V a *token* when we emphasize the fact that it is in the range of f. We say that a token *v is on a vertex u in f* if $v=f(u)$. A *swap* on G is a synonym for an edge of G, which behaves as a transposition. For a configuration f and a swap $e \in E$, the configuration obtained by applying e to f, which we denote by fe, is defined by

$$fe(u)=\begin{cases} f(v) & \text{if } e=\{u,v\}, \\ f(u) & \text{otherwise.} \end{cases}$$

For a sequence $\vec{e}=\langle e_1,\dots,e_m\rangle$ of swaps, the length m is denoted by $|\vec{e}|$. For $i \le m$, by $\vec{e}_{|\le i}$ we denote the prefix $\langle e_1,\dots,e_i\rangle$. The configuration $f\vec{e}$ obtained by applying $\vec{e}$ to f is $(\ldots((fe_1)e_2)\ldots)e_m$. We say that the token $f(u)$ on u *is moved to* v by $\vec{e}$ if $f\vec{e}(v)=f(u)$. We count the total moves of each token $u\in V$ in the application as $\mathrm{move}(f,\vec{e},u)=|\{i\in\{1,\ldots,m\}\mid(f\vec{e}_{|\le i-1})^{-1}(u)\neq(f\vec{e}_{|\le i})^{-1}(u)\}|$.

Clearly $\mathrm{move}(f,\vec{e},u)\ge \mathrm{dist}(f^{-1}(u),(f\vec{e})^{-1}(u))$, where $\mathrm{dist}(u_1,u_2)$ denotes the length of a shortest path between u_1 and u_2, and $\sum_{u\in V}\mathrm{move}(f,\vec{e},u)=2|\vec{e}|$.

We denote the set of *solutions* for a configuration f by $\mathrm{SOL}(G,f)=\{\vec{e}\mid\vec{e}\text{ is a swap sequence on } G \text{ such that } f\vec{e} \text{ is the identity}\}$.

A solution $\vec{e}_0\in\mathrm{SOL}(G,f)$ is said to be *optimal* if $|\vec{e}_0|=\min\{|\vec{e}|\mid\vec{e}\in\mathrm{SOL}(G,f)\}$. The length of an optimal solution is denoted by $\mathrm{OPT}(G,f)$.

Problem 1

(Token Swapping Problem, TSP).

- **Instance:** A graph *G*, a configuration *f* on *G* and a natural number *k*.
- **Question:** $\mathrm{OPT}(G,f)\le k$?

2.1 TSP Is NP-complete

This subsection proves the NP-hardness of the TSP by a reduction from the 3DM, which is known to be NP-complete [8].

Problem 2

(Three dimensional matching problem, 3DM).

- **Instance:** Three disjoint sets A_1,A_2,A_3 such that $|A_1|=|A_2|=|A_3|$ and a set $T\subseteq A_1\times A_2\times A_3$.
- **Question:** Is there $M\subseteq T$ such that $|M|=|A_1|$ and every element of $A_1\cup A_2\cup A_3$ occurs just once in *M*?

An instance of the 3DM is denoted by (*A*, *T*) where $A=A_1\cup A_2\cup A_3$ assuming that the partition is understood. Let $A_k=\{a_{k,1},\ldots,a_{k,n}\}$ for $k\in\{1,2,3\}$ and $T=\{t_1,\ldots,t_m\}$. For notational convenience we write $a\in t$ if $a\in A$ occurs in $t\in T$ by identifying *t* with the set of the elements of *t*. We construct an instance (G_T, f) of the TSP as follows. The vertex set of G_T is $V_A \cup V_T$ with $V_A = \{u_{k,i}, u'_{k,i} \mid k \in \{1,2,3\} \text{ and } i \in \{1,\dots,n\}\}$, $V_T = \{v_{j,k}, v'_{j,k} \mid j \in \{1,\dots,m\} \text{ and } k \in \{1,2,3\}\}$.

