Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 23, no. 1, pp. 29–70 (2019)
DOI: 10.7155/jgaa.00483

The Time Complexity of Permutation Routing via Matching, Token Swapping and a Variant

Jun Kawahara[1] *Toshiki Saitoh*[2] *Ryo Yoshinaka*[3]

[1]Graduate School of Science and Technology,
Nara Institute of Science and Technology, Japan
[2]Faculty of Computer Science and Systems Engineering,
Kyushu Institute of Technology, Japan
[3]Graduate School of Information Sciences, Tohoku University, Japan

Abstract

The problems of Permutation Routing via Matching and Token Swapping are reconfiguration problems on graphs. This paper is concerned with the complexity of those problems and a colored variant. For a given graph where each vertex has a unique token on it, those problems require to find a shortest way to modify a token placement into another by swapping tokens on adjacent vertices. While all pairs of tokens on a matching can be exchanged at once in Permutation Routing via Matching, Token Swapping allows only one pair of tokens can be swapped. 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. We investigate the time complexity of several restricted cases of those problems and show when those problems become tractable and remain intractable.

Submitted:	Reviewed:	Revised:	Accepted:
September 2017	May 2018	June 2018	August 2018
	Final:	Published:	
	August 2018	January 2019	

Article type: Regular paper
Communicated by: M.S. Rahman, H.-C. Yen and S.-H. Poon

Research supported by JSPS KAKENHI Grant Number 16K16006, CREST Grant Numbers JPMJCR1401 and JPMJCR1402, and NAIST Bigdata project

E-mail addresses: jkawahara@is.naist.jp (Jun Kawahara) toshikis@ces.kyutech.ac.jp (Toshiki Saitoh) ryoshinaka@tohoku.ac.jp (Ryo Yoshinaka)

1 Introduction

Alon et al. [1] have proposed a problem called *Permutation Routing via Matching* as a simple variant of routing problems.[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 in one step is to pick a matching and swap the two tokens on the ends of each edge in the matching. The problem is to decide how many steps are needed to realize the goal token placement. The bottom half of Figure 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 in the matchings $\{\{1, 2\}, \{3, 4\}\}$ and $\{\{1, 3\}, \{2, 4\}\}$ in this order, we achieve the goal. The original paper of Alon et al. [1] and following papers are mostly interested in the maximum number of steps, denoted $\mathsf{rt}(G)$, needed to realize the goal configuration from any initial configuration for an input graph G. For example, Alon et al. [1] have shown $\mathsf{rt}(K_n) = 2$ for complete graphs K_n, Zhang [21] has shown $\mathsf{rt}(T) = 3n/2+(\log n)$ for trees T of n vertices, and Li et al. [13] have shown $\mathsf{rt}(K_{m,n}) \in \lfloor 3m/2n \rfloor + O(1)$ for bipartite graphs $K_{m,n}$ with $m \geq n$ and $\mathsf{rt}(C_n) = n - 1$ for $n \geq 3$ for cycles C_n. This paper is concerned with the problem where an initial configuration also constitutes an input and discusses its computational complexity. We will show the following results, which were independently obtained by Banerjee and Richards [2].

- Permutation Routing via Matching is NP-complete even to decide whether an instance admits a 3-step solution (Theorem 3).
- To decide whether a 2-step solution exists can be answered in polynomial-time (Theorem 5).

In addition, we present a polynomial-time algorithm that approximately solves Permutation Routing via Matching on paths whose output is at most one larger than that of the exact answer (Theorem 7).

Token Swapping, introduced by Yamanaka et al. [19], can be seen as permutation routing via "edges". In this setting we can swap only two tokens on an edge at each step. Figure 1 shows that we require 4 steps in Token Swapping to realize the goal configuration, while 2 steps are enough in Routing via Matching. Yamanaka et al. have presented several positive results on this problem in addition to classical results which can be seen as special cases [9]. Namely, graph classes for which Token Swapping can be solved in polynomial-time are paths, cycles, complete graphs and complete bipartite graphs. They showed that Token Swapping for general graphs belongs to NP. The NP-hardness is recently shown in preliminary versions [11, 12] of this paper and by Miltzow et al. [14] and Bonnet et al. [3] independently. On the other hand, some polynomial-time

[1] In the preliminary version [12] of this paper, this old problem was called Parallel Token Swapping due to the ignorance of the authors.

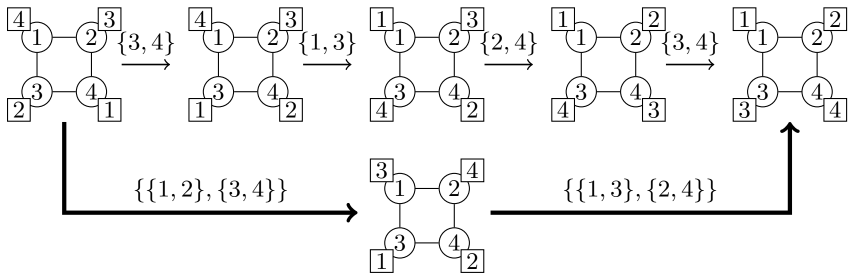

Figure 1: Vertices and tokens are shown by circles and squares, respectively. Optimal solutions for Token Swapping and Permutation Routing via Matching are shown by small and big arrows, respectively.

approximation algorithms are known for different classes of graphs including the general case [8, 14, 19]. Our NP-hardness result is tight with respect to the degree bound, as the problem can be solved in polynomial-time if input graphs have vertex degree at most 2.

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

Moreover, we present two polynomial-time solvable subcases of Token Swapping. 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.

A variant of Token Swapping is *c-Colored Token Swapping*. Tokens and vertices in this setting are colored by one of c admissible colors. We decide how many swaps are required to relocate the tokens so that each vertex has a token of the same color. Since different tokens and vertices may have the same color, there can be many possible destinations for each token. Yamanaka et al. [20] have shown that 3-Colored Token Swapping is NP-complete while 2-Colored Token Swapping is solvable in polynomial time. This problem and a further generalization are also studied in [3]. In this paper we consider the colored version of Routing via Matching and show that it is also NP-complete.

- 2-Coloring Routing via Matching is NP-complete even to decide whether an instance admits a 3-step solution (Theorem 9).
- 3-Coloring Routing via Matching is NP-complete even to decide whether an instance admits a 2-step solution (Theorem 11).

The former result contrasts the fact that the 2-Colored Token Swapping is solvable in polynomial-time [20]. The latter contrasts that to decide whether a 2-step solution exists for Permutation Routing is in P (Theorem 5). In addition, we present another contrastive result.

- It is decidable in polynomial-time whether a 2-step solution exists in 2-Coloring Routing via Matching (Theorem 12).

One may consider permutation routing on graphs as a special case of the *Minimum Generator Sequence Problem* [6]. The problem 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 [9], while it becomes NP-complete if k is given in unary representation [6]. In our settings, permutation sets Π are restricted to the ones that have a graph representation. However, this does not necessarily mean that the NP-hardness of Permutation Routing via Matching implies the hardness of the Minimum Generator Sequence Problem, since the description size of all the admissible parallel swaps on a graph is exponential in the graph size.

2 Time Complexity of Token Swapping

We denote by $G = (V, E)$ an undirected simple 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 $i \leq m$, by $\vec{e}_{|\leq i}$ we denote the prefix $\langle e_1, \dots, e_i \rangle$. The configuration $f\vec{e}$ obtained by applying $\vec{e}$ to f is $(\dots((fe_1)e_2)\dots)e_m$. 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

$$\mathsf{move}(f, \vec{e}, u) = |\{\, i \in \{1, \dots, m\} \mid (f\vec{e}_{|\leq i-1})^{-1}(u) \neq (f\vec{e}_{|\leq i})^{-1}(u) \,\}|\,.$$

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

We denote the set of *solutions* for a configuration f by

$$\mathsf{TS}(G, f) = \{\, \vec{e} \mid \vec{e} \text{ is a swap sequence on } G \text{ such that } f\vec{e} \text{ is the identity} \,\}\,.$$

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

Problem 1 (Token Swapping)

Instance: A connected graph G, a configuration f on G and a natural number k.

Question: $\mathsf{ts}(G, f) \leq k$?

2.1 Token Swapping Is NP-complete

This subsection proves the NP-hardness of Token Swapping by a reduction from the 3DM, which is known to be NP-complete [10].

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 Token Swapping as follows. The vertex set of G_T is $V_A \cup V_T$ with

$$V_A = \{\, u_{k,i}, u'_{k,i} \mid k \in \{1,2,3\} \text{ and } i \in \{1, \ldots, n\}\}\,,$$
$$V_T = \{\, v_{j,k}, v'_{j,k} \mid j \in \{1, \ldots, m\} \text{ and } k \in \{1,2,3\}\}\,.$$

The edge set E_T is given by

$$E_T = \{\, \{u_{k,i}, v'_{j,k}\}, \{u'_{k,i}, v_{j,k}\} \mid a_{k,i} \in A_k \text{ occurs in } t_j \in T\,\}$$
$$\cup \{\, \{v_{j,k}, v'_{j,l}\} \subseteq V_T \mid j \in \{1, \ldots, m\} \text{ and } k, l \in \{1,2,3\} \text{ with } k \neq l\,\}\,.$$

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

$$f(u_{k,i}) = u'_{k,i} \text{ and } f(u'_{k,i}) = u_{k,i} \text{ for all } a_{k,i} \in A_k \text{ and } k \in \{1,2,3\}\,,$$
$$f(v_{j,k}) = v_{j,k} \text{ and } f(v'_{j,k}) = v'_{j,k} \text{ for all } t_j \in T \text{ and } k \in \{1,2,3\}\,.$$

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.

Example 1 Let $A = A_1 \cup A_2 \cup A_3$ and $T = \{t_1, t_2, t_3\}$ where $A_k = \{a_{k,1}, a_{k,2}\}$ for $k \in \{1, 2, 3\}$, $t_1 = \{a_{1,1}, a_{2,1}, a_{3,1}\}$, $t_2 = \{a_{1,1}, a_{2,1}, a_{3,2}\}$ and $t_3 = \{a_{1,2}, a_{2,2}, a_{3,2}\}$. Figure 2 shows the graph and initial configuration reduced from the 3DM instance (A, T). This instance (A, T) has a solution $M = \{t_1, t_3\}$. The proof of Lemma 1 will give how to find an optimal solution for the reduced Token Swapping instance corresponding to M. A part of the solution is illustrated in Figure 3.

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

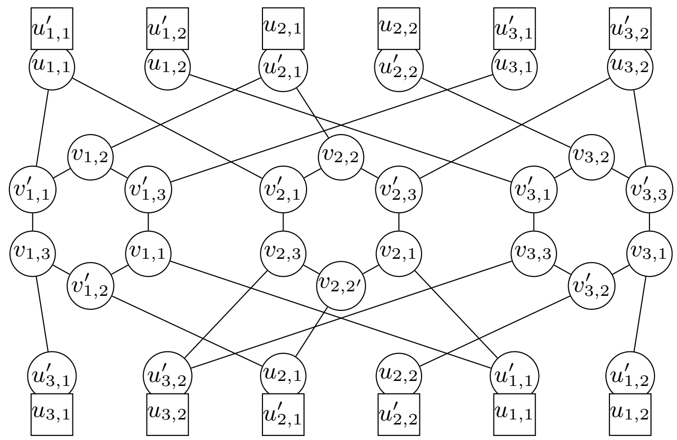

Figure 2: The graph and initial configuration of Token Swapping reduced from the 3DM instance in Example 1. Vertices and tokens are denoted by circles and squares, respectively. The tokens which are already on the goal vertices in the initial configuration are omitted.

solution, then one can find an optimal solution for (G_T, f) of length $21n$, where $9n$ 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* $\mathsf{ts}(G_T, f) \le 21n$ *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 $21n$ 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* $\mathsf{ts}(G_T, f) \le 21n$ *with* $n = |A_1|$ *then* (A, T) *has a solution.*

Proof: We first show that $21n$ is a lower bound on $\mathsf{ts}(G_T, f)$. Let σ be a solution in $\mathsf{TS}(G, f)$. For each token $u_{k,i} \in V_A$, we have

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

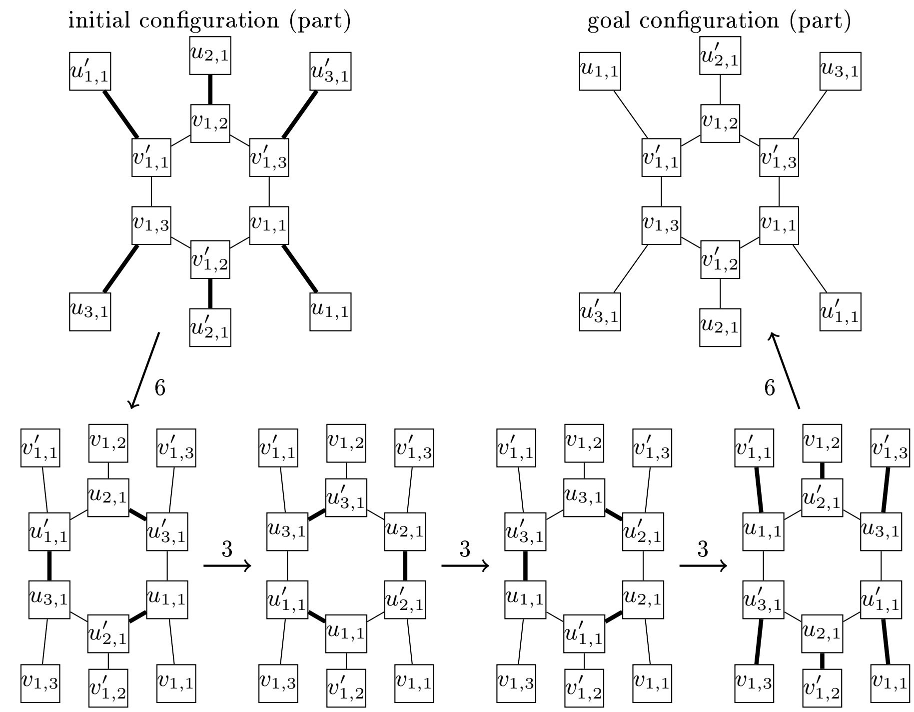

Figure 3: The 3DM instance (A, T) of Example 1 has a solution $M = \{t_1, t_3\}$. The optimal solution given in the proof of Lemma 1 that exchanges $u_{k,1}$ and $u'_{k,1}$ for all $k \in \{1, 2, 3\}$ via the t_1-cycle is illustrated here, where we suppress vertex names. By swapping the tokens on the bold edges in each configuration, we obtain the succeeding one pointed by an arrow. The number by each arrow shows the number of swaps. The swap sequence consists of 21 swaps in total. By doing the same on t_3-cycle with respect to $u_{1,2}, u_{2,2}, u_{3,2}, u'_{1,2}, u'_{2,2}, u'_{3,2}$, we obtain the goal configuration.

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 $\mathsf{move}(f,\sigma,\tau(u_{k,i})) \geq 2$, 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

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

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

(1) $\mathsf{move}(f,\sigma,x) = 5$ for all $x \in V_A$,

(2) $\mathsf{move}(f,\sigma,y) \neq 0$ for $y \in V_T$ if and only if $y = \tau(x)$ for some $x \in V_A$.

Let $M_\sigma = \{\, y \in V_T \mid \mathsf{move}(f,\sigma,y) \neq 0 \,\} = \{\, \tau(x) \in V_T \mid x \in V_A \,\}$. 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 $\widetilde{M_\sigma} = \{\, t_j \in T \mid v_{j,1} \in M_\sigma \,\}$ 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 [7]. 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 *Token Swapping is NP-complete even on bipartite graphs with maximum vertex degree* 3.

The NP-hardness of Token Swapping was independently proven by Miltzow et al. [14] and by Bonnet et al. [3]. The graphs obtained by the reduction of Miltzow et al. have a degree bound but it is not as small as our constraint. Our bound 3 is tight, as Token Swapping on graphs with degree at most 2, i.e., paths and cycles, is solvable in polynomial-time. Bonnet et al. [3] have given no degree constraint but their graphs have tree-width 2 and diameter 6. Therefore, their and our results are incomparable.

2.2 PTIME Subcases of Token Swapping

In this subsection, we present two graph classes on which Token Swapping 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 $V = \{\, -m, \dots, -1, 0, 1, \dots, n \,\}$ 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 $V = \{-m, \dots, -1, 0, 1, \dots, n\}$ 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 Token Swapping on lollipop and star-path graphs in polynomial time, respectively. Proofs of the correctness are found in Appendices A and B.

