

Title	Reconfiguration Problems of Shifting Tokens on Graphs
Author(s)	Myint, Win Hlaing Hlaing
Citation	
Issue Date	2021-09
Type	Thesis or Dissertation
Text version	author
URL	<a href="http://hdl.handle.net/10119/17544">http://hdl.handle.net/10119/17544</a>
Rights	
Description	Supervisor:金子 峰雄, 先端科学技術研究科, 修士(情報科学)

Master's Thesis

Reconfiguration Problems of Shifting Tokens on Graphs

Win Hlaing Hlaing Myint

Supervisor Mineo Kaneko

Graduate School of Advanced Science and Technology  
Japan Advanced Institute of Science and Technology  
Master of Science (Information Science)

September, 2021

## Abstract

Reconfiguration problems are computational problems that explore connectivity between two feasible solutions or arrangements of a given problem. The general questions that are considered in reconfiguration problems are: can any arrangement be reconfigured to any other (connectivity); what is the worst-case number of steps required (diameter); and what is the complexity of computing the minimum number of steps required to get from one given configuration to another given configuration (distance).

The famous 15-puzzle inspired the token reconfiguration problem