The edge set E_T is given by $E_T = \{\{u_{k,i}, v'_{j,k}\}, \{u'_{k,i}, v_{j,k}\} \mid a_{k,i} \in A_k \text{ occurs in } t_j \in T\} \cup \{\{v_{j,k}, v'_{j,l}\} \subseteq V_T \mid j \in \{1,\dots,n\} \text{ and } k \neq l\}$.

We call the subgraph induced by $\{v_{j,1}, v'_{j,2}, v_{j,3}, v'_{j,1}, v_{j,2}, v'_{j,3}\}$ the t_j-cycle. The initial configuration *f* is defined by $f(u_{k,i}) = u'_{k,i}$ and $f(u'_{k,i}) = u_{k,i}$ for all $a_{k,i} \in A_k$ and $k \in \{1,2,3\}$, $f(v_{j,k}) = v_{j,k}$ and $f(v'_{j,k}) = v'_{j,k}$ for all $t_j \in T$ and $k \in \{1,2,3\}$.

In the initial configuration *f*, all and only the tokens in V_A are misplaced. Each token $u_{k,i} \in V_A$ on the vertex $u'_{k,i}$ must be moved to $u_{k,i}$ via (a part of) t_j-cycle for some $t_j \in T$ in which $a_{k,i}$ occurs. To design a short solution for (G_T, f), it is desirable to have swaps at which both of the swapped tokens get closer to the destination. If (*A*, *T*) admits a solution, then one can find an optimal solution for (G_T, f) of length 21*n*, where 9*n* of the swaps satisfy this property as we will see in Lemma 1. On the other hand, such an “efficient” solution is possible only when (*A*, *T*) admits a solution as shown in Lemma 2.

Lemma 1

If (*A*, *T*) has a solution then $OPT(G_T, f) \leq 21n$ with $n = |A_1|$.

Proof

We show in the next paragraph that for each $t_j \in T$, there is a sequence σ_j of 21 swaps such that $g\sigma_j$ is identical to *g* except $(g\sigma_j)(u_{k,i}) = g(u'_{k,i})$ and $(g\sigma_j)(u'_{k,i}) = g(u_{k,i})$ if $a_{k,i}$ occurs in t_j for any configuration *g*. If $M \subseteq T$ is a solution, by collecting σ_j for all $t_j \in M$, we obtain a swap sequence σ_M of length 21*n* such that $f\sigma_M$ is the identity.
Let $t_j = (a_{1,i_1}, a_{2,i_2}, a_{3,i_3})$. We first move each of the tokens u_{k,i_k} on the vertex u'_{k,i_k} to the vertex $v_{j,k}$ and the

tokens u'_{k,i_k} on u_{k,i_k} to $v'_{j,k}$. We then move the tokens u_{k,i_k} on $v_{j,k}$ to the opposite vertex $v'_{j,k}$ of the t_j-cycle for each $k\in\{1,2,3\}$ while moving u'_{k,i_k} on $v'_{j,k}$ to $v_{j,k}$ in the opposite direction simultaneously. At last we make swaps on the same 6 edges we used in the first phase. The above procedure consists of 21 swaps and gives the desired configuration. □

Lemma 2

If $\mathrm{OPT}(G_T,f)\le 21n$ with $n=|A_1|$ then (*A*, *T*) has a solution.

Proof

We first show that 21*n* is a lower bound on $\mathrm{OPT}(G_T,f)$. Suppose that $f\sigma$ is the identity. For each token $u_{k,i}\in V_A$, we have

$$\mathrm{move}(f,\sigma,u_{k,i})\ge \mathrm{dist}(u_{k,i},f^{-1}(u_{k,i}))=\mathrm{dist}(u_{k,i},u'_{k,i})=5.$$