Algorithm 1 Algorithm for Token Swapping on Lollipop Graphs

```
Input: A lollipop graph L_{m,n} and a configuration f on L_{m,n}
for k = n, ..., 1, 0, −1, ..., −m do
  Move the token k to the vertex k directly;
end for
```

Algorithm 2 Algorithm for Token Swapping on Star-Path Graphs

```
Input: A star-path graph Q_{m,n} and a configuration f on Q_{m,n}
for k = n, ..., 1, 0, −1, ..., −m do
  while the token on the vertex 0 has a label 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 Permutation Routing via Matching

Permutation Routing via Matching can be seen as the parallel version of Token Swapping. Definitions and notation for Token Swapping are straightforwardly generalized as follows. 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 $fS(u) = f(v)$ if $\{u, v\} \in S$ and $fS(u) = f(u)$ if $u \notin \bigcup S$. This definition is straightforwardly generalized for sequences $\vec{S} = \langle S_1, \dots, S_m \rangle$ of parallel swaps by $f\vec{S} = (\dots((fS_1)S_2)\dots)S_m$. Let

$$\mathsf{RT}(G, f) = \{\vec{S} \mid \vec{S} \text{ is a parallel swap sequence s.t. } f\vec{S} \text{ is the identity}\}$$
$$\mathsf{rt}(G, f) = \min\{|\vec{S}| \mid \vec{S} \in \mathsf{RT}(G, f)\}.$$

Problem 3 (Permutation Routing via Matching)

Instance: A connected graph G, a configuration f on G and a natural number k.

Question: $\mathsf{rt}(G, f) \leq k$?

It is trivial that $\mathsf{rt}(G, f) \leq \mathsf{ts}(G, f) \leq \mathsf{rt}(G, f)|V|/2$, since any parallel swap S consists of at most $|V|/2$ (single) swaps. Since $\mathsf{ts}(G, f) \leq |V|(|V| - 1)/2$ holds [19], Permutation Routing via Matching belongs to NP.

3.1 Routing Permutations via Matching Is NP-complete

We show that Routing Permutations via Matching is NP-hard 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 Sep-SAT is a finite collection F of 3-clauses, which can be partitioned into three subsets $F_1, F_2, F_3 \subseteq F$ such that for each variable $x \in X$, the positive literal x occurs just once in each of F_1, F_2 and never in F_3, and the negative literal $\neg x$ occurs just once in F_3 and never in F_1 nor F_2. Note that one can find a partition $\{F_1, F_2, F_3\}$ of a Sep-SAT instance F in linear time.

Theorem 2 *Sep-SAT is NP-complete.*

Proof: See Appendix C. □

We give a reduction from Sep-SAT to Permutation Routing via Matching. For a given instance $F = \{C_1, \ldots, C_n\}$ over a variable set $X = \{x_1, \ldots, x_m\}$ of 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

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

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

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

The fact that G_F is bipartite can be seen by partitioning V_F into

$$\{\, u_i, u_{i,2}, u_{i,4} \mid 1 \leq i \leq m \,\} \cup \{\, v_j \mid C_j \in F_2 \,\} \cup \{\, v'_j, v_{j,i} \mid C_j \in F_1 \cup F_3 \,\}$$

and the rest. Vertices v_j and v'_j have degree at most 3 for $j \in \{1, \ldots, n\}$, while $u_{i,k}$ has degree 4 for $i \in \{1, \ldots, m\}$ 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 $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, n\}$.

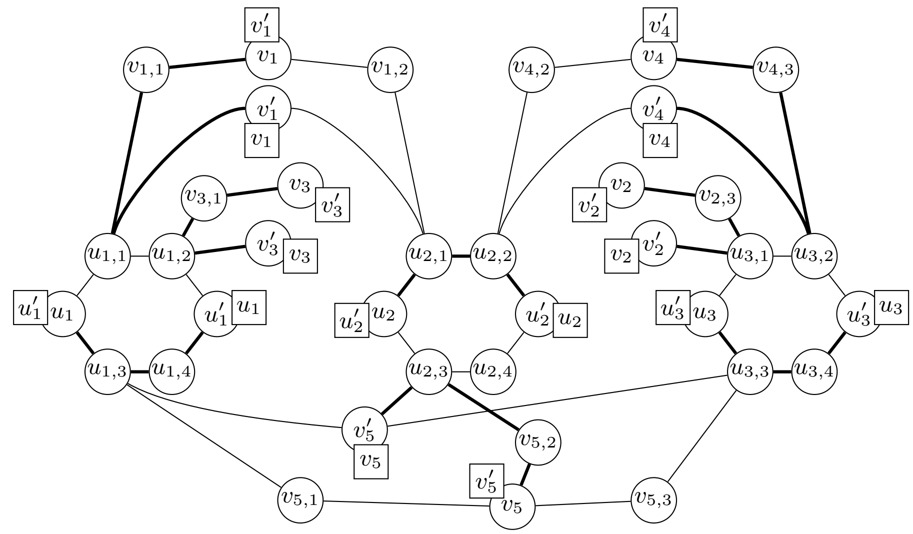

Figure 4: The instance of Permutation Routing via Matching obtained from the Sep-SAT instance F of Example 2. By moving misplaced tokens along the bold edges, the goal configuration is realized in 3 steps. The reduction graph described in the proof for Theorem 4 has essentially the same shape.

Example 2 For $X = \{x_1, x_2, x_3\}$, let F consist of $C_1 = \{x_1, x_2\}$, $C_2 = \{x_3\}$, $C_3 = \{x_1\}$, $C_4 = \{x_2, x_3\}$ and $C_5 = \{\neg x_1, \neg x_2, \neg x_3\}$. Then F is partitioned into $F_1 = \{C_1, C_2\}$, $F_2 = \{C_3, C_4\}$ and $F_3 = \{C_5\}$, where each variable occurs just once in each F_k with $k \in \{1, 2, 3\}$. Moreover, F_1 and F_2 have only positive literals and F_3 has only negative literals. Therefore, F is a Sep-SAT instance. Figure 4 shows the reduction from F. The formula F is satisfied by assigning 1 to x_1, x_3 and 0 to x_2. Corresponding to this assignment, by moving misplaced tokens along the bold edges in Figure 4, the goal configuration is realized in 3 steps.

Since $\mathsf{dist}(w, f(w)) = 3$ if $w \neq f(w)$, obviously $\mathsf{rt}(G_F, f) \geq 3$. We will show that F is satisfiable if and only if this lower bound is achieved. 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.

Lemma 3 *The formula F is satisfiable if and only if* $\mathsf{rt}(G_F, f) = 3$.

Proof: Suppose that there is $\phi : X \to \{0,1\}$ satisfying F. Then each clause must have a literal to which ϕ assigns 1. Let $\psi : F \to X$ be such that $\psi(C_j) \in C_j$ and $\phi(\psi(C_j)) = 1$ if $C_j \in F_1 \cup F_2$, and $\neg\psi(C_j) \in C_j$ and $\phi(\psi(C_j)) = 0$ if $C_j \in F_3$. Define

$$\begin{aligned} S_1 = \{\, &\{u_i, u_{i,1}\}, \{u'_i, u_{i,2}\} \mid \phi(x_i) = 0 \,\} \cup \{\, \{u_i, u_{i,3}\}, \{u'_i, u_{i,4}\} \mid \phi(x_i) = 1 \,\} \\ &\cup \{\, \{v_j, v_{j,i}\}, \{v'_j, u_{i,k}\} \mid \psi(C_j) = x_i \text{ and } C_j \in F_k \,\}\,, \\ S_2 = \{\, &\{u_{i,1}, u_{i,2}\} \mid \phi(x_i) = 0 \,\} \cup \{\, \{u_{i,3}, u_{i,4}\} \mid \phi(x_i) = 1 \,\} \\ &\cup \{\, \{v_{j,i}, u_{i,k}\} \mid \psi(C_j) = x_i \text{ and } C_j \in F_k \,\}\,. \end{aligned}$$

It is not hard to see that $\langle S_1, S_2, S_1 \rangle$ is a solution for (G_F, f).

Conversely, suppose that (G_F, f) admits a solution $\langle S_1, S_2, S_3 \rangle$. Since the token on u_i is moved to u'_i by the three steps, the path that u'_i takes should be either $(u_i, u_{i,1}, u_{i,2}, u'_i)$ or $(u_i, u_{i,3}, u_{i,4}, u'_i)$. In other words, S_2 contains at least one of $\{u_{i,1}, u_{i,2}\}$ and $\{u_{i,3}, u_{i,4}\}$. We prove that F is satisfied by the assignment $\phi : X \to \{0,1\}$ defined as

$$\phi(x_i) = \begin{cases} 0 & \text{if } \{u_{i,1}, u_{i,2}\} \in S_2\,, \\ 1 & \text{otherwise.} \end{cases}$$

For each $C_j \in F_1$, the token on v_j must be moved to v'_j via $u_{i,1}$ for some i such that $x_i \in C_j$. That is, $\{v_{j,i}, u_{i,1}\} \in S_2$. Since S_2 is a parallel swap, $\{u_{i,1}, u_{i,2}\} \notin S_2$ in this case, which means $\phi(x_i) = 1$. Hence C_j is satisfied by ϕ. Almost the same arguments show that clauses in F_2 and F_3 are also satisfied by ϕ. □

Theorem 3 *For any fixed $k \geq 3$, to decide whether* $\mathsf{rt}(G, f) \leq k$ *is NP-complete even when G is restricted to be a bipartite graph with maximum vertex degree* 4.

Proof: Lemma 3 proves the theorem for $k = 3$. For $k = 3 + h$ with $h > 0$, by adding an extra path of length h to each of the vertices u_i, u'_i, v_j, v'_j for $i \in \{1, \dots, m\}$ and $j \in \{1, \dots, n\}$, and putting the corresponding tokens on the end of those paths in the initial configuration, we obtain an instance with the desired property. That is, we add the following paths:

$$\begin{aligned} &(u_{i,-h}, \dots, u_{i,-1}, u_i), (u'_i, u'_{i,-1}, \dots, u'_{i,-h}) \text{ for each } i \in \{1, \dots, m\}, \\ &(v_{j,-h}, \dots, v_{j,-1}, v_j), (v'_j, v'_{j,-1}, \dots, v'_{j,-h}) \text{ for each } j \in \{1, \dots, n\}. \end{aligned}$$

For those vertices on the new paths, we let

$$\begin{aligned} &f(u_{i,-h}) = u'_i,\ f(u_{i,-l}) = u_{i,-l-1} \text{ for } 1 \leq l < h \text{ and } f(u_i) = u_{i,-1}, \\ &f(u'_{i,-h}) = u_i,\ f(u'_{i,-l}) = u'_{i,-l-1} \text{ for } 1 \leq l < h \text{ and } f(u'_i) = u'_{i,-1}, \\ &f(v_{j,-h}) = v'_j,\ f(v_{j,-l}) = v_{j,-l-1} \text{ for } 1 \leq l < h \text{ and } f(v_j) = v_{j,-1}, \\ &f(v'_{j,-h}) = v_j,\ f(v'_{j,-l}) = v'_{j,-l-1} \text{ for } 1 \leq l < h \text{ and } f(v'_j) = v'_{j,-1}. \end{aligned}$$

□

Banerjee and Richards [2] have shown Theorem 3 using a different reduction.

One can modify our 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

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

The new graph G_F' contains the following paths of length 5:

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

The initial configuration f is defined in the same manner as the previous construction. It is identity except $f(u_i) = u_i'$, $f(u_i') = u_i$, $f(v_j) = v_j'$, and $f(v_j') = v_j$ for $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, n\}$. The formula F is satisfiable if and only if $\mathsf{rt}(G_F', f) = 5$.

Theorem 4 *For any fixed $k \ge 5$, to decide whether $\mathsf{rt}(G, f) \le k$ is NP-complete even when G is restricted to be a bipartite graph with maximum vertex degree 3.*

3.2 PTIME Subcases

In this subsection we discuss tractable subcases of Permutation Routing via Matching. In contrast to Theorem 3, it is decidable in polynomial time whether an instance of Permutation Routing via Matching admits a 2-step solution. In addition, we present an approximation algorithm for finding a solution for Permutation Routing via Matching on paths whose length can be at most one larger than that of an optimal solution.

3.2.1 2-Step Permutation Routing via Matching

It is well-known that any permutation can be expressed as a product of 2 involutions, which means that any problem instance of Permutation Routing via Matching on a complete graph has a 2-step solution. Graphs we treat are not necessarily complete but the arguments by Petersen and Tenner [16, Lemma 2.3] on involution factorization lead to the following observation, which is useful to decide whether $\mathsf{rt}(G, f) \le 2$ for general graphs G.

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

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

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

Theorem 5 *It is decidable in polynomial time if* $\mathsf{rt}(G, f) \leq 2$ *for any* G *and* f*.*[2]

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

$$\Gamma_f = \{\, \{[u]_f, [v]_f\} \mid \mathsf{rt}(G_{[u]_f \cup [v]_f}, f_{u,v}) \leq 2 \,\}$$

can be computed in polynomial time by Proposition 1. It is clear that $\mathsf{rt}(G, f) \leq 2$ 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 [5]. □

One can calculate the number of 2-step solutions in $\mathsf{RT}(K_n, f)$ for any configuration f on the complete graph K_n using Petersen and Tenner's formula [16]. However, it is hard for general graphs.

Theorem 6 *It is a #P-complete problem to calculate the number of 2-step solutions in* $\mathsf{RT}(G, f)$ *for bipartite graphs* G*.*

Proof: We show the theorem by a reduction from the problem of calculating the number of perfect matchings in a bipartite graph H, which is known to be #P-complete [17]. For a graph $H = (V, E)$, 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 $u \in V$. If $\langle S, T\rangle \in \mathsf{RT}(G, f)$, then for each $u \in V$ there is $v \in V$ such that $\{u, v\} \in E$ and either $\{u_1, v_1\}, \{u_2, v_2\} \in S$ and $\{u_1, v_2\}, \{u_2, v_1\} \in T$ or $\{u_1, v_2\}, \{u_2, v_1\} \in S$ and $\{u_1, v_1\}, \{u_2, v_2\} \in T$. Then it is easy to see that $\mathsf{RT}(G, f)$ has 2^m 2-step solutions if H has m perfect matchings. Note that if H is bipartite, then so is G. □

3.2.2 Approximation Algorithm for the Permutation Routing via Matching on Paths

We present an approximation algorithm for the Permutation Routing via Matching on paths which outputs a parallel swap sequence whose length is no more than $\mathsf{rt}(P_n, f) + 1$, where $P_n = (\{\,1, \dots, n\}, \{\,\{i, i+1\} \mid 1 \leq i < n\,\})$ and f is a configuration on P_n. We say that a swap $\{i, i+1\}$ 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 $i \geq 1$, and for each even number j, all swaps in S_j are of the form $\{2i, 2i+1\}$ for some $i \geq 1$. Our algorithm computes a reasonable odd-even parallel swap sequence in a greedy manner.

Lemma 4 *Suppose that* $g = fS$ *for a reasonable parallel swap* S *w.r.t.* f*. For any* $\langle S_1, \dots, S_m\rangle \in \mathsf{RT}(P_n, f)$*, there is* $\langle S'_1, \dots, S'_m\rangle \in \mathsf{RT}(P_n, g)$ *such that* $S'_j \subseteq S_j$ *for all* $j \in \{1, \dots, m\}$*.*

Algorithm 3 Approximation algorithm for Permutation Routing via Matching on paths

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

Proof: It is enough to show the lemma for the case where $|S| = 1$. Suppose that $S = \{\{i, i+1\}\}$ with $f(i) > f(i+1)$. By $\langle S_1, \dots, S_m\rangle \in \mathsf{RT}(P_n, f)$, at some step we must exchange the positions of the tokens $f(i)$ and $f(i+1)$ in $\langle S_1, \dots, S_m\rangle$. Let k be the least number such that $\{f_k^{-1}(f(i)), f_k^{-1}(f(i+1))\} \in S_k$ where $f_k = f\langle S_1, \dots, S_k\rangle$. Define $S'_k = S_k - \{\{f_k^{-1}(f(i)), f_k^{-1}(f(i+1))\}\}$ and $S'_j = S_j$ for all the other $j \in \{1, \dots, m\} - \{k\}$. Then for any $j \in \{1, \dots, m\}$, f_j and $g_j = g\langle S'_1, \dots, S'_j\rangle$ are identical except when $j < k$ the positions of tokens $f(i)$ and $f(i+1)$ are switched. □

Let us denote the output of Algorithm 3 by $\mathsf{AP}(P_n, f_0)$. Clearly $\mathsf{AP}(P_n, f_0) \in \mathsf{RT}(P_n, f_0)$.

Corollary 1 *For any odd-even solution $\vec{S} \in \mathsf{RT}(P_n, f_0)$, we have $|\mathsf{AP}(P_n, f_0)| \le |\vec{S}|$.*

Proof: It is obvious that $\mathsf{AP}(P_n, f_0) \in \mathsf{RT}(P_n, f_0)$ and it is odd-even. Suppose that $\vec{S} = \langle S_1, \dots, S_m\rangle \neq \mathsf{AP}(P_n, f_0)$. Without loss of generality we may assume that $\vec{S}$ is reasonable. Let $\vec{T} = \langle T_1, \dots, T_k\rangle = \mathsf{AP}(P_n, f_0)$. If $m \ge k$, we have done. Suppose $m < k$. Since the proper prefix $\langle T_1, \dots, T_m\rangle$ of $\vec{T}$ is not a solution, there must exist $j \le m$ such that $S_1 = T_1, \dots, S_{j-1} = T_{j-1}$ and $S_j \neq T_j$. Since $\vec{S}$ is reasonable and Algorithm 3 is greedy, $S_j \subsetneq T_j$ holds. Applying Lemma 4 to $f_j = f_0\langle S_1, \dots, S_j\rangle$ and $S = T_j - S_j$, we obtain $S'_{j+1} \subseteq S_{j+1}, \dots, S'_m \subseteq S_m$ such that $\langle S_1, \dots, S_{j-1}, T_j, S'_{j+1}, \dots, S'_m\rangle \in \mathsf{RT}(P_n, f_0)$. By definition the new solution $\vec{S}\,' = \langle T_1, \dots, T_{j-1}, T_j, S'_{j+1}, \dots, S'_m\rangle$ is odd-even. Hence one can apply the same argument to $\vec{S}\,'$ and finally get $\langle T_1, \dots, T_m\rangle \in \mathsf{RT}(P_n, f_0)$. □

Theorem 7 $|\mathsf{AP}(P_n, f_0)| \le \mathsf{rt}(P_n, f_0) + 1$.

Proof: By Corollary 1, it is enough to show that every swap sequence $\vec{S} = \langle S_1, \dots, S_m\rangle$ admits an equivalent odd-even sequence $\vec{S}\,'$ such that $|\vec{S}\,'| \le |\vec{S}| + 1$. 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 $\mathsf{Œ}(\vec{S}) = \langle S'_1, \dots, S'_{m+1}\rangle$ 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 $f\vec{S} = f\text{Œ}(\vec{S})$. Therefore, for an optimal reasonable solution $\vec{S}_0$, we have $|\vec{S}_0| + 1 = |\text{Œ}(\vec{S}_0)| \geq |\mathsf{AP}(P_n, f_0)|$. □

Example 3 Let us consider the initial configuration $f_0 : \langle 3, 2, 5, 1, 7, 6, 4\rangle$ on P_7, where we express a configuration f as a sequence $\langle f(1), \dots, f(7)\rangle$. According to the output by Algorithm 3, the configuration changes as follows:

$$\begin{aligned} f_0 &: \langle \underline{3, 2}, \underline{5, 1}, \underline{7, 6}, 4\rangle\,, \\ f_1 &: \langle 2, \underline{3, 1}, 5, 6, \underline{7, 4}\rangle\,, \\ f_2 &: \langle \underline{2, 1}, 3, 5, \underline{6, 4}, 7\rangle\,, \\ f_3 &: \langle 1, 2, 3, \underline{5, 4}, 6, 7\rangle\,, \\ f_4 &: \langle 1, 2, 3, 4, 5, 6, 7\rangle\,, \end{aligned}$$

than which an optimal swapping sequence is shorter by one:

$$\begin{aligned} f_0 &: \langle \underline{3, 2}, \underline{5, 1}, 7, \underline{6, 4}\rangle\,, \\ f'_1 &: \langle 2, \underline{3, 1}, 5, \underline{7, 4}, 6\rangle\,, \\ f'_2 &: \langle \underline{2, 1}, 3, \underline{5, 4}, \underline{7, 6}\rangle\,, \\ f'_3 &: \langle 1, 2, 3, 4, 5, 6, 7\rangle\,. \end{aligned}$$

4 Coloring Routing via Matching

Colored Token Swapping is a generalization of Token Swapping, 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 Token Swapping, obviously it is NP-hard. Yamanaka et al. [20] have investigated subcases of Colored Token Swapping called c-Colored Token Swapping where the codomain of colorings is restricted to $\{1, \dots, c\}$. Along this line, we discuss the colored version of Permutation Routing via Matching in this section.

Problem 4 (c-Coloring Routing via Matching)

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 $\mathsf{rt}(G, f, g) = \min\{\, |\vec{S}| \mid f\vec{S} = g \,\}$ for two consistent colorings f and g. Since $\mathsf{rt}(G, f, g)$ can be bounded by $\mathsf{rt}(G, h)$ for some configuration h, the c-Coloring Routing via Matching belongs to NP.

4.1 Hardness of the c-Coloring Routing via Matching

Yamanaka et al. [20] have shown that the 3-Colored Token Swapping 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-Coloring Routing via Matching. We then obtain the following theorem as a corollary to their discussion.

Theorem 8 *To decide whether $\mathsf{rt}(G, f, g) \le 3$ 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 2-Colored Token Swapping is solvable in polynomial time on the other hand. In contrast, we prove that the 2-Coloring Routing via Matching is still NP-hard.

Theorem 9 *For any fixed $k \ge 3$, to decide whether $\mathsf{rt}(G, f, g) \le k$ is NP-hard for a bipartite graph G with maximum vertex degree 4 and 2-colorings f and g.*

Proof: We prove the theorem by a reduction from Sep-SAT. We use the same graph used in the proof of Lemma 3 to show the theorem for $k = 3$. For $k > 3$, the technique used in Theorem 3 can be used. 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 $C_j \in F_2$. Figure 5 illustrates the gadget related to a variable x_1 that occurs positively in $C_1 \in F_1$, $C_2 \in F_2$ and negatively in $C_3 \in F_3$, where each vertex w with $f(w) = 2$ has a black box on it and one with $g(w) = 2$ is represented with a bold rim.

If F is satisfiable, then exactly the same parallel swap sequence in the proof of Lemma 3 witnesses $\mathsf{rt}(G_F, f, g) \le 3$. It is enough to show that if $g = f\vec{S}$ with $|\vec{S}| \le 3$, then the token colored 2 on u_i is moved to u'_i for each $x_i \in X$, the one on v_j is moved to v'_j for $C_j \in F_1 \cup F_3$, and the one on v'_j is moved to v_j for $C_j \in F_2$. The token on v_j must go to a vertex w such that $g(w) = 2$ and $\mathsf{dist}(v_j, w) \le 3$. For $C_j \in F_1 \cup F_3$, the only vertex that meets the condition is v'_j. On the other hand, for each $C_j \in F_2$, the vertex v_j requires a token colored 2 moved from somewhere w, i.e., $f(w) = 2$ and $\mathsf{dist}(v_j, w) \le 3$. The only possibility is the vertex v'_j. Therefore, the token on u_i for $i \in \{1, \dots, m\}$ can be moved to neither v'_j nor v_j for any $j \in \{1, \dots, n\}$. The unique possible destination of u_i is u'_i. □

We can also show the following using the ideas for proving Theorems 4 and 9.

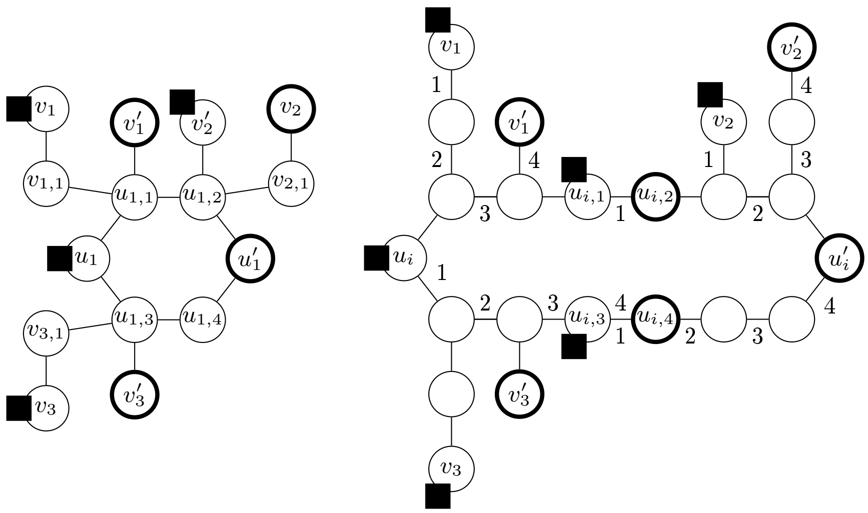

Figure 5: Gadgets used to show Theorems 9 (left) and 10 (right). We must convey all black boxes (tokens of color 2) to marked vertices (vertices of color 2) via matching.

Theorem 10 *For any fixed $k \geq 4$, to decide whether $\mathsf{rt}(G, f, g) \leq k$ is NP-hard even if G is a bipartite graph with maximum vertex degree 3 and f and g are 2-colorings.*

Proof: The theorem is shown based on the reduction from Sep-SAT used for Theorems 3, 4 and 9 again. The gadget we use for this theorem is shown on the right in Figure 5 for $k = 4$, where we suppose $x_i \in C_1 \in F_1$, $x_i \in C_2 \in F_2$, and $\neg x_i \in C_3 \in F_3$. Each vertex w with a black box satisfies $f(w) = 2$ and one represented with a bold rim satisfies $g(w) = 2$. Otherwise $f(w) = g(w) = 1$. Just like we did to show Theorem 4, this gadget is obtained from the one used in the proof of Theorem 9 by splitting every vertex of degree 4 into three so that those vertices have degree at most 3. This lengthens the paths from v_j to v'_j, which also makes the distance between u_i and u'_i even bigger: namely $\mathsf{dist}(u_i, u'_i) = 7$. We give the color 2 to vertices $u_{i,2}$ and $u_{i,4}$ in the middle of the two paths between u_i and u'_i so that the token of color 2 on u_i in the initial configuration can reach either of the destinations in 4 steps. Symmetrically, we put tokens of color 2 on $u_{i,1}$ and $u_{i,3}$ in the initial configuration, one of which should go to the vertex u'_i. The non-uniqueness of the destinations of tokens in Coloring Routing via Matching enables us to lower the number of steps required to realize the goal configuration by one comparing to Theorem 4 for Permutation Routing via Matching.

Suppose that F is satisfied by some ϕ. If $\phi(x_i) = 0$, then the vertex on u_i, $u_{i,1}$ and $u_{i,3}$ will go to $u_{i,2}$, u'_i and $u_{i,4}$, respectively. Otherwise, they will go to $u_{i,4}$, $u_{i,2}$ and u'_i. Then each v_j can be moved to v'_j within 4 steps using an edge on a u_i-u'_i path if either $x_i \in C_j \in F_1 \cup F_2$ and $\phi(x_i) = 1$ or $\neg x_i \in C_j \in F_3$

and $\phi(x_i) = 0$. Figure 5 illustrates the case where $\phi(x_i) = 1$, $x_i \in C_1 \in F_1$, $x_i \in C_2 \in F_2$ and $\neg x_i \in C_3 \in F_3$. Numbers labeling edges show when they are used in a 4-step solution $\langle S_1, \dots, S_4 \rangle$.

On the other hand, suppose that $\mathsf{rt}(G, f, g) \le 4$. Considering the destination of the token on the vertex v_j for $C_j \in F_1 \cup F_3$, the unique vertex w such that $f(v_j) = g(w)$ and $\mathsf{dist}(v_j, w) \le 4$ is $w = v'_j$. Similarly, considering the vertex v_j for $C_j \in F_2$, the unique vertex w such that $g(v'_j) = f(w)$ and $\mathsf{dist}(v'_j, w) \le 4$ is $w = v_j$. Therefore, all the tokens on v_j are moved to the vertex v'_j. This means that the only possible destinations of the token on u_i are $u_{i,2}$ and $u_{i,4}$. If u_i is moved to $u_{i,2}$, then the only possible destination of the token on $u_{i,1}$ is u'_i, and thus the token on $u_{i,3}$ must go to $u_{i,4}$. In this case, v_j for $x_i \in C_j \in F_1 \cup F_2$ cannot go to v'_j via the edge on the u_i-u'_i path. In other words, there must be h such that $x_h \in C_j$, u_h is moved to $u_{h,4}$, and v_j is moved to v'_j via the edge on the u_h-u'_h path. The exactly symmetric argument holds when u_i is moved to $u_{i,4}$. It is now clear that F is satisfied by ϕ such that $\phi(x_i) = 0$ if and only if the token on u_i goes to $u_{i,2}$.

The theorem for $k > 4$ can be shown by inserting extra paths into appropriate places like we did in the proof of Theorem 3. □

The next theorem contrasts the result of Theorem 5, which shows that it is polynomial-time decidable whether a 2-step solution exists in Permutation Routing.

Theorem 11 *It is NP-hard to decide whether* $\mathsf{rt}(G, f, g) \le 2$ *for a graph* G *of vertex degree at most* 4 *and* 3*-colorings* f *and* g*.*

Proof: The proof of this theorem again uses a variant of the reduction used in the proof of Lemma 3 and Theorem 9. To make a 2-step solution possible, we contract edges $\{v_j, v_{j,i}\}$ for $x_i \in C_j \in F_1 \cup F_2$ and $\neg x_i \in C_j \in F_3$ in the graph defined in Section 3.1.[3] That is, we define $G_F = (V_F, E_F)$ so that

$$V_F = \{\, u_i, u'_i, u_{i,1}, u_{i,2}, u_{i,3}, u_{i,4} \mid 1 \le i \le m \,\} \cup \{\, v_j, v'_j \mid 1 \le j \le n \,\}$$

and E_F contains the following paths

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

The initial and goal 3-colorings f and g are respectively given as

	u_i	u'_i	$u_{i,1}$	$u_{i,2}$	$u_{i,3}$	$u_{i,4}$	v_j	v'_j	v_k	v'_k
f	2	1	1	2	1	2	3	1	3	2
g	1	2	1	2	1	2	1	3	2	3