The adjacent vertices of the vertex $u'_{k,i}$ are $v_{j,k}$ such that $a_{k,i}\in t_j$. Among those, let $\tau(u_{k,i})\in V_T$ be the vertex to which $u_{k,i}$ goes for its first step, i.e., the first occurrence of $u'_{k,i}$ in σ is as $\{u'_{k,i},\tau(u_{k,i})\}$. This means that $\mathrm{move}(f,\sigma,\tau(u_{k,i}))\ge 2$, since the token $\tau(u_{k,i})$ must once leave from and later come back to the vertex $\tau(u_{k,i})$. The symmetric discussion holds for all tokens $u'_{k,i}$. Therefore, noting that τ is an injection, we obtain

$$|\sigma|=\frac{1}{2}\sum_{x\in V_A\cup V_T}\mathrm{move}(f,\sigma,x)\ge \frac{1}{2}\sum_{x\in V_A}(\mathrm{move}(f,\sigma,x)+\mathrm{move}(f,\sigma,\tau(x)))\ge 21n.$$

This has shown that if $f\sigma$ is the identity and $|\sigma|\le 21n$, then

1. (1) $\mathrm{move}(f,\sigma,x)=5$ for all $x\in V_A$,
2. (2) $\mathrm{move}(f,\sigma,y)\ne 0$ for $y\in V_T$ if and only if $y=\tau(x)$ for some $x\in V_A$.

Let $M_\sigma=\{y\in V_T \mid \mathrm{move}(f,\sigma,y)\ne 0\}=\{\tau(x)\in V_T \mid x\in V_A\}$. We are now going to prove that if $v_{j,1}\in M_\sigma$ then $\{v_{j,2},v_{j,3},v'_{j,1},v'_{j,2},v'_{j,3}\}\subseteq M_\sigma$, which implies that $\tilde{M}_\sigma=\{t_j\in T \mid v_{j,1}\in M_\sigma\}$ is a solution for (*A*, *T*).
Suppose $v_{j,1}\in M_\sigma$ and let $t_j\cap A_1=\{a_{1,i}\}$. This means that $\tau(u_{1,i})=v_{j,1}$ and $u_{1,i}$ goes

from $u'_{1,i}$ to $u_{1,i}$ through $(u'_{1,i}, v_{j,1}, v'_{j,2}, v_{j,3}, v'_{j,1}, u_{1,i})$ or $(u'_{1,i}, v_{j,1}, v'_{j,3}, v_{j,2}, v'_{j,1}, u_{1,i})$ by (2) and (1). In either case, $v'_{j,1} \in M_\sigma$. Suppose that $u_{1,i}$ takes the former $(u'_{1,i}, v_{j,1}, v'_{j,2}, v_{j,3}, v'_{j,1}, u_{1,i})$. Then $v'_{j,2}, v_{j,3} \in M_\sigma$. Just like $v_{j,1} \in M_\sigma$ implies $v'_{j,1} \in M_\sigma$, we now see $v_{j,2}, v'_{j,3} \in M_\sigma$. □

It is known that the 3DM is still NP-complete if each $a \in A$ occurs at most three times in *T* [5]. Assuming that *T* satisfies this constraint, it is easy to see that G_T is a bipartite graph with maximum vertex degree 3.

Theorem 1

The TSP is NP-complete even on bipartite graphs with maximum vertex degree 3.

2.2 PTIME Subcases of TSP

In this subsection, we present two graph classes on which the TSP can be solved in polynomial time. One is that of *lollipop graphs*, which are obtained by connecting a path and a complete graph with a bridge. That is, a lollipop graph is $L_{m,n} = (V, E)$ where $V = \{-m, \ldots, -1, 0, 1, \ldots, n\}$ and

$$E = \{\{i, j\} \subseteq V \mid i < j \leq 0 \text{ or } j = i + 1 > 0\}.$$

The other class consists of graphs obtained by connecting a path and a star. A *star-path graph* is $Q_{m,n} = (V, E)$ such that $V = \{-m, \ldots, -1, 0, 1, \ldots, n\}$ and

$$E = \{\{i, 0\} \subseteq V \mid i < 0\} \cup \{\{i, i+1\} \subseteq V \mid i \geq 0\}.$$

Algorithms 1 and 2 give optimal solutions for the TSP on lollipop and star-path graphs in polynomial time, respectively. Proofs are found in [10].