for $i \in \{1, \dots, m\}$, $C_j \in F_1 \cup F_3$ and $C_k \in F_2$. Figure 6 illustrates the reduction where the values of f and g are shown in rectangles and circles, respectively.

[3] Actually one can show Lemma 3 using this simplified graph, too.

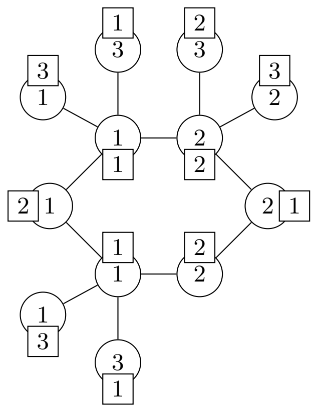

Figure 6: Gadget used to show Theorem 11

Suppose that F is satisfied by an assignment ϕ. There is a function $\psi : F \to X$ such that $\psi(C_j) \in C_j$ and $\phi(\psi(C_j)) = 1$ if $C_j \in F_1 \cup F_2$, and $\neg\psi(C_j) \in C_j$ and $\phi(\psi(C_j)) = 0$ if $C_j \in F_3$. Define

$$\begin{aligned} S_1 &= \{\, \{u_i, u_{i,1}\}, \{u'_i, u_{i,2}\} \mid \phi(x_i) = 0 \,\} \cup \{\, \{u_i, u_{i,3}\}, \{u'_i, u_{i,4}\} \mid \phi(x_i) = 1 \,\} \\ &\quad \cup \{\, \{v_j, u_{i,k}\} \mid \psi(C_j) = x_i \text{ and } C_j \in F_k \,\}, \\ S_2 &= \{\, \{u_{i,1}, u_{i,2}\} \mid \phi(x_i) = 0 \,\} \cup \{\, \{u_{i,3}, u_{i,4}\} \mid \phi(x_i) = 1 \,\} \\ &\quad \cup \{\, \{v'_j, u_{i,k}\} \mid \psi(C_j) = x_i \text{ and } C_j \in F_k \,\}. \end{aligned}$$

It is easy to see that $g = f\langle S_1, S_2\rangle$.

We now suppose the converse, $\langle S_1, S_2\rangle \in \mathsf{RT}(G_F, f, g)$. For each u_i, the only vertices w such that $\mathsf{dist}(u_i, w) \le 2$ and $f(u_i) = g(w) = 2$ are $u_{i,2}$ and $u_{i,4}$. The only u_i-$u_{i,k}$-path of length at most 2 is $(u_i, u_{i,k-1}, u_{i,k})$ for $k \in \{2, 4\}$. Thus, either $\{u_i, u_{i,1}\} \in S_1$ and $\{u_{i,1}, u_{i,2}\} \in S_2$ or $\{u_i, u_{i,3}\} \in S_1$ and $\{u_{i,3}, u_{i,4}\} \in S_2$. We will show that ϕ defined by

$$\phi(x_i) = \begin{cases} 0 & \text{if } \{u_i, u_{i,1}\} \in S_1, \\ 1 & \text{if } \{u_i, u_{i,3}\} \in S_1 \end{cases}$$

satisfies F. Each v_j with $j \in \{1, \ldots, n\}$ has only one vertex w such that $\mathsf{dist}(v_j, w) \le 2$ and $f(v_j) = g(w) = 3$, which is $w = v'_j$. The paths of length at most 2 between v_j and v'_j are of the form $(v_j, u_{i,k}, v'_j)$ for some i and k such that $x_i \in C_j \in F_k$ or $\neg x_i \in C_j \in F_k$. For those i and k, $\{v_j, u_{i,k}\} \in S_1$ and $\{v'_j, u_{i,k}\} \in S_2$ holds. Suppose $C_j \in F_1$. In this case, $\{u_i, u_{i,1}\} \notin S_1$, which implies $\phi(x_i) = 1$. By $x_i \in C_j$, C_j is satisfied. Suppose $C_j \in F_2$. Then $\{u_{i,1}, u_{i,2}\} \notin S_2$, which implies $\{u_i, u_{i,1}\} \notin S_1$ and $\phi(x_i) = 1$. By $x_i \in C_j$, C_j is satisfied. Suppose $C_j \in F_3$. In this case, $\{u_i, u_{i,3}\} \notin S_1$, which implies $\phi(x_i) = 0$. By $\neg x_i \in C_j$, C_j is satisfied. □

4.2 2-Step 2-Coloring Routing via Matching Is Easy

In the previous subsection we have shown that c-Coloring Routing via Matching is hard even to decide whether a k-step solution exists if $c \geq 3$ and $k \geq 2$ or $c \geq 2$ and $k \geq 3$. We will show that it is easy if $c, k \leq 2$. Suppose that $\langle S_1, S_2 \rangle$ is a 2-step solution for (G, f, g) where f and g are consistent 2-colorings on $G = (V, E)$. We say that a swap $\{u, v\}$ is *vacuous* for f if $f(u) = f(v)$.

Lemma 5 *If (G, f, g) admits a 2-step solution, there is $\langle S_1, S_2 \rangle \in \mathsf{RT}(G, f, g)$ such that*

- $S_1 \cap S_2 = \varnothing$,
- *no swaps in S_1 and in S_2 are vacuous for f and for fS_1, respectively,*
- *$S_1 \cup S_2$ gives a path matching in G.*

Proof: The first two items are trivial. Assuming $\langle S_1, S_2 \rangle$ satisfies the first two, we show the last. If $\{u, v\}, \{v, w\} \in S_1 \cup S_2$ with $v \neq w$, then either $\{u, v\} \in S_1$ and $\{v, w\} \in S_2$ or $\{u, v\} \in S_2$ and $\{v, w\} \in S_1$. This implies that $G' = (V, S_1 \cup S_2)$ has degree bound 2. Moreover, if G' has a cycle, then the size must be even. We show that if G' contains a cycle $(u_1, v_1, u_2, \ldots, u_n, v_n, u_1)$, then $f(u_i) = g(u_i)$ and $f(v_i) = g(v_i)$ for all i. That is, those edges in the cycle can be removed from S_1 and S_2. Hereafter, by u_j we mean u_i such that $1 \leq i \leq n$ for $i \equiv j \pmod{n}$. Without loss of generality, assume $\{u_i, v_i\} \in S_1$ for all i and $\{v_i, u_{i+1}\} \in S_2$ for all i. Since $\{u_i, v_i\} \in S_1$ is not vacuous for f, $f(u_i) \neq f(v_i)$ for all i. Since $\{v_i, u_{i+1}\} \in S_2$ is not vacuous for fS_1, $fS_1(v_i) \neq fS_1(u_{i+1})$, i.e., $f(u_i) \neq f(v_{i+1})$ for all i. Hence, $f(v_i) = f(v_{i+1})$ for all i. Moreover, $g(v_i) = f\langle S_1, S_2 \rangle(v_i) = f(v_{i+1})$ for all i. That is, $f(v_i) = g(v_i)$ for all i. Similarly we have $f(u_i) = g(u_i)$ for all i. Those tokens need not be moved at all. □

Hereafter we consider only 2-step solutions that satisfy the condition of Lemma 5.

Lemma 6 *Let $(u_1, \ldots, u_n)$ be a (maximal) path in $G' = (V, S_1 \cup S_2)$ for a 2-step solution $\langle S_1, S_2 \rangle \in \mathsf{RT}(G, f, g)$ satisfying the condition of Lemma 5. If $n = 2$, then $f(u_1) = g(u_2) \neq f(u_2) = g(u_1)$. If $n \geq 3$,*

- *for all $i \in \{2, \ldots, n-2\}$, $f(u_i) = g(u_i) \neq f(u_{i+1}) = g(u_{i+1})$,*
- *if $\{u_1, u_2\} \in S_1$ then $f(u_1) \neq f(u_2)$ and $g(u_1) = g(u_2)$,*
- *if $\{u_1, u_2\} \in S_2$ then $f(u_1) = f(u_2)$ and $g(u_1) \neq g(u_2)$,*
- $f(u_1) = g(u_n) \neq g(u_1) = f(u_n)$.

Proof: For $n = 2$, the lemma holds trivially. We assume $n \geq 3$. For readability, we rename each vertex u_i by i. Suppose $\{1, 2\} \in S_1$, which implies $\{2i+1, 2i+2\} \in S_1$ for $0 \leq i \leq \lfloor (n-2)/2 \rfloor$ and $\{2i+2, 2i+3\} \in S_2$ for $0 \leq i \leq \lfloor (n-3)/2 \rfloor$. Since $\{2i+1, 2i+2\} \in S_1$ is not vacuous for f, $f(2i+1) \neq f(2i+2)$ for $0 \leq i \leq \lfloor (n-2)/2 \rfloor$. Since $\{2i+2, 2i+3\} \in S_2$

is not vacuous for fS_1, $fS_1(2i+2) \neq fS_1(2i+3)$, i.e., $f(2i+1) \neq f(2i+4)$ for $0 \leq i \leq \lfloor (n-4)/2 \rfloor$. That is,

$$f(1) \neq f(2) \neq \cdots \neq f(n-1)\,.$$

By $g(2i+2) = f\langle S_1, S_2\rangle(2i+2) = fS_1(2i+3) = f(2i+4)$ for $0 \leq i \leq \lfloor (n-4)/2 \rfloor$, $g(2i+2) = f(2i+2)$ for $0 \leq i \leq \lfloor (n-4)/2 \rfloor$. By $g(2i+3) = f\langle S_1, S_2\rangle(2i+3) = fS_1(2i+2) = f(2i+1)$ for $0 \leq i \leq \lfloor (n-3)/2 \rfloor$, $g(2i+3) = f(2i+1)$ for $0 \leq i \leq \lfloor (n-3)/2 \rfloor$. On the other hand, $g(1) = f\langle S_1, S_2\rangle(1) = fS_1(1) = f(2) \neq f(1)$. Therefore,

$$f(1) \neq g(1) = f(2) = g(2) \neq f(3) = g(3) \neq \cdots \neq f(n-1) = g(n-1)\,.$$

We have shown the first and second items of the lemma. Recall that $\langle S_1, S_2\rangle \in \mathsf{RT}(G, f, g)$ implies $\langle S_2, S_1\rangle \in \mathsf{RT}(G, g, f)$ and moreover, $\langle S_2, S_1\rangle$ satisfies the condition of Lemma 5. This symmetry proves the third. The fourth is a corollary to those three. The second and third items imply $f(1) \neq g(1)$, by $f(2) = g(2)$. By the symmetry, $f(n) \neq g(n)$. Since f and g restricted to the path $\{1, \ldots, n\}$ are consistent, it must hold $f(1) = g(n)$ and $f(n) = g(1)$. □

Lemma 7 *Let* $P_n = (\{1, \ldots, n\}, \{\, \{i, i+1\} \mid 1 \leq i < n \,\})$. *If*

- *for all* $i \in \{2, \ldots, n-2\}$, $f(i) = g(i) \neq f(i+1) = g(i+1)$,
- $f(1) = g(n) \neq g(1) = f(n)$,

then (P_n, f, g) *admits a 2-step solution.*

Proof: Let $S_1 = \{\, \{2i+1, 2i+2\} \mid 0 \leq i \leq \lfloor (n-2)/2 \rfloor \,\}$ and S_2 be the rest. If $f(1) \neq f(2)$, $\langle S_1, S_2\rangle$ is a solution. If $f(1) = f(2)$, $\langle S_2, S_1\rangle$ is a solution. □

We will reduce the concerned problem to the Vertex-Disjoint Path Problem, which can be solved in polynomial-time [18].

Problem 5 (Vertex-Disjoint Path Problem)

Instance: A directed graph $G = (V, E)$, two distinguished vertices $s, t \in V$ and $k \in \mathbb{N}$.

Question: Are there k s-t-paths in G which are vertex-disjoint except s and t?

For a given instance (G, f, g) with $G = (V, E)$ of 2-Coloring Routing via Matching, we give an instance (H, s, t, k) of the Vertex-Disjoint Path Problem as follows. Let us partition V into

$$\begin{aligned} V_s &= \{\, u \mid f(u) = 1 \text{ and } g(u) = 2 \,\}\,, \\ V_t &= \{\, u \mid f(u) = 2 \text{ and } g(u) = 1 \,\}\,, \\ V_1 &= \{\, u \mid f(u) = g(u) = 1 \,\}\,, \\ V_2 &= \{\, u \mid f(u) = g(u) = 2 \,\} \end{aligned}$$

and define $H = (V', F)$ by $V' = V \cup \{s, t\}$ and

$$F = \{ (u, v) \in (V_s \times V_t) \cup (V_s \times (V_1 \cup V_2)) \cup ((V_1 \cup V_2) \times V_t) \\ \cup (V_1 \times V_2) \cup (V_2 \times V_1) \mid \{u, v\} \in E \} \cup (\{s\} \times V_s) \cup (V_t \times \{t\}) .$$

Lemma 8 *(G, f, g) admits a 2-step solution if and only if (H, s, t) admits $|V_s|$ disjoint paths.*

Proof: Suppose $\langle S_1, S_2 \rangle \in \mathsf{RT}(G, f, g)$, which satisfies the condition of Lemma 5. The graph $G' = (V, S_1 \cup S_2)$ consists of exactly $|V_s|$ disjoint paths by Lemma 6. Clearly (H, s, t) has corresponding $|V_s|$ disjoint s-t-paths.

Suppose (H, s, t) admits $|V_s|$ disjoint s-t-paths. For each path $(s, u_1, \ldots, u_n, t)$, $(u_1, \ldots, u_n)$ satisfies the condition of Lemma 7. Since those paths are disjoint, (G, f, g) admits a 2-step solution. □

Theorem 12 *It is decidable in polynomial time if $\mathsf{rt}(G, f, g) \leq 2$ for consistent 2-colorings f and g on a graph G.*

Acknowledgement

The authors are deeply grateful to Dana Richards for making us aware of existing studies on Permutation Routing via Matching including their manuscripts. We also would like to thank the anonymous reviewers of the preliminary conference version [12] and of this journal version.

References

[1] N. Alon, F. R. K. Chung, and R. L. Graham. Routing permutations on graphs via matchings. *SIAM Journal on Discrete Mathematics*, 7(3):513–530, 1994. doi:10.1137/S0895480192236628.

[2] I. Banerjee and D. Richards. New results on routing via matchings on graphs. In R. Klasing and M. Zeitoun, editors, *Fundamentals of Computation Theory*, pages 69–81. Springer Berlin Heidelberg, 2017. doi:10.1007/978-3-662-55751-8_7.

[3] É. Bonnet, T. Miltzow, and P. Rzążewski. Complexity of token swapping and its variants. *Algorithmica*, 80(9):2656–2682, 2018. doi:10.1007/s00453-017-0387-0.

[4] S. A. Cook. The complexity of theorem-proving procedures. In M. A. Harrison, R. B. Banerji, and J. D. Ullman, editors, *Proc. of the 3rd Annual ACM Symposium on Theory of Computing*, pages 151–158. ACM, 1971. doi:10.1145/800157.805047.

[5] J. Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965. doi:10.4153/CJM-1965-045-4.

[6] S. Even and O. Goldreich. The minimum-length generator sequence problem is NP-hard. *Journal of Algorithms*, 2(3):311–313, 1981. doi:10.1016/0196-6774(81)90029-8.

[7] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[8] L. S. Heath and J. P. C. Vergara. Sorting by short swaps. *Journal of Computational Biology*, 10(5):775–789, 2003. doi:10.1089/106652703322539097.

[9] M. Jerrum. The complexity of finding minimum-length generator sequences. *Theoretical Computer Science*, 36:265–289, 1985. doi:10.1016/0304-3975(85)90047-7.

[10] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Proc. of a symposium on the Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972. URL: http://www.cs.berkeley.edu/~luca/cs172/karp.pdf.

[11] J. Kawahara, T. Saitoh, and R. Yoshinaka. A note on the complexity of the token swapping problem (in Japanese). *IPSJ SIG Technical Report*, 2015-AL-156(3):1–7, January 2016.

[12] J. Kawahara, T. Saitoh, and R. Yoshinaka. The time complexity of the token swapping problem and its parallel variants. In S. Poon, M. S. Rahman,

and H. Yen, editors, *11th International Conference and Workshops on Algorithms and Computation*, volume 10167 of *Lecture Notes in Computer Science*, pages 448–459. Springer, 2017. doi:10.1007/978-3-319-53925-6_35.

[13] W. Li, L. Lu, and Y. Yang. Routing numbers of cycles, complete bipartite graphs, and hypercubes. *SIAM Journal on Discrete Mathematics*, 24(4):1482–1494, 2010. doi:10.1137/090776317.