Open image in new window

Algorithm 1. TSP Algorithm for Lollipop Graphs

Input: A lollipop graph $L_{m,n}$ and a configuration f on $L_{m,n}$
for $k = n, \ldots, 1, 0, -1, \ldots, -m$ **do**
  Move the token k to the vertex k directly;
end for

Open image in new window

Algorithm 2. TSP Algorithm for Star-Path Graphs

Input: A star-path graph $Q_{m,n}$ and a configuration f on $Q_{m,n}$
for $k = n, \ldots, 1, 0, -1, \ldots, -m$ **do**
  while the token on the vertex 0 has an index less than 0 **do**
    Move the token on the vertex 0 to its goal vertex;
  end while
  Move the token k to the vertex k;
end for

3 Parallel Token Swapping Problem

The *parallel token swapping problem (PTSP)* is the parallel version of the TSP. Definitions and notation for the TSP are straightforwardly generalized for the PTSP. A *parallel swap* S on G is a synonym for an involution which is a subset of E, or for a matching of G, i.e., $S \subseteq E$ such that $\{u, v_1\}, \{u, v_2\} \in S$ implies $v_1 = v_2$. For a configuration f and a parallel swap $S \subseteq E$, the configuration obtained by applying S to f is defined by $f_S(u) = f(v)$ if $\{u, v\} \in S$ and $f_S(u) = f(u)$ if $u \notin \bigcup S$. Let

$$\mathrm{P\text{-}SOL}(G,f) = \{\vec{S} \mid \vec{S} \text{ is a parallel swap sequence s.t. } f_{\vec{S}} \text{ is the identity}\}$$ $$\mathrm{P\text{-}OPT}(G,f) = \min\{|\vec{S}| \mid \vec{S} \in \mathrm{P\text{-}SOL}(G,f)\}.$$

Problem 3

(Parallel Token Swapping Problem, PTSP).

- **Instance:** A graph G, a configuration f on G and a natural number k.
- **Question:** $\mathrm{P\text{-}OPT}(G,f) \le k$?

It is trivial that $\mathrm{P\text{-}OPT}(G,f) \le \mathrm{OPT}(G,f) \le \mathrm{P\text{-}OPT}(G,f)|V|/2$, since any parallel swap S consists of at most $|V|/2$ (single) swaps. Since $\mathrm{OPT}(G,f) \le |V|(|V|-1)/2$ holds [14], the PTSP belongs to NP.

Yamanaka et al. [14] discussed the relation between the TSP and parallel sorting on an SIMD machine consisting of several processors with local memory which are connected by a network [1]. The relation to the PTSP is more direct.

Theorem 2

If there is a parallel sorting algorithm with r rounds for an interconnection network G, then $\mathrm{P\text{-}OPT}(G,f) \le r$ for any configuration f on G.

3.1 PTSP Is NP-complete

We show the NP-hardness of the PTSP by a reduction from a restricted kind of the satisfiability problem, which we call *PPN-Separable 3SAT* (*Sep-SAT* for short). For a set *X* of *(Boolean) variables*, $\neg X$ denotes the set of their negative literals. A *3-clause* is a subset of $X \cup \neg X$ whose cardinality is at most 3. An instance of the Sep-SAT consists of three finite collections F_1, F_2, F_3 of 3-clauses such that for each variable $x \in X$, the positive literal *x* occurs just once in each of F_1, F_2 and the negative literal $\neg x$ occurs just once in F_3. We will simply denote a Sep-SAT instance as $F = F_1 \cup F_2 \cup F_3$, from which one can find the right partition in polynomial time.

Theorem 3

[10]. The Sep-SAT is NP-complete.

We give a reduction from the Sep-SAT to the PTSP. For a given instance $F = \{C_1, \dots, C_n\}$ over a variable set $X = \{x_1, \dots, x_m\}$ of the Sep-SAT, we define a graph $G_F = (V_F, E_F)$ in the following manner. Let F be partitioned into F_1, F_2, F_3 where each of F_1 and F_2 has just one occurrence of each variable as a positive literal and F_3 has just one occurrence of each negative literal. Define

$$V_F = \{u_i, u'_i, u_{i,1}, u_{i,2}, u_{i,3}, u_{i,4} \mid 1 \le i \le m\} \cup \{v_j, v'_j \mid 1 \le j \le n\} \cup \{v_{j,i} \mid x_i \in C_j \text{ or } \neg x_i \in C_j\}.$$

The edge set E_F is the least set that makes G_F contain the following paths of length 3:

$$(u_i, u_{i,1}, u_{i,2}, u'_i) \text{ and } (u_i, u_{i,3}, u_{i,4}, u'_i) \text{ for each } i \in \{1, \dots, m\}, \quad (v_j, v_{j,i}, u_{i,k}, v'_j) \text{ if } x_i \in C_j \in F_k \text{ or } \neg x_i \in C_j \in F_k.$$

It is not hard to see that G_F is a bipartite graph. Vertices v_j and v'_j have degree at most 3 for $j \in \{1, \dots, n\}$, while $u_{i,k}$ has degree 4 for $i \in \{1, \dots, m\}$ and $k \in \{1, 2, 3\}$. The initial configuration *f* is defined to be the identity except

$$f(u_i) = u'_i,\ f(u'_i) = u_i,\ f(v_j) = v'_j,\ f(v'_j) = v_j,$$

for each $i \in \{1, \dots, m\}$ and $j \in \{1, \dots, n\}$.
Since $\mathrm{dist}(w, f(w)) = 3$ if $w \ne f(w)$, obviously $\text{P-OPT}(G_F, f) \ge 3$.

Here we describe an intuition behind the reduction by giving the following observation between a 3-step solution for (G_F,f) and a solution for F:

- tokens u_i and u'_i pass vertices $u_{i,1}$ and $u_{i,2}$ iff x_i should be assigned 0, while they pass over $u_{i,3}$ and $u_{i,4}$ iff x_i should be assigned 1,
- if tokens v_j and v'_j pass a vertex $u_{i,k}$ for some $k\in\{1,2\}$ then $C_j\in F_k$ is satisfied thanks to x_i, while if they pass over $u_{i,3}$ then $C_j\in F_3$ is satisfied thanks to $\neg x_i$.

Of course it is contradictory that a clause $C_j\in F_1$ is satisfied by $x_i\in C_j$ which is assigned 0. This impossibility corresponds to the fact that there are no *i*, *j* such that both u_i and v_j with $C_j\in F_1$ go to their respective goals via $u_{i,1}$ in a 3-step solution.

Theorem 4

To decide whether $\text{P-OPT}(G,f)\le 3$ is NP-complete even when *G* is restricted to be a bipartite graph with maximum vertex degree 4.

One can modify the above reduction so that every vertex has degree at most 3 by dividing vertices $u_{i,k}$ into two vertices of degree at most 3. Let $V_F=\{u_i,u'_i,u_{i,1},u'_{i,1},u_{i,2},u'_{i,2},u_{i,3},u'_{i,3},u_{i,4},u'_{i,4}\mid 1\le i\le m\}\cup\{v_j,v'_j\mid 1\le j\le n\}\cup\{v_{j,i},v'_{j,i}\mid x_i\in C_j \text{ or } \neg x_i\in C_j\}$.

Our graph G_F contains the following paths of length 5: $(u_i,u_{i,1},u'_{i,1},u_{i,2},u'_{i,2},u'_i)$ and $(u_i,u_{i,3},u'_{i,3},u_{i,4},u'_{i,4},u'_i)$ for each $i\in\{1,\dots,m\}$, $(v_j,v_{j,i},u_{i,k},u'_{i,k},v'_{j,i},v'_j)$ if $x_i\in C_j\in F_k$ or $\neg x_i\in C_j\in F_k$.