[14] T. Miltzow, L. Narins, Y. Okamoto, G. Rote, A. Thomas, and T. Uno. Approximation and hardness of token swapping. In P. Sankowski and C. D. Zaroliagis, editors, *24th Annual European Symposium on Algorithms*, volume 57 of *LIPIcs*, pages 66:1–66:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPIcs.ESA.2016.66.

[15] I. Pak. Reduced decompositions of permutations in terms of star transpositions, generalized catalan numbers and k-ARY trees. *Discrete Mathematics*, 204(1-3):329–335, 1999. doi:10.1016/S0012-365X(98)00377-X.

[16] T. K. Petersen and B. E. Tenner. How to write a permutation as a product of involutions (and why you might care). *Integers*, 13(#A63), 2013.

[17] L. G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8:189–201, 1979. doi:10.1016/0304-3975(79)90044-6.

[18] Z. Xie, H. Leng, Z. Chen, and J. Zhang. Finding arc and vertex-disjoint paths in networks. In W. Z. Bo Yang, Y. Dai, L. T. Yang, and J. Ma, editors, *8th IEEE International Conference on Dependable, Autonomic and Secure Computing*, pages 539–544. IEEE Computer Society, 2009. doi:10.1109/DASC.2009.75.

[19] 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*, 586:81–94, 2015. doi:10.1016/j.tcs.2015.01.052.

[20] K. Yamanaka, T. Horiyama, J. M. Keil, D. G. Kirkpatrick, Y. Otachi, T. Saitoh, R. Uehara, and Y. Uno. Swapping colored tokens on graphs. *Theoretical Computer Science*, 729:1–10, 2018. doi:10.1016/j.tcs.2018.03.016.

[21] L. Zhang. Optimal bounds for matching routing on trees. *SIAM Journal on Discrete Mathematics*, 12(1):64–77, 1999. doi:10.1137/S0895480197323159.

A Proof that Token Swapping on Lollipop Graphs Is in P

This appendix gives a proof that Algorithm 1 computes an optimal swapping sequence on lollipop graphs. We will give an evaluation function on configurations on lollipop graphs $L_{m,n}$ such that any swap changes the value by one, every swap by the algorithm reduces the value by one, and the value is 0 if and only if the configuration is the identity. Algorithm 1 first moves non-negative tokens to the goal vertices on the path and then moves negative ones in the clique. The number of swaps needed to move a token $j \in \{0, \dots, n\}$ is evaluated by

$$\pi(f,j) = \begin{cases} j+1 & \text{if } f^{-1}(j) < 0, \\ \min(j+1, \, \mathrm{Inv}(f,j)) & \text{if } f^{-1}(j) \geq 0, \end{cases}$$

where

$$\mathrm{Inv}(f,j) = |\{\, i \mid i < j \text{ and } f^{-1}(i) > f^{-1}(j) \,\}| \,.$$

So it takes

$$\pi(f) = \sum_{j=0}^{n} \pi(f,j)$$

swaps to move the non-negative tokens to the goal vertices in total. We then move the negative tokens in the clique. For a configuration f' such that $f'(j) = j$ for all $j \geq 0$, the number of swaps needed is

$$\nu(f') = m - |\Lambda_{f'}| \text{ where } \Lambda_{f'} = \{\, [i]_{f'} \mid i < 0 \,\}\,. \text{ (See e.g. [9])}$$

We need to evaluate $|\Lambda_{f'}|$ for $f' = f\vec{S}$ where $\vec{S}$ moves all the non-negative tokens to their goals. Let us call an injection f from $\{-m, \dots, k\}$ to $\{-m, \dots, n\}$ for some $k \in \{-1, 0, \dots, n\}$ a *pseudo configuration* if the range of f includes $\{-m, \dots, -1\}$. For notational simplicity, a pseudo configuration f will often be identified with the sequence $\langle f(-1), \dots, f(-m), f(0), \dots, f(k)\rangle$ or the sequence pair $(\langle f(-1), \dots, f(-m)\rangle; \langle f(0), \dots, f(k)\rangle)$. For a pseudo configuration $(\vec{i}; \vec{j})$ where $\vec{i} = \langle f(-1), \dots, f(-m)\rangle$ and $\vec{j} = \langle f(0), f(1), \dots, f(k)\rangle$, we define ν recursively on $|\vec{j}|$ by

$$\nu(\vec{i}; \vec{j}) = \begin{cases} m - |\Lambda_f| & \text{if } \vec{j} \text{ is empty}, \\ \nu(\vec{i}; \langle f(1), \dots, f(k)\rangle) & \text{if } c < f(0), \\ \nu(\vec{i}[f(0)/c]; \langle f(1), \dots, f(k)\rangle) & \text{if } c > f(0), \end{cases}$$

where $c = \max(\vec{i})$ and $[a/b]$ replaces b by a. That is,

$$f[a/b](i) = \begin{cases} f(i) & \text{if } f(i) \neq b, \\ a & \text{if } f(i) = b. \end{cases}$$

Note that if $c = \max(\vec{i}) > f(0)$, then $c \geq 0$ and thus $(\vec{i}[f(0)/c]; \langle f(1), \dots, f(k)\rangle)$ is a pseudo configuration. Our evaluation function Φ is given as

$$\Phi(f) = \pi(f) + \nu(f)\,.$$

Note that ν is defined on pseudo configurations but π and Φ are defined on (proper) configurations. It is clear that $\Phi(f) \geq 0$ for any configuration and the equation holds if and only if f is the identity.

Lemma 9 *For any $\vec{i}, \vec{j}$, there is a sequence $\vec{i}'$ consisting of the m smallest elements from $\vec{i} \cdot \vec{j}$, where $\vec{i} \cdot \vec{j}$ denotes the concatenation of $\vec{i}$ and $\vec{j}$, such that for any $\vec{k}$*

$$\nu(\vec{i}; \vec{j} \cdot \vec{k}) = \nu(\vec{i}'; \vec{k}),$$

provided that $(\vec{i}; \vec{j} \cdot \vec{k})$ is a pseudo configuration.

Proof: The lemma can be shown by induction on $|\vec{j}|$ just following the definition of ν. □

Lemma 10 *If $\vec{i} \cdot \vec{j}_1$ contains m or more tokens smaller than $a \geq 0$, then*

$$\nu(\vec{i}; \vec{j}_1 \cdot a \cdot \vec{j}_2) = \nu(\vec{i}; \vec{j}_1 \cdot \vec{j}_2),$$

provided that $(\vec{i}; \vec{j}_1 \cdot a \cdot \vec{j}_2)$ is a pseudo configuration.

Proof: By induction on $|\vec{j}_1|$. □

Now we are going to prove that any possible swap on the graph changes the value of Φ by one. We have three cases depending on where a swap takes place. First we consider the case where a swap takes inside the clique.

Lemma 11 *Let $f = (\vec{i}; \vec{j})$ and $g = (\vec{i}'; \vec{j})$ be pseudo configurations such that $\vec{i}' = \vec{i}[a/b, b/a]$ for some distinct tokens a, b. Then*

$$|\nu(f) - \nu(g)| = 1.$$

Proof: We show this by induction on $|\vec{j}|$. If $\vec{j}$ is not empty, the claim follows the induction hypothesis immediately. If $\vec{j}$ is empty, f and g are configurations on the clique of $\{-1, \ldots, -m\}$.
Case 1. Suppose $[a]_f = [b]_f$. Let $k = |[a]_f|$ and $b = f^j(a)$. Then $g^i(a) = f^i(a)$ for $i < j$, $g^j(a) = a$, $g^i(b) = f^{j+i}(a)$ for $i < k - j$ and $g^{k-j}(b) = b$. That is, $[a]_f = [a]_g \cup [b]_g$, $[a]_g \neq [b]_g$ and $|\Lambda_g| = |\Lambda_f| + 1$. Hence $\nu(g) = \nu(f) - 1$.
Case 2. Suppose $[a]_f \neq [b]_f$. Let $k_a = |[a]_f|$ and $k_b = |[b]_f|$. Then $g^i(a) = f^i(a)$ for $i \in \{0, \ldots, k_a - 1\}$, $g^{k_a+i}(a) = f^i(b)$ for $i \in \{0, \ldots, k_b - 1\}$ and $g^{k_a+k_b}(a) = a$. That is, $[a]_g = [b]_g = [a]_f \cup [b]_f$ and $|\Lambda_g| = |\Lambda_f| - 1$. Hence $\nu(g) = \nu(f) + 1$. □

Corollary 2 *Suppose that $g = fe$ for some swap $e \subseteq \{-1, \ldots, -m\}$. Then $|\Phi(g) - \Phi(f)| = 1$.*

Proof: Clearly $\pi(f) = \pi(g)$ by definition. The claim follows Lemma 11. □

The following lemma is concerned with the value of Φ when a swap takes at the joint of the clique and the path.

Lemma 12 *If $g = f\{h, 0\}$ with $h < 0$, then $|\Phi(g) - \Phi(f)| = 1$.*

Proof: We may assume without loss of generality that $h = -1$ for the symmetry. Let

$$f = (a \cdot \vec{i};\, b \cdot \vec{j})\,,$$
$$g = (b \cdot \vec{i};\, a \cdot \vec{j})\,.$$

Without loss of generality we assume $a > b$.
Case 1. Suppose $a, b < 0$. Clearly $\pi(f) = \pi(g)$ and $\max(\vec{i}) \geq 0$. We have $|\nu(f) - \nu(g)| = 1$ by applying Lemma 11 to the fact

$$\nu(f) = \nu(a \cdot \vec{i}[b/c];\, \vec{j})\,,$$
$$\nu(g) = \nu(b \cdot \vec{i}[a/c];\, \vec{j})\,,$$

where $c = \max(\vec{i})$.
Case 2. Suppose $a \geq 0 > b$.
Case 2.1. Suppose $a > \max(\vec{i})$. We have

$$\nu(f) = \nu(g) = \nu(b \cdot \vec{i};\, \vec{j})\,.$$

All the m elements of $b \cdot \vec{i}$ are smaller than a, which are among $m + a$ tokens smaller than a. Therefore, $\vec{j}$ contains exactly a tokens smaller than a, which means $\pi(g, a) = a$. On the other hand, $\pi(f, a) = a + 1$ by definition. For all other positive tokens k, $\pi(f, k) = \pi(g, k)$ holds.

All in all, $\Phi(f) - \Phi(g) = 1$.
Case 2.2. Suppose $\max(\vec{i}) > a$. Let $c = \max(\vec{i})$. We have

$$\nu(f) = \nu(a \cdot \vec{i}[b/c]; \vec{j})\,,$$
$$\nu(g) = \nu(b \cdot \vec{i}[a/c]; \vec{j})\,,$$

and $|\nu(f) - \nu(g)| = 1$ by Lemma 11.

It remains to show $\pi(f, k) = \pi(g, k)$ for all positive tokens k, which is clear for $k \neq a$. By definition $\pi(f, a) = a + 1$. The fact $c > a$ implies at most $m - 1$ tokens in $b \cdot \vec{i}$ are smaller than a, which are among $m + a$ tokens smaller than a. Hence $\vec{j}$ contains at least $a + 1$ tokens smaller than a, which means $\pi(g, a) = a + 1$.

All in all, $|\Phi(f) - \Phi(g)| = 1$.
Case 3. Suppose $a > b \geq 0$. This case is almost identical to Case 2 except that we need to confirm $\pi(f, b) = \pi(g, b)$ in addition. The fact $a > b$ implies at most $m - 1$ tokens in $a \cdot \vec{i}$ are smaller than b, and $\vec{j}$ contains at least $b + 1$ tokens smaller than b, which means $\pi(f, b) = \pi(g, b) = b + 1$. □

The last case we consider is when a swap takes on the path.

Lemma 13 *If $g = f\{k, k+1\}$ for some $k \geq 0$, then $|\Phi(g) - \Phi(f)| = 1$.*

Proof: Let

$$f = (\vec{i};\, \vec{j}_1 \cdot \langle a, b \rangle \cdot \vec{j}_2),$$
$$g = (\vec{i};\, \vec{j}_1 \cdot \langle b, a \rangle \cdot \vec{j}_2).$$

By Lemma 9, there exists $\vec{i}'$ consisting of the m smallest tokens from $\vec{i} \cdot \vec{j}_1$ such that

$$\nu(f) = \nu(\vec{i}';\, \langle a, b \rangle \cdot \vec{j}_2)\,,$$
$$\nu(g) = \nu(\vec{i}';\, \langle b, a \rangle \cdot \vec{j}_2)\,.$$

Without loss of generality we assume $a > b$.
Case 1. Suppose $a, b < 0$. Clearly $\pi(f) = \pi(g)$. For the two largest tokens c and d in $\vec{i}'$ with $c > d$, we have

$$\nu(f) = \nu(\vec{i}'[a/c, b/d]; \vec{j}_2)\,,$$
$$\nu(g) = \nu(\vec{i}'[b/c, a/d]; \vec{j}_2)\,.$$

Lemma 11 implies $|\Phi(f) - \Phi(g)| = 1$.
Case 2. Suppose $a \geq 0$. We have $\mathrm{Inv}(f, a) = \mathrm{Inv}(g, a) + 1$.
Case 2.1. Suppose $\mathrm{Inv}(f, a) \leq a$. In this case, we have $\pi(g, a) = \mathrm{Inv}(g, a) = \mathrm{Inv}(f, a) - 1 = \pi(f, a) - 1$ and thus $\pi(f) = \pi(g) + 1$. The fact that $b \cdot \vec{j}_2$ contains at most a tokens smaller than a implies that $\vec{i} \cdot \vec{j}_1$ contains at least m tokens smaller than a. That is, all of $\vec{i}'$ are smaller than a. By Lemma 10, we have

$$\nu(f) = \nu(g) = \nu(\vec{i}';\, b \cdot \vec{j}_2)\,.$$

All in all, $\Phi(f) = \Phi(g) + 1$.
Case 2.2. Suppose $\mathrm{Inv}(f, a) = a + 1$. In this case, we have $\pi(f, a) = a + 1$, $\pi(g, a) = \mathrm{Inv}(g, a) = a$ and thus $\pi(f) = \pi(g) + 1$. The fact that $b \cdot \vec{j}_2$ contains exactly $a + 1$ tokens smaller than a implies that $\vec{i} \cdot \vec{j}_1$ contains exactly $m - 1$ tokens smaller than a. That is, all of $\vec{i}'$ are smaller than a except one token $c = \max(\vec{i}')$. Therefore,

$$\nu(f) = \nu(\vec{i}'[a/c];\, b \cdot \vec{j}_2) = \nu(\vec{i}'[b/c];\, \vec{j}_2)\,,$$
$$\nu(g) = \nu(\vec{i}'[b/c];\, a \cdot \vec{j}_2) = \nu(\vec{i}'[b/c];\, \vec{j}_2)\,.$$

All in all, $\Phi(f) = \Phi(g) + 1$.
Case 2.3. Suppose $\mathrm{Inv}(f, a) > a + 1$. In this case, we have $\pi(f, a) = \pi(g, a) = a + 1$ and $\pi(f) = \pi(g)$. The fact that $b \cdot \vec{j}_2$ contains at least $a + 2$ tokens smaller than a implies that $\vec{i} \cdot \vec{j}_1$ contains at most $m - 2$ tokens smaller than a. That is, the two largest tokens c and d in $\vec{i}'$ are bigger than a. Therefore,

$$\nu(f) = \nu(\vec{i}'[a/c, b/d]; \vec{j}_2)\,,$$
$$\nu(g) = \nu(\vec{i}'[b/c, a/d]; \vec{j}_2)\,.$$

Lemma 11 implies $|\nu(f) - \nu(g)| = 1$. All in all, $|\Phi(f) - \Phi(g)| = 1$. □

Corollary 3 $\Phi(f) \leq \mathsf{ts}(L_{m,n}, f)$.

Proof: By Corollary 2 and Lemmas 12 and 13. □

Lemma 14 *Suppose that our algorithm changes f to g at a point in the run. Then $\Phi(g) = \Phi(f) - 1$.*