The initial configuration *f* is defined similarly to the previous construction.

Theorem 5

To decide whether $\text{P-OPT}(G,f)\le 5$ is NP-complete even when *G* is restricted to be a bipartite graph with maximum vertex degree 3.

3.2 PTIME Subcases of PTSP

In this subsection we discuss tractable subcases of the PTSP. In contrast to Theorem 4, the 2-step PTSP is decidable in polynomial time. In addition, we present an approximation algorithm for finding a solution for the PTSP on paths whose length can be at most one larger than that of an optimal solution.

2-Step PTSP. It is well-known that any permutation can be expressed as a product of 2 involutions, which means that any problem instance of the PTSP on a complete graph has a 2-step solution. Graphs we treat are not necessarily complete but the arguments by Petersen and Tenner [12] on involution factorization lead to the following observation, which is useful to decide whether $\text{P-OPT}(G,f)\le 2$ for general graphs *G*.

Corollary 1

$\langle S,T\rangle\in\text{P-SOL}(G,f)$ if and only if the set of orbits under *f* is partitioned as $\{\{[u_1]_f,[v_1]_f\},\dots,\{[u_k]_f,[v_k]_f\}\}$ (possibly $[u_j]_f=[v_j]_f$ for some $j\in\{1,\dots,k\}$) so that for every $j\in\{1,\dots,k\}$,

$$\{f^i(u_j),f^{-i}(v_j)\}\in\check{S} \text{ and } \{f^{i+1}(u_j),f^{-i}(v_j)\}\in\check{T} \text{ for all } i\in\mathbb{Z},$$

where $\check{S}=S\cup\{\{v\}\mid v\in V-\bigcup S\}$ for a parallel swap *S*.

Theorem 6

It is decidable in polynomial time if $\text{P-OPT}(G,f)\le 2$ for any *G* and *f*.

Proof

Suppose *G* and *f* are given. One can compute in polynomial time all the orbits $[\cdot]_f$. Let us denote the subgraph of *G* induced by a vertex set $U\subseteq V$ by G_U and the sub-configuration of *f* restricted to $[u]_f\cup[v]_f$ by $f_{u,v}$. The set

$$\Gamma_f=\{\{[u]_f,[v]_f\}\mid \text{P-OPT}(G_{[u]_f\cup[v]_f},f_{u,v})\le 2\}$$

can be computed in polynomial time by Corollary 1. It is clear that $\text{P-OPT}(G,f)\le 2$ if and only if there is a subset $\Gamma\subseteq\Gamma_f$ in which every orbit occurs exactly once. This problem is a very minor variant of the problem of finding a perfect matching on a graph, which can be solved in polynomial time [3]. □

One can calculate the number of 2-step solutions in $\text{P-SOL}(K_n,f)$ for any configuration on the complete graph K_n using Petersen and Tenner's formula [12]. On the other hand, it is a #P-complete problem to calculate $|\text{P-SOL}(G,f)|$ for general graphs *G*. This can be shown by a reduction from the problem of calculating the number of perfect matchings in a bipartite graph, which is known to be #P-complete [13]. For $H=(V,E)$, let the vertex set of *G* be $V'=\{u_i\mid u\in V \text{ and } i\in\{1,2\}\}$ and the edge set $E'=\{(u_i,v_j)\mid(u,v)\in E \text{ and } i,j\in\{1,2\}\}$.

The initial configuration is defined by $f(u_1)=u_2$ and $f(u_2)=u_1$ for all $u \in V$. Then it is easy to see that $|\text{P-SOL}(G,f)|=2^m$ for the number m of perfect matchings in H. Note that if H is bipartite, then so is G.