Proof: Suppose that the algorithm moves a token $a \geq 0$. If $f^{-1}(a) < 0$ then Case 2.1 of the proof of Lemma 12 applies and we have $\Phi(f) = \Phi(g) + 1$. If $f^{-1}(a) \geq 0$, the fact that $f(i) < a$ for all $i < 0$ implies that $\mathrm{Inv}(f, a) \leq a$. Hence Case 2.1 of the proof of Lemma 13 applies and we have $\Phi(f) = \Phi(g) + 1$.

Suppose that the algorithm moves a token $a < 0$. Then Case 1 of the proof of Lemma 11 applies. We conclude $\Phi(f) = \Phi(g) + 1$. □

Therefore, our algorithm gives a solution of $\Phi(f)$ steps, which is optimal by Corollary 3.

Theorem 13 *Token Swapping on lollipop graphs can be solved in polynomial time.*

B Proof that Token Swapping on Star-Path Graphs Is in P

This appendix gives a proof that Algorithm 2 computes an optimal swapping sequence on star-path graphs $Q_{m,n}$ in a manner similar to Appendix A. The number of swaps needed to move non-negative tokens to the goal vertices is evaluated by the same function π. On the other hand, the number of swaps needed to relocate negative tokens is evaluated differently from the case of lollipop graphs. The algorithm involves two types of swaps: the ones in the inner **while** loop and the others. Let us call the former Type A and the latter Type B. The negative tokens which must be moved are in $N_f = \{\, f(i) \in \{-m, \dots, -1\} \mid f(i) \neq i \,\}$. Among those, some are on a non-negative vertex and some are on a negative vertex. Tokens of the former type will be forced to move to 0 by the moves of non-negative tokens (Type B) and then go to the goal vertex by one step (Type A). Moves of Type B of those tokens are counted by π. On the other hand, tokens i of the latter type form equivalence classes $[i]_f \subseteq N_f$, which require $[i]_f + 1$ swaps to be relocated to the goal vertices. Let

$$\Delta_f = \{\, [i]_f \subseteq N_f \mid i < 0 \,\}$$

and

$$\mu(f) = |N_f| + |\Delta_f| \,.$$

This value $\mu(f)$ correctly evaluates the number of swaps required to relocate negative tokens in the star graph [15, 19]. One might think $\pi(f) + \mu(f)$ could be the right evaluation for $\mathsf{ts}(Q_{m,n}, f)$. However, when the vertex 0 is occupied by a negative token $i < 0$ and the vertex i is occupied by the positive token j which is the largest among the tokens on negative vertices, then the move of i to i (Type A) causes the right move of j to 0, which reduces the number of swaps required to move j to the goal. That is, actually π overestimates the number of swaps for j. We must discount the evaluation from $\pi(f) + \mu(f)$. For a pseudo configuration $f = (\vec{i};\, \vec{j}) = (\langle i_1, \dots, i_m\rangle;\, \langle j_1, \dots, j_k\rangle)$ and $c = \max(\vec{i})$, define

$$\delta(\vec{i};\, \vec{j}) = \begin{cases} 0 & \text{if } c < 0, \\ \delta(\vec{i};\, \langle j_2, \dots, j_k\rangle) & \text{if } j_1 > c \geq 0, \\ \delta(\vec{i}[j_1/c];\, \langle j_2, \dots, j_k\rangle) & \text{if } c > j_1 \geq 0, \\ \delta(\vec{i}[j_1/i_{-j_1}];\, \vec{j}[i_{-j_1}/j_1]) - 1 & \text{if } j_1 < 0 \text{ and } i_{-j_1} = c, \\ \delta(\vec{i}[j_1/i_{-j_1}];\, \vec{j}[i_{-j_1}/j_1]) & \text{otherwise.} \end{cases}$$

Note that if $j_1 < 0$, then $c \geq 0$. The discount function δ is well-defined, since the sum of the number of the misplaced tokens in $\vec{i}$ and the length of $\vec{j}$ decreases by one on the right-hand side in the above definition when $c \geq 0$.

Our evaluation function Ψ is given as

$$\Psi(f) = \pi(f) + \mu(f) + \delta(f)\,.$$

It is clear that $\Psi(f) = 0$ if f is the identity.

For a pseudo configuration $(\langle i_1, \dots, i_m\rangle;\, a)$, let us define

$$\gamma(\langle i_1, \dots, i_m\rangle;\, a) = \begin{cases} \gamma(\langle i_1, \dots, i_{-a-1}, a, i_{-a+1}, \dots, i_m\rangle;\, i_{-a}) & \text{if } a < 0, \\ (\langle i_1, \dots, i_m\rangle;\, a) & \text{if } a \geq 0. \end{cases}$$

The function γ simulates the **while** loop of Algorithm 2 in the sense that if the algorithm has $(\vec{i};\, a \cdot \vec{j})$ as the value of f at the beginning of the **while** loop, it will be $(\vec{i}';\, a' \cdot \vec{j})$ when exiting the loop for $(\vec{i}';\, a') = \gamma(\vec{i};\, a)$.

Lemma 15 *Let $\gamma(\vec{i};\, a) = (\vec{i}';\, b)$. Then*

$$\delta(\vec{i};\, a \cdot \vec{j}) = \begin{cases} \delta(\vec{i}';\, b \cdot \vec{j}) - 1 & \textit{if } b = \max(\vec{i}), \\ \delta(\vec{i}';\, b \cdot \vec{j}) & \textit{otherwise.} \end{cases}$$

Proof: We show the lemma by induction on the definition of γ. If $\gamma(\vec{i};\, a) = (\vec{i};\, a)$, a does not occur in $\vec{i}$, so $\delta(\vec{i};\, a) = \delta(\vec{i};\, a)$. Otherwise, suppose $a < 0$ and $\gamma(\vec{i};\, a) = \gamma(\vec{i}[a/i_{-a}];\, i_{-a})$. Remember $\max(\vec{i}) \geq 0$. If $i_{-a} = \max(\vec{i})$, then $\gamma(\vec{i}[a/i_{-a}];\, i_{-a}) = (\vec{i}[a/i_{-a}];\, i_{-a})$ and

$$\delta(\vec{i};\, a) = \delta(\vec{i}[a/i_{-a}];\, i_{-a}) - 1\,.$$

If $i_{-a} < \max(\vec{i})$, then $\delta(\vec{i};\, a) = \delta(\vec{i}[a/i_{-a}];\, i_{-a})$ and $\gamma(\vec{i};\, a) = \gamma(\vec{i}[a/i_{-a}];\, i_{-a})$. Since $a < 0 \leq \max(\vec{i})$ and $i_{-a} < \max(\vec{i})$, we have $\max(\vec{i}) = \max(\vec{i}[a/i_{-a}])$. By the induction hypothesis, we obtain the lemma. □

Lemma 16 *For any $\vec{i}, \vec{j}$, there are an integer $\alpha \leq 0$ and a sequence $\vec{i}'$ consisting of the m smallest elements from $\vec{i} \cdot \vec{j}$ such that for any $\vec{k}$*

$$\delta(\vec{i};\, \vec{j} \cdot \vec{k}) = \delta(\vec{i}';\, \vec{k}) + \alpha$$

provided that $(\vec{i};\, \vec{j} \cdot \vec{k})$ is a pseudo configuration.

Proof: This is immediate by the definition of δ. If the definition derives the equation $\delta(\vec{i};\vec{j}) = \delta(\vec{i}';\vec{j}')+\alpha = \alpha$ with $\max(\vec{i}') < 0$, then $\delta(\vec{i};\vec{j}\cdot\vec{k}) = \delta(\vec{i}';\vec{k})+\alpha$ holds anyway. □

Lemma 17 *If $\vec{i}\cdot\vec{j}_1$ contains m or more tokens smaller than $k \geq 0$, then*

$$\delta(\vec{i};\vec{j}_1\cdot k\cdot\vec{j}_2) = \delta(\vec{i};\vec{j}_1\cdot\vec{j}_2)$$

provided that $(\vec{i};\vec{j}_1\cdot k\cdot\vec{j}_2)$ is a pseudo configuration.

Proof: We show the lemma by induction on the definition of δ. Let $c = \max(\vec{i})$. If $c < 0$, then $\delta(\vec{i};\vec{j}_1\cdot k\cdot\vec{j}_2) = \delta(\vec{i};\vec{j}_1\cdot\vec{j}_2) = 0$. Suppose $c \geq 0$. If $\vec{j}_1$ is empty, $k > c \geq 0$ by the assumption. The equation holds immediately by definition. If $\vec{j}_1$ is not empty, the recursive definition of δ gives $\vec{i}'$ and $\vec{j}_1'$ such that

$$\delta(\vec{i};\vec{j}_1\cdot k\cdot\vec{j}_2) = \delta(\vec{i}';\vec{j}_1'\cdot k\cdot\vec{j}_2) + \alpha$$
$$\delta(\vec{i};\vec{j}_1\cdot\vec{j}_2) = \delta(\vec{i}';\vec{j}_1'\cdot\vec{j}_2) + \alpha$$

for some $\alpha \in \{0,-1\}$. To apply the induction hypothesis, it suffices to show that $\vec{i}'\cdot\vec{j}_1'$ contains at least m tokens smaller than $k \geq 0$. The only non-trivial case is that the number of tokens smaller than k in $\vec{i}'\cdot\vec{j}_1'$ is smaller than that in $\vec{i}\cdot\vec{j}_1$. In such a case, for the first element j_0 of $\vec{j}_1$, either $j_0 > c \geq 0$ and $j_0 < k$ (j_0 is absent in $\vec{i}'\cdot\vec{j}_1'$) or $c > j_0 \geq 0$ and $c < k$ (c is absent in $\vec{i}'\cdot\vec{j}_1'$) holds. In the former case, $k > c$ implies that all the m tokens in $\vec{i}' = \vec{i}$ are smaller than k. In the latter case, $k > c > j_0$ implies all the m tokens in $\vec{i}' = \vec{i}[j_0/c]$ are smaller than k. □

Corollary 4 *For any configuration $f = (\vec{i};\vec{j}_1\cdot k\cdot\vec{j}_2)$ with $k \geq 0$, if $\mathrm{Inv}(f,k) \leq k$ then*

$$\delta(\vec{i};\vec{j}_1\cdot k\cdot\vec{j}_2) = \delta(\vec{i};\vec{j}_1\cdot\vec{j}_2)\,.$$

In particular if $\vec{i}$ contains negative tokens only, $\delta(f) = 0$.

Proof: Recall that there exist just $k+m$ tokens smaller than k. The fact $\mathrm{Inv}(f,k) \leq k$ means that $\vec{j}_2$ contains at most k tokens smaller than k, so $\vec{i}\cdot\vec{j}_1$ must have at least m such tokens. Lemma 17 applies. □

B.1 Ψ Evaluates Our Algorithm

Lemma 18 *Suppose that our algorithm changes f to g at a point in the run. Then $\Psi(g) = \Psi(f) - 1$.*

Proof: We have two types of swaps.
Case A. When the algorithm moves the token $f(0) < 0$ to the vertex $f(0)$ (Type A). In this case we have $|N_g| = |N_f| - 1$. Let $a = f(0)$ and $b = f(a)$, which implies $g(0) = b$ and $g(a) = a$. Let $I = \{f(-1),\dots,f(-m)\}$ be the set of tokens on the negative vertices in f.

Case A.1. $0 \leq b = \max I$. Clearly $\delta(f) = \delta(g) - 1$ and $\mu(f) = \mu(g) + 1$. Since $\pi(f,i) = \pi(g,i)$ for all $i \neq b$, it is enough to show $\pi(f,b) = \pi(g,b) + 1$. By definition $\pi(f,b) = b + 1$. Recall that there are exactly $b + m$ tokens that are smaller than b. Since the m tokens on the negative vertices in g are all smaller than b, there are exactly b tokens smaller than b on non-negative vertices under g. That is, $\mathrm{Inv}(g,b) = b$ and thus $\pi(g,b) = b = \pi(f,b) - 1$.

Case A.2. $0 \leq b < \max I$. Clearly $\delta(f) = \delta(g)$ and $\mu(f) = \mu(g) + 1$. To see $\pi(f,i) = \pi(g,i)$ for all $i \in \{0,\dots,n\} - \{b\}$ is trivial, so it is enough to show $\pi(f,b) = \pi(g,b)$. By definition $\pi(f,b) = b+1$. Recall that there are exactly $b+m$ tokens that are smaller than b, of which at most $m-1$ tokens can be on negative vertices in g, since at least one negative vertex is occupied by a token bigger than b. Therefore, there are at least $b+1$ tokens smaller than b on non-negative vertices in g. That is, $\mathrm{Inv}(g,b) \geq b+1$ and thus $\pi(g,b) = b+1 = \pi(f,b)$.

Case A.3. $b < 0$. Clearly $\pi(f,i) = \pi(g,i)$ for all $i \geq 0$ and $\delta(f) = \delta(g)$ by definition. One can easily see $\Delta_f = \Delta_g$, for $[a]_f = [b]_f \notin \Delta_f$, $[a]_g \notin \Delta_g$ and $[b]_g \notin \Delta_g$. Hence $\mu(g) = \mu(f) - 1$.

Case B. When the algorithm moves a token k as a move of Type B.

Case B.1. $k \geq 0$ and $f^{-1}(k) < 0$. Let $a = f^{-1}(k)$ and $f(0) = b$, that is, $g(a) = b$ and $g(0) = k$. By the behavior of the algorithm, we have $f(i) \leq k$ for all $i \leq 0$. Since $b \geq 0$, we have $\mu(f) = \mu(g)$ and $\delta(f) = \delta(g)$. It is trivially true that $\pi(f,i) = \pi(g,i)$ for all $i \in \{0,\dots,n\} - \{k,b\}$. Thus it is enough to show that $\pi(g,k) + \pi(g,b) = \pi(f,k) + \pi(f,b) - 1$. By definition $\pi(f,k) = k+1$ and $\pi(g,b) = b+1$. Since all the m tokens on the negative vertices in g are smaller than k, the other k tokens smaller than k are found on some non-negative vertices. That is, $\mathrm{Inv}(g,k) = k$ and thus $\pi(g,k) = k = \pi(f,k) - 1$. On the other hand in f, at least one token, namely k, on a negative vertex is bigger than b. Therefore, at least $b+1$ tokens smaller than b are on some non-negative vertices in f. That is, $\mathrm{Inv}(f,b) \geq b+1$ and thus $\pi(f,b) = b+1$. Therefore, $\pi(g) = \pi(f) - 1$.

Case B.2. $k \geq 0$ and $f^{-1}(k) \geq 0$. Clearly $\mu(g) = \mu(f)$, $\mathrm{Inv}(g,k) = \mathrm{Inv}(f,k)-1$ and $\mathrm{Inv}(g,j) = \mathrm{Inv}(f,j)$ for all $j \in \{0,\dots,n\} - \{k\}$. By the behavior of the algorithm, $f(j) = j$ for all $j > k$ and thus $\mathrm{Inv}(f,k) \leq k$ and $\pi(g,k) = \pi(f,k)-1$. Hence $\pi(g) = \pi(f) - 1$. Corollary 4 implies $\delta(g) = \delta(f)$.

Case B.3. $k < 0$. The case where $f^{-1}(k) = 0$ can be discussed as in Case A.3. We assume $f^{-1}(k) < 0$, in which case we have $f(i) = i$ for all $i \geq 0$ by the behavior of the algorithm. Clearly $[k]_f \in \Delta_f$ and $\Delta_g = \Delta_f - \{[k]_f\}$, thus $|\Delta_g| = |\Delta_f| - 1$ and $\mu(g) = \mu(f) - 1$. On the other hand, $\pi(f,0) = 0$ and $\pi(g,0) = 1$, while $\pi(f,j) = \pi(g,j)$ for all $j > 0$. We have $\delta(f) = 0$ by Corollary 4 and $\delta(g) = -1$ by Lemma 15. □

Corollary 5 *For any configuration f, $\Psi(f) \geq 0$. Moreover, $\Psi(f) = 0$ if and only if f is the identity.*

B.2 Ψ Is the Right Evaluation Function

Now we are going to prove that any possible swap on the graph changes the value of Ψ by one. We have 6 cases depending on the signs of swapped tokens and the vertices where the swap takes place. Namely we discuss cases where the tokens are both non-negative (Lemma 21), where one is non-negative and the other is negative (Lemma 22) and where both are negative (Lemma 23). Each case has two subcases depending on whether one of the tokens is on a negative vertex or not. Lemmas 19 and 20 are useful to prove those lemmas.

Lemma 19 *Let $(\vec{i};\vec{j})$ and $(\vec{i}\,';\vec{j})$ be pseudo configurations such that $\vec{i}$ contains two distinct non-negative numbers $a, b \geq 0$ and $\vec{i}\,' = \vec{i}[a/b, b/a]$. Then*

$$|\delta(\vec{i};\,\vec{j}) - \delta(\vec{i}\,';\,\vec{j})| = 1\,.$$

Note that $\vec{j}$ cannot be empty, since $\vec{i}\cdot\vec{j}$ is a pseudo configuration.

Proof: It is enough to show that for any $a, b \geq 0$, $\vec{i}, \vec{j}$ and d,

$$|\delta(\langle a,b\rangle\cdot\vec{i};\,d\cdot\vec{j}) - \delta(\langle b,a\rangle\cdot\vec{i};\,d\cdot\vec{j})| = 1\,.$$

We show this claim by induction on the definition of δ. If $d \notin \{-1,-2\}$, the proof is trivial. For the symmetry, we discuss the case where $d = -1$ only. Without loss of generality we assume $a < b$. Let $c = \max(\vec{i})$.
Case 1. In the case where $b > c$, we have

$$\begin{aligned}
\delta(\langle a,b\rangle\cdot\vec{i};\,-1\cdot\vec{j}) &= \delta(\langle -1,b\rangle\cdot\vec{i};\,a\cdot\vec{j}) = \delta(\langle -1,a\rangle\cdot\vec{i};\,\vec{j})\,,\\
\delta(\langle b,a\rangle\cdot\vec{i};\,-1\cdot\vec{j}) &= \delta(\langle -1,a\rangle\cdot\vec{i};\,\vec{j}) - 1\,.
\end{aligned}$$

The claim holds.
Case 2. In the case where $b < c$,

$$\begin{aligned}
\delta(\langle a,b\rangle\cdot\vec{i};\,-1\cdot\vec{j}) &= \delta(\langle -1,b\rangle\cdot\vec{i};\,a\cdot\vec{j}) = \delta(\langle -1,b\rangle\cdot\vec{i}[a/c];\,\vec{j}),\\
\delta(\langle b,a\rangle\cdot\vec{i};\,-1\cdot\vec{j}) &= \delta(\langle -1,a\rangle\cdot\vec{i};\,b\cdot\vec{j}) = \delta(\langle -1,a\rangle\cdot\vec{i}[b/c];\,\vec{j})\,.
\end{aligned}$$

The claim follows the induction hypothesis. □

Let $f = (\vec{i};\,\vec{j})$ be a pseudo configuration where $\vec{i}$ contains a negative token a. The *a-resolution* of $\vec{i}$ is defined by

$$\vec{i}\,' = \vec{i}[a/f(a), f(a)/f^2(a), \ldots, f^{k-1}(a)/f^k(a), f^k(a)/a]$$

where k is the least natural number such that either $f^{k+1}(a) = a$ or $f^k(a) \geq 0$. That is, we relocate tokens $a, f(a), \ldots, f^{k-1}(a)$ on negative vertices to their respective goals and push $f^k(a)$ out to a, which is actually its goal if $f^{k+1}(a) = a$. We also call $g = (\vec{i}\,';\,\vec{j})$ the *a-resolution* of f. If $\gamma(\vec{i};\,a) = (\vec{j};\,b)$ and $a < 0$, it is easy to see that $\vec{j}$ is the a-resolutions of $\vec{i}[a/b]$.

Lemma 20 *If g is the a-resolution of f, then $\delta(f) = \delta(g)$.*

Proof: By induction on the definition of δ. □

Lemma 21 *Suppose that g is obtained from f by swapping non-negative tokens. Then* $|\Psi(g) - \Psi(f)| = 1$.

Proof: Suppose that non-negative tokens a and b are swapped. Obviously $\mu(g) = \mu(f)$. We have two cases depending on where those tokens are swapped.
Case 1. The swap takes place on two non-negative vertices. That is,

$$f = (\vec{i};\, \vec{j}_1 \cdot \langle a, b\rangle \cdot \vec{j}_2),$$
$$g = (\vec{i};\, \vec{j}_1 \cdot \langle b, a\rangle \cdot \vec{j}_2),$$

for some $a, b \geq 0$. Without loss of generality we assume $a < b$. We have $\pi(f, i) = \pi(g, i)$ for all $i \in \{0, \dots, n\} - \{b\}$, while $\mathrm{Inv}(g, b) = \mathrm{Inv}(f, b) + 1$. By Lemma 16, there exists $\vec{i}'$ consisting of the m smallest tokens from $\vec{i} \cdot \vec{j}_1$ such that

$$\delta(f) = \delta(\vec{i}';\, \langle a, b\rangle \cdot \vec{j}_2) + \alpha,$$
$$\delta(g) = \delta(\vec{i}';\, \langle b, a\rangle \cdot \vec{j}_2) + \alpha$$

for some $\alpha \leq 0$.
Case 1.1. $\mathrm{Inv}(f, b) < b$ and $\mathrm{Inv}(g, b) < b + 1$. In this case, we have $\pi(f, b) = \mathrm{Inv}(f, b)$, $\pi(g, b) = \mathrm{Inv}(g, b)$ and thus $\pi(g) = \pi(f) + 1$. Corollary 4 applies to both f and g and we obtain

$$\delta(g) = \delta(f) = \delta(\vec{i};\, \vec{j}_1 \cdot a \cdot \vec{j}_2)$$

and $\Psi(g) = \Psi(f) + 1$.
Case 1.2. $\mathrm{Inv}(f, b) = b$ and $\mathrm{Inv}(g, b) = b + 1$. In this case, we have $\pi(g) = \pi(f) + 1$. Since $\vec{j}_2$ contains b tokens smaller than b, on the other hand $\vec{i} \cdot \vec{j}_1$ contains exactly $m - 1$ tokens smaller than b. Let c be the unique element of $\vec{i}'$ which is bigger than b. Then

$$\delta(f) = \delta(\vec{i}'[a/c];\, b \cdot \vec{j}) + \alpha = \delta(\vec{i}'[a/c];\, \vec{j}) + \alpha,$$
$$\delta(g) = \delta(\vec{i}'[b/c];\, a \cdot \vec{j}) + \alpha = \delta(\vec{i}'[a/c];\, \vec{j}) + \alpha,$$

since b is the biggest in $\vec{i}'[b/c]$. Hence $\delta(g) = \delta(f)$ and $\Psi(g) = \Psi(f) + 1$.
Case 1.3. $\mathrm{Inv}(f, b) > b$ and $\mathrm{Inv}(g, b) > b + 1$. In this case, $\pi(f, b) = \pi(g, b) = b + 1$ and $\pi(g) = \pi(f)$. Since $\vec{j}_2$ contains at least $b + 1$ tokens smaller than b, $\vec{i} \cdot \vec{j}_1$ contains at most $m - 2$ tokens smaller than b. Hence $\vec{i}'$ contains at least 2 tokens bigger than b. Let c_1 and c_2 be the biggest and second biggest in $\vec{i}'$, respectively. Then

$$\delta(f) = \delta(\vec{i}'[a/c_1, b/c_2];\, b \cdot \vec{j}) + \alpha,$$
$$\delta(g) = \delta(\vec{i}'[b/c_1, a/c_2];\, a \cdot \vec{j}) + \alpha.$$

By Lemma 19, we obtain $|\delta(g) - \delta(f)| = 1$ and $|\Psi(g) - \Psi(f)| = 1$.

Case 2. The swap takes place between a negative vertex and 0. Without loss of generality we may assume that the negative position is -1 and $f(-1) < g(-1)$. That is,

$$f = (a \cdot \vec{i};\ b \cdot \vec{j}),$$
$$g = (b \cdot \vec{i};\ a \cdot \vec{j}),$$

where $0 \le a < b$. By definition $\pi(f, a) = a + 1$ and $\pi(g, b) = b + 1$. Since there are $a + m$ tokens smaller than a, of which at most $m - 1$ tokens can be in $b \cdot \vec{i}$, we have $\mathrm{Inv}(g, a) \ge a + 1$. That is, $\pi(g, a) = \pi(f, a)$. On the other hand, since there are $b+m$ tokens smaller than b, of which at most m tokens can be in $a \cdot \vec{i}$, we have $\mathrm{Inv}(f, b) \ge b$.
Case 2.1. $\mathrm{Inv}(f, b) = b$, which means $\pi(f, b) = b = \pi(g, b) - 1$. In this case, all the elements of $\vec{i}$ must be smaller than b. We have

$$\delta(f) = \delta(g) = \delta(a \cdot \vec{i};\ \vec{j})\,.$$

Therefore, $\Psi(g) = \Psi(f) + 1$.
Case 2.2. $\mathrm{Inv}(f, b) > b$, which means $\pi(f, b) = b + 1 = \pi(g, b)$. In this case, there must be $c > b$ in $\vec{i}$. We have

$$\delta(f) = \delta(a \cdot \vec{i};\ b \cdot \vec{j}) = \delta(a \cdot \vec{i}[b/c];\ \vec{j})\,,$$
$$\delta(g) = \delta(b \cdot \vec{i};\ a \cdot \vec{j}) = \delta(b \cdot \vec{i}[a/c];\ \vec{j})\,.$$

Lemma 19 implies $|\delta(g) - \delta(f)| = 1$. Therefore, $\Psi(g) = \Psi(f) + 1$. □

Lemma 22 *Suppose that g is obtained from f by swapping a non-negative token and a negative one. Then $|\Psi(g) - \Psi(f)| = 1$.*

Proof: **Case 1.** The swap takes place on non-negative vertices. Let

$$f = (\vec{i};\ \vec{j}_1 \cdot \langle a, b\rangle \cdot \vec{j}_2),$$
$$g = (\vec{i};\ \vec{j}_1 \cdot \langle b, a\rangle \cdot \vec{j}_2),$$

where $a < 0 \le b$. Obviously, $\mu(f) = \mu(g)$, $\mathrm{Inv}(g, b) = \mathrm{Inv}(f, b) + 1$ and $\mathrm{Inv}(g, k) = \mathrm{Inv}(f, k)$ for any other $k \ge 0$. By Lemma 16, there exists $\vec{i}'$ consisting of the m smallest tokens from $\vec{i} \cdot \vec{j}_1$ such that

$$\delta(f) = \delta(\vec{i}';\ \langle a, b\rangle \cdot \vec{j}_2) + \alpha,$$
$$\delta(g) = \delta(\vec{i}';\ \langle b, a\rangle \cdot \vec{j}_2) + \alpha$$

for some $\alpha \le 0$.
Case 1.1. $\mathrm{Inv}(f, b) < b$ and $\mathrm{Inv}(g, b) < b + 1$. In this case, we have $\pi(g) = \pi(f) + 1$. Corollary 4 applies to both f and g and we obtain $\delta(g) = \delta(f)$ and $\Psi(g) = \Psi(f) + 1$.
Case 1.2. $\mathrm{Inv}(f, b) = b$ and $\mathrm{Inv}(g, b) = b + 1$. In this case, we have $\pi(g) = \pi(f) + 1$. It is enough to show $\delta(f) = \delta(g)$. Since $\vec{j}_2$ contains b tokens smaller

than b, $\vec{i}\cdot\vec{j}_1$ and $\vec{i}\,'$ contain exactly $m-1$ tokens smaller than b. Let $c = \max(\vec{i}\,')$, which is the unique element of $\vec{i}\,'$ bigger than b. Let $(\vec{i}\,'';\, d) = \gamma(\vec{i}\,';\, a)$.

Suppose $c = d$. We have $\gamma(\vec{i}\,'[b/d];\, a) = (\vec{i}\,'';\, b)$, where b is the biggest in $\vec{i}\,'[b/d]$. Thus

$$\begin{aligned}\delta(f) &= \delta(\vec{i}\,'';\, b\cdot\vec{j}_2) - 1 + \alpha = \delta(\vec{i}\,'';\, \vec{j}_2) - 1 + \alpha\,,\\ \delta(g) &= \delta(\vec{i}\,'[b/d];\, a\cdot\vec{j}_2) + \alpha = \delta(\vec{i}\,'';\, \vec{j}_2) - 1 + \alpha\end{aligned}$$

by Lemma 15.

If $c > d$, we have $\gamma(\vec{i}\,'[b/c];\, a) = (\vec{i}\,''[b/c];\, d)$ and

$$\begin{aligned}\delta(f) &= \delta(\vec{i}\,'';\, \langle d,b\rangle\cdot\vec{j}_2) + \alpha = \delta(\vec{i}\,''[d/c];\, b\cdot\vec{j}_2) + \alpha = \delta(\vec{i}\,''[d/c];\, \vec{j}_2) + \alpha\,,\\ \delta(g) &= \delta(\vec{i}\,'[b/c];\, a\cdot\vec{j}_2) + \alpha = \delta(\vec{i}\,''[b/c];\, d\cdot\vec{j}_2) + \alpha = \delta(\vec{i}\,''[d/c];\, \vec{j}_2) + \alpha\end{aligned}$$

by Lemma 15.

Case 1.3. $\mathrm{Inv}(f,b) > b$ and $\mathrm{Inv}(g,b) > b+1$. In this case, we have $\pi(g) = \pi(f)$. It is enough to show $|\delta(g) - \delta(f)| = 1$. Since $\vec{j}_2$ contains at least $b+1$ tokens smaller than b, $\vec{i}\cdot\vec{j}_1$ contains at most $m-2$ tokens smaller than b. Hence $\vec{i}\,'$ contains at least 2 tokens bigger than b. Let c_1 and c_2 be the biggest and second biggest in $\vec{i}\,'$, respectively. Let $(\vec{i}\,''; d) = \gamma(\vec{i}\,';\, a)$.

If $d = c_1$, then by Lemma 15,

$$\begin{aligned}\delta(f) &= \delta(\vec{i}\,'';\, b\cdot\vec{j}_2) - 1 + \alpha\,,\\ \delta(g) &= \delta(\vec{i}\,'[b/c_1];\, a\cdot\vec{j}_2) + \alpha = \delta(\vec{i}\,'';\, b\cdot\vec{j}_2) + \alpha\,.\end{aligned}$$

If $d = c_2$, then by Lemma 15,

$$\begin{aligned}\delta(f) &= \delta(\vec{i}\,'';\, \langle c_2,b\rangle\cdot\vec{j}_2) + \alpha = \delta(\vec{i}\,''[c_2/c_1];\, b\cdot\vec{j}_2) + \alpha = \delta(\vec{i}\,''[b/c_1];\, \vec{j}_2) + \alpha\,,\\ \delta(g) &= \delta(\vec{i}\,'[b/c_1];\, a\cdot\vec{j}_2) + \alpha = \delta(\vec{i}\,''[b/c_1];\, \vec{j}_2) - 1 + \alpha\,.\end{aligned}$$

If $d < c_2$, then by Lemma 15,

$$\begin{aligned}\delta(f) &= \delta(\vec{i}\,'';\, \langle d,b\rangle\cdot\vec{j}_2) + \alpha = \delta(\vec{i}\,''[d/c_1][b/c_2];\, \vec{j}_2) + \alpha\,,\\ \delta(g) &= \delta(\vec{i}\,'[b/c_1];\, a\cdot\vec{j}_2) + \alpha = \delta(\vec{i}\,''[b/c_1];\, d\cdot\vec{j}_2) + \alpha = \delta(\vec{i}\,''[b/c_1][d/c_2];\, \vec{j}_2) + \alpha\,.\end{aligned}$$

Lemma 19 implies $|\delta(g) - \delta(f)| = |\delta(\vec{i}\,''[b/c_1][d/c_2]) - \delta(\vec{i}\,''[d/c_1][b/c_2])| = 1$.

Case 2. The swap takes place on 0 and a negative vertex. Without loss of generality we may assume

$$\begin{aligned}f &= (a\cdot\vec{i};\, b\cdot\vec{j}),\\ g &= (b\cdot\vec{i};\, a\cdot\vec{j}),\end{aligned}$$

where $a < 0 \le b$. For the case where $a = -1$, we have already proved that $\Psi(g) = \Psi(f) + 1$ in Lemma 18. Hereafter we assume that $a \neq -1$. Clearly $\pi(g,i) = \pi(f,i)$ for all $i \in \{0,\dots,n\} - \{b\}$ and $\pi(g,b) = b+1$. $\pi(f,b) = b$ if and only if every token in $\vec{i}$ is smaller than b. Let $(\vec{i}\,';\, d) = \gamma(b\cdot\vec{i};\, a)$.

Case 2.1. Suppose $\pi(f,b) = b$, in which case $\pi(g) = \pi(f) + 1$.