Approximation Algorithm for the PTSP on Paths. We present an approximation algorithm for the PTSP on paths which outputs a parallel swap sequence whose length is no more than $\text{P-OPT}(P_n,f)+1$, where $P_n=(\{1,\dots,n\},\{\{i,i+1\} \mid 1 \le i < n\})$ and f is a configuration on P_n. We say that a swap $\{i,i+1\} \in E$ is *reasonable w.r.t. f* if $f(i)>f(i+1)$, and moreover, a parallel swap sequence $\vec{S}=\langle S_1,\dots,S_m\rangle$ is *reasonable w.r.t. f* if every $e \in S_j$ is reasonable w.r.t. $f\langle S_1,\dots,S_{j-1}\rangle$ for all $j \in \{1,\dots,m\}$. The parallel swap sequence $\langle S_1,\dots,S_m\rangle$ output by Algorithm 3 is reasonable and satisfies the condition which we call the *odd-even condition*: for each odd number j, all swaps in S_j are of the form $\{2i-1,2i\}$ for some $i \ge 1$, and for each even number j, all swaps in S_j are of the form $\{2i,2i+1\}$ for some $i \ge 1$.

Lemma 3

Suppose that $g=fS$ for a reasonable parallel swap S w.r.t. f. For any $\langle S_1,\dots,S_m\rangle \in \text{P-SOL}(P_n,f)$, there is $\langle S'_1,\dots,S'_m\rangle \in \text{P-SOL}(P_n,g)$ such that $S'_j \subseteq S_j$ for all $j \in \{1,\dots,m\}$.

The lemma implies that we may assume without loss of generality that an optimal solution $\langle S_1,\dots,S_m\rangle$ is reasonable and moreover if $f\langle S_1,\dots,S_j\rangle(i)>f\langle S_1,\dots,S_j\rangle(i+1)$ then $\{i,i+1\} \cap \bigcup S_{j+1} \neq \emptyset$ for $j<m$.

Open image in new window

Algorithm 3. Approximation algorithm for PTSP on paths

Input: A configuration f_0 on P_n
Output: A solution $\vec{S} \in \text{P-SOL}(P_n, f_0)$
Let $j = 0$;
while f_j is not identity **do**
  Let $j = j + 1$, $S_j = \{\{i, i+1\} \mid f_{j-1}(i) > f_{j-1}(i+1)$ and $i + j$ is even$\}$ and $f_j = f_{j-1}S_j$;
end while
return $\langle S_1, \dots, S_j\rangle$;

Let us denote the output of Algorithm 3 by $AP(P_n,f_0)$.

Theorem 7

$AP(P_n,f_0) \in$ P-SOL(P_n,f_0) and $|AP(P_n,f_0)| \le$ P-OPT$(P_n,f_0)+1$.

Proof

Let $\vec{T} = AP(P_n,f_0)$. It is obvious that $\vec{T} \in$ P-SOL(P_n,f_0) and it is odd-even. It is easy to see by Lemma 3 that $|\vec{T}| \le |\vec{S}|$ for any odd-even solution $\vec{S} \in$ P-SOL(P_n,f_0).

We next show that every swap sequence $\vec{S} = \langle S_1,\dots,S_m\rangle$ admits an equivalent odd-even sequence that is not much longer than the original. Without loss of generality we assume that $S_j \cap S_{j+1} = \emptyset$ for any *j* (in fact, any reasonable parallel swap sequence meets this condition). For a parallel swap sequence $\vec{S} = \langle S_1,\dots,S_m\rangle$, define Open image in new window $\text{Œ}(\vec{S}) = \langle S'_1,\dots,S'_{m+1}\rangle$ by delaying swaps which do not meet the odd-even condition, that is,

$$S'_j = \{\{i,i+1\} \in S_j \cup S_{j-1} \mid i+j \text{ is even}\}$$

for $j=1,\dots,m+1$ assuming that $S_0 = S_{m+1} = \emptyset$. By the parity restriction, each S'_j is a parallel swap. It is easy to show by induction on *j* that

$$f\langle S'_1,\dots,S'_j\rangle(i) = \begin{cases} f\langle S_1,\dots,S_{j-1}\rangle(i) & \text{if } \{i,i+1\} \in S_j \text{ and } i+j \text{ is odd,} \\ f\langle S_1,\dots,S_j\rangle(i) & \text{otherwise,} \end{cases}$$

for each $j \in \{1,\dots,m+1\}$, which implies that Open image in new window $f\vec{S} = f\text{Œ}(\vec{S})$. Therefore, for an optimal reasonable solution $\vec{S}_0$, we have Open image in new window $|\vec{S}_0| + 1 = |\text{Œ}(\vec{S}_0)| \ge |\vec{T}|$. □□

4 Parallel Colored Token Swapping Problem

The *colored token swapping problem* (CTSP) is a generalization of the TSP, where each token is colored and different tokens may have the same color. By swapping tokens on adjacent vertices, the goal coloring configuration should be realized. More formally, a *coloring* is a map *f* from *V* to $\mathbb{N}$. The definition of a swap application to a configuration can be applied to colorings with no change. We say that two colorings *f* and *g* are *consistent* if $|f^{-1}(i)| = |g^{-1}(i)|$ for all $i \in \mathbb{N}$. Since the problem is a generalization of the TSP, obviously it is NP-hard. Yamanaka et al. [15] have investigated subcases of the CTSP called the *c*-CTSP where the codomain of colorings is restricted to $\{1,\dots,c\}$. We discuss the parallel version of the *c*-CTSP in this section.

Problem 4

(Parallel c -Colored Token Swapping Problem, c -PCTSP).

- **Instance:** A graph G, two consistent c-colorings f and g, and a number $k \in \mathbb{N}$.
- **Question:** Is there $\vec{S}$ with $|\vec{S}| \le k$ such that $f\vec{S} = g$?

Define $\text{P-OPT}(G,f,g)=\min\{|\vec{S}| \mid f\vec{S}=g\}$ for two consistent colorings f and g. Since $\text{P-OPT}(G,f,g)$ can be bounded by $\text{P-OPT}(G,h)$ for some configuration h, the c-PTSP belongs to NP.

Yamanaka et al. have shown that the 3-CTSP is NP-hard by a reduction from the 3DM. It is not hard to see that their reduction works to prove the NP-hardness of the 3-PCTSP. We then obtain the following theorem as a corollary to their discussion.

Theorem 8

To decide whether $\text{P-OPT}(G,f,g) \le 3$ is NP-hard even if G is restricted to be a planar bipartite graph with maximum vertex degree 3 and f and g are 3-colorings.

Yamanaka et al. have shown that the 2-CTSP is solvable in polynomial time on the other hand. In contrast, we prove that the 2-PCTSP is still NP-hard.

Theorem 9

To decide whether $\text{P-OPT}(G,f,g) \le 3$ is NP-hard even if G is restricted to be a bipartite graph with maximum vertex degree 4 and 2-colorings f and g.

Proof

We prove the theorem by a reduction from the Sep-SAT. We use the same graph used in the proof of Theorem 4. The initial and goal colorings f and g are defined to be $f(w)=1$ and $g(w)=1$ for all w but $f(u_i)=g(u'_i)=2$ for each $x_i \in X$, $f(v_j)=g(v'_j)=2$ for each $C_j \in F_1 \cup F_3$ and $f(v'_j)=g(v_j)=2$ for each $C_j \in F_2$. The claim that F is satisfiable if and only if $\text{P-OPT}(G_F,f,g)=3$ can be established by the same manner as the proof of Theorem 4. □

We can also show the following using the ideas for proving Theorems 5 and 8.

Theorem 10

To decide whether $P\text{-}OPT(G,f,g)\leq 5P\text{-}OPT(G,f,g)\leq 5$ is NP-hard even if *G* is restricted to be a bipartite graph with maximum vertex degree 3 and *f* and *g* are 2-colorings.