If $d = b$, there is $k \geq 1$ such that $g^i(a) < -1$ for $i \in \{0, \dots, k-1\}$ and $g^k(a) = -1$. Since $f(i) = g(i)$ for $i < -1$ and $f(-1) = a$, we have $[a]_f \in \Delta_f$. Hence $\Delta_f = \Delta_g \cup \{[a]_f\}$ and thus $\mu(g) = \mu(f) - 1$. We have

$$\delta(f) = \delta(a \cdot \vec{i};\, b \cdot \vec{j}) = \delta(a \cdot \vec{i};\, \vec{j})\,,$$
$$\delta(g) = \delta(b \cdot \vec{i};\, a \cdot \vec{j}) = \delta(\vec{i}';\, \vec{j}) - 1\,.$$

Since $\vec{i}'$ is the a-resolution of $a \cdot \vec{i}$, by Lemma 20, we have $\delta(\vec{i}';\, \vec{j}) = \delta(a \cdot \vec{i};\, \vec{j})$ and thus $\Psi(g) = \Psi(f) - 1$.

If $d < b$, $[a]_f \notin \Delta_f$. In this case, $\mu(g) = \mu(f)$ and $\vec{i}'$ has the form $b \cdot \vec{i}''$.

$$\delta(f) = \delta(a \cdot \vec{i};\, \vec{j})\,,$$
$$\delta(g) = \delta(b \cdot \vec{i}'';\, d \cdot \vec{j}) = \delta(d \cdot \vec{i}'';\, \vec{j})\,.$$

Since $d \cdot \vec{i}''$ is the a-resolution of $a \cdot \vec{i}$, by Lemma 20, we have $\delta(a \cdot \vec{i};\, \vec{j}) = \delta(d \cdot \vec{i}'';\, \vec{j})$ and thus $\Psi(g) = \Psi(f) + 1$.

Case 2.2. Suppose $\pi(f,b) = b + 1$, in which case $\pi(g) = \pi(f)$. Since there are at least $b+1$ tokens in $\vec{j}$ smaller than b, there are at most $m-1$ tokens smaller than b in $\vec{i}$. Let $c = \max(\vec{i})$, which is therefore bigger than b.

If $d = c$, $[a]_f \notin \Delta_f$. In this case, $\mu(g) = \mu(f)$ and

$$\delta(f) = \delta(a \cdot \vec{i}[b/c];\, \vec{j}),$$
$$\delta(g) = \delta(b \cdot \vec{i};\, a \cdot \vec{j}) = \delta(\vec{i}';\, \vec{j}) - 1\,.$$

Since $\vec{i}'$ is the a-resolution of $a \cdot \vec{i}[b/c]$, we have $\delta(g) = \delta(f) - 1$ and thus $\Psi(g) = \Psi(f) - 1$.

If $d = b$, $[a]_f \in \Delta_f$ and $[b]_g \notin \Delta_g$ by the same reason as in Case 2.1. In this case, $\mu(g) = \mu(f) - 1$ and

$$\delta(f) = \delta(a \cdot \vec{i};\, b \cdot \vec{j}) = \delta(a \cdot \vec{i}[b/c];\, \vec{j}),$$
$$\delta(g) = \delta(b \cdot \vec{i};\, a \cdot \vec{j}) = \delta(\vec{i}';\, b \cdot \vec{j}) = \delta(\vec{i}'[b/c];\, \vec{j})\,.$$

Since $\vec{i}'[b/c]$ is the a-resolution of $a \cdot \vec{i}[b/c]$, we have $\delta(f) = \delta(g)$ and thus $\Psi(g) = \Psi(f) - 1$.

If $d \notin \{b, c\}$, $[a]_f \notin \Delta_f$. In this case, $\mu(g) = \mu(f)$. There is $\vec{i}''$ such that $\vec{i}' = b \cdot \vec{i}''$. Let h be obtained from g by exchanging the tokens b and d. Then $|\delta(h) - \delta(g)| = 1$ and

$$\delta(f) = \delta(a \cdot \vec{i};\, b \cdot \vec{j}) = \delta(a \cdot \vec{i}[b/c];\, \vec{j}),$$
$$\delta(g) = \delta(b \cdot \vec{i};\, a \cdot \vec{j}) = \delta(b \cdot \vec{i}'';\, d \cdot \vec{j}) = \delta(b \cdot \vec{i}''[d/c];\, \vec{j})\,,$$
$$\delta(h) = \delta(d \cdot (\vec{i}[b/d]);\, a \cdot \vec{j}) = \delta(d \cdot \vec{i}'';\, b \cdot \vec{j}) = \delta(d \cdot \vec{i}''[b/c];\, \vec{j})\,.$$

Since $d \cdot \vec{i}''[b/c]$ is the a-resolution of $a \cdot \vec{i}[b/c]$, we have $\delta(f) = \delta(h)$ and thus $|\delta(g) - \delta(f)| = 1$. $|\Psi(g) - \Psi(f)| = 1$. □

Lemma 23 *Suppose that g is obtained from f by swapping negative tokens. Then* $|\Psi(g) - \Psi(f)| = 1$.

Proof: Clearly $\pi(f) = \pi(g)$.
Case 1. The swap takes place on non-negative vertices. Clearly $\mu(f) = \mu(g)$. It is enough to show $|\delta(g) - \delta(f)| = 1$. We may assume by Lemma 16

$$\begin{aligned}\delta(f) &= \delta(\vec{i};\, \langle a, b\rangle \cdot \vec{j})\,,\\ \delta(g) &= \delta(\vec{i};\, \langle b, a\rangle \cdot \vec{j})\,,\end{aligned}$$

where $a, b < 0$. Let $(\vec{i}_a; a') = \gamma(\vec{i};\, a)$ and $(\vec{i}_b; b') = \gamma(\vec{i};\, b)$. It is easy to see that there is $\vec{i}_{a,b}$ such that $\gamma(\vec{i}_a;\, b) = (\vec{i}_{a,b};\, b')$ and $\gamma(\vec{i}_b;\, a) = (\vec{i}_{a,b};\, a')$. Without loss of generality we assume $0 \le a' < b'$. Let c_1 and c_2 be the biggest and the second biggest in $\vec{i}$.
Case 1.1. $b' = c_1$. By Lemma 15,

$$\begin{aligned}\delta(f) &= \delta(\vec{i}_a;\, \langle a', b\rangle \cdot \vec{j}) = \delta(\vec{i}_a[a'/b'];\, b \cdot \vec{j}) = \delta(\vec{i}_{a,b};\, a' \cdot \vec{j}) - [a' = c_2]\,,\\ \delta(g) &= \delta(\vec{i}_b;\, a \cdot \vec{j}) - 1 = \delta(\vec{i}_{a,b};\, a' \cdot \vec{j}) - 1 - [a' = c_2]\,,\end{aligned}$$

where $[a' = c_2] = 1$ if $a' = c_2$ and $[a' = c_2] = 0$ otherwise. Therefore, $|\delta(f) - \delta(g)| = 1$.
Case 1.2. $b' = c_2$.

$$\begin{aligned}\delta(f) &= \delta(\vec{i}_a;\, \langle a', b\rangle \cdot \vec{j}) = \delta(\vec{i}_a[a'/c_1];\, b \cdot \vec{j}) = \delta(\vec{i}_{a,b}[a'/c_1];\, \vec{j}) - 1\,,\\ \delta(g) &= \delta(\vec{i}_b;\, \langle b', a\rangle \cdot \vec{j}) = \delta(\vec{i}_b[b'/c_1];\, a \cdot \vec{j}) = \delta(\vec{i}_{a,b}[b'/c_1];\, a' \cdot \vec{j})\\ &= \delta(\vec{i}_{a,b}[a'/c_1];\, \vec{j})\,.\end{aligned}$$

Case 1.3. $b' < c_2$.

$$\begin{aligned}\delta(f) &= \delta(\vec{i}_{a,b}[a'/c_1][b'/c_2];\, \vec{j})\,,\\ \delta(g) &= \delta(\vec{i}_b;\, \langle b', a\rangle \cdot \vec{j}) = \delta(\vec{i}_b[b'/c_1];\, a \cdot \vec{j}) = \delta(\vec{i}_{a,b}[b'/c_1];\, a' \cdot \vec{j})\\ &= \delta(\vec{i}_{a,b}[b'/c_1][a'/c_2];\, \vec{j})\,.\end{aligned}$$

By Lemma 19, $|\delta(g) - \delta(f)| = 1$.
Case 2. The swap takes place between a negative vertex and 0. Without loss of generality we may assume

$$\begin{aligned}f &= (a \cdot \vec{i};\, b \cdot \vec{j}),\\ g &= (b \cdot \vec{i};\, a \cdot \vec{j}),\end{aligned}$$

where $a, b < 0$. For the case where $a = -1$ or $b = -1$, we have already proved that $|\Psi(g) - \Psi(f)| = 1$ in Lemma 18. So we assume $a, b \neq -1$. Let $(\vec{i}_b;\, b') = \gamma(a \cdot \vec{i};\, b)$. There are negative tokens $b_0, \dots, b_k < 0$ in $a \cdot \vec{i}$ such that $b_i = f^i(b) < 0$ for all $i \le k$ and $f(b_k) = b' \ge 0$. Similarly for $(\vec{i}_a;\, a') = \gamma(b \cdot \vec{i};\, a)$, there are negative tokens $a_0, \dots, a_l < 0$ in $b \cdot \vec{i}$ such that $a_i = g^i(a) < 0$ for all $i \le l$ and

$g(a_l) = a' \geq 0$. Let θ_a and θ_b be replacements $[a_0/a_1, \ldots, a_{l-1}/a_l, a_l/a']$ and $[b_0/b_1, \ldots, b_{k-1}/b_k, b_k/b']$, respectively. Then $\vec{i}_a = (b \cdot \vec{i})\theta_a$ and $\vec{i}_b = (a \cdot \vec{i})\theta_b$.

Case 2.1. Suppose that the sequence $\langle a_0, \ldots, a_l \rangle$ contains -1. By $g(-1) = b$, we have

$$\langle a_0, \ldots, a_l, a' \rangle = \langle a_0, \ldots, a_{l-k-2}, -1, b_0, \ldots, b_k, b' \rangle ,$$

where $a_{l-k-1} = -1$, $a_{l-k} = b_0$ and $a' = b'$. Since $f^{l-k-1}(a) = g^{l-k-1}(a) = -1$ and $f(-1) = a$, we have $[a]_f = \{a_0, \ldots, a_{l-k-1}\} \in \Delta_f$. On the other hand, $g^{k+1}(b) = f^{k+1}(b) = b' \geq 0$ means that $[b]_f \notin \Delta_f$ and $[b]_g = [a]_g \notin \Delta_g$. Therefore, $\mu(f) = \mu(g) + 1$. Observing that

$$\begin{aligned}
\vec{i}_b &= (a \cdot \vec{i})\theta_b , \\
\vec{i}_a &= (b \cdot \vec{i})[a_0/a_1, \ldots, a_{l-k-1}/a_{l-k}]\theta_b \\
&= (a \cdot \vec{i})[b_0/a_0][a_0/a_1, \ldots, a_{l-k-1}/a_{l-k}]\theta_b \\
&= (a \cdot \vec{i})[a_{l-k-1}/a_0, a_0/a_1, \ldots, a_{l-k-2}/a_{l-k-1}]\theta_b \\
&= (a \cdot \vec{i})\theta_b[a_0/a_1, \ldots, a_{l-k-2}/a_{l-k-1}, a_{l-k-1}/a_0] \\
&= \vec{i}_b[a_0/a_1, \ldots, a_{l-k-2}/a_{l-k-1}, a_{l-k-1}/a_0] ,
\end{aligned}$$

we see that $\vec{i}_a$ is the a_0-resolution of $\vec{i}_b$. Therefore, by Lemmas 15 and 20,

$$\delta(f) = \delta(\vec{i}_b;\, b' \cdot \vec{j}) - d = \delta(\vec{i}_a;\, b' \cdot \vec{j}) - d = \delta(g) ,$$

where $d = 1$ if $b' = \max(\vec{i})$ and $d = 0$ otherwise. All in all, we have $|\Psi(f) - \Psi(g)| = 1$.

The case where $\langle b_0, \ldots, b_k \rangle$ contains -1 can be treated in the same way.

Case 2.2. Suppose that -1 occurs neither in $\langle a_0, \ldots, a_l \rangle$ nor $\langle b_0, \ldots, b_k \rangle$. It is easy to see that the two sequences $\langle b_0, \ldots, b_k, b' \rangle$ and $\langle a_0, \ldots, a_l, a' \rangle$ have no common elements. Hence $[a_0]_f \notin \Delta_f$ and $[b_0]_g \notin \Delta_g$. We obtain $\mu(f) = \mu(g)$. Without loss of generality, we may assume $a' < b'$. Let $h = f[a'/b', b'/a']$ be obtained from f by exchanging the positions of the tokens a' and b'. Since Lemma 19 ensures $|\delta(f) - \delta(h)| = 1$, it is enough to show $\delta(g) = \delta(h)$. By Lemma 15 and the fact $a' < b' \leq \max(\vec{i})$,

$$\delta(g) = \delta(\vec{i}_a;\, a' \cdot \vec{j}) = \delta((b_0 \cdot \vec{i})\theta_a;\, a' \cdot \vec{j})$$

The b_0-resolution of $\vec{i}_a$ is

$$(b_0 \cdot \vec{i})\theta_a[b_0/b_1, \ldots, b_{k-1}/b_k, b_k/b', b'/b_0] = (b' \cdot \vec{i})\theta_a\theta_b .$$

On the other hand,

$$\begin{aligned}
\delta(h) &= \delta(f[a'/b', b'/a']) = \delta((a_0 \cdot \vec{i})[a'/b', b'/a'][b_0/b_1, \ldots, b_{k-1}/b_k, b_k/a'];\, a' \cdot \vec{j}) \\
&= \delta((a_0 \cdot \vec{i})[b_0/b_1, \ldots, b_{k-1}/b_k, b_k/b', b'/a'];\, a' \cdot \vec{j}) \\
&= \delta((a_0 \cdot \vec{i})\theta_b[b'/a'];\, a' \cdot \vec{j}) .
\end{aligned}$$

The a_0-resolution of $(a_0 \cdot \vec{i})\theta_b[b'/a']$ is given as

$$\begin{aligned}&(a_0 \cdot \vec{i})\theta_b[b'/a'][a_0/a_1, \dots, a_{l-1}/a_l, a_l/b', b'/a_0]\\&= (a_0 \cdot \vec{i})\theta_b[a_l/a'][a_0/a_1, \dots, a_{l-1}/a_l, b'/a_0]\\&= (b' \cdot \vec{i})\theta_b[a_0/a_1, \dots, a_{l-1}/a_l, a_l/a']\\&= (b' \cdot \vec{i})\theta_b\theta_a = (b' \cdot \vec{i})\theta_a\theta_b\,,\end{aligned}$$

since θ_a and θ_b are independent. Therefore, $\delta(g) = \delta(h)$ by Lemma 20. □

Theorem 14 *Token Swapping on star-path graphs can be solved in polynomial time.*

Proof: By Lemmas 21, 22, 23 and 18, the number of swaps needed is exactly $\Psi(f)$. Obviously Ψ is computable in polynomial time. □

C Proof that the PPN-Separable 3SAT Is NP-hard

We show the NP-hardness of Sep-SAT by a reduction from the (usual) 3SAT [4]. For a given CNF F on X, we may without loss of generality assume that for each $x \in X$, the positive literal x and the negative one $\neg x$ occur exactly the same number of times in F. Otherwise, if x occurs k more times than $\neg x$ does, we add clauses $\{\neg x, y_i, \neg y_i\}$ to F for all $i \in \{1, \dots, k\}$ where y_i are new Boolean variables. Now, for a given CNF F on $X = \{x_1, \dots, x_m\}$ such that the positive and negative literals x_i and $\neg x_i$ occur exactly the same number of times for each Boolean variable $x_i \in X$, we construct $F' = F_1 \cup F_2 \cup F_3$ on X' such that

- F is satisfiable if and only if F' is satisfiable,
- each positive literal x_i occurs just once in each of F_1 and F_2,
- each negative literal $\neg x_i$ occurs just once in F_3.

Let n_i be the number of occurrences of the positive literal x_i in F (thus of the negative literal $\neg x_i$) for each $x_i \in X$.

1. Let $X' = \{\, x_{i,j}, \bar{x}_{i,j} \mid 1 \le i \le m,\, 1 \le j \le n_i \,\}$.
2. Let F_1 be obtained from F by replacing the j-th occurrence of the positive literal x_i with $x_{i,j}$, and the j-th occurrence of the negative literal $\neg x_i$ with $\bar{x}_{i,j}$ for $j \in \{1, \dots, n_i\}$.
3. Let $F_2 = \{\, \{x_{i,j}, \bar{x}_{i,j}\} \mid 1 \le i \le m \text{ and } 1 \le j \le n_i \,\}$.
4. Let $F_3 = \{\, \{\neg x_{i,j}, \neg \bar{x}_{i,j+1}\} \mid 1 \le i \le m \text{ and } 1 \le j < n_i \,\} \cup \{\, \{\neg x_{i,n_i}, \neg \bar{x}_{i,1}\} \mid 1 \le i \le m \,\}$.

Clearly F' is an instance of Sep-SAT. If a map $\phi : X \to \{0,1\}$ satisfies F, then $\phi' : X' \to \{0,1\}$ satisfies F' where $\phi'(x_{i,j}) = 1 - \phi'(\bar{x}_{i,j}) = \phi(x_i)$ for each i and j. Conversely, suppose that F' is satisfied by $\phi' : X' \to \{0,1\}$. The clauses of F_2 and F_3 ensure that $\phi'(x_{i,j}) = 1 - \phi'(\bar{x}_{i,j}) = \phi'(x_{i,1})$ for all $j \in \{1, \dots, n_i\}$. Then it is now clear that ϕ defined by $\phi(x_i) = \phi'(x_{i,1})$ satisfies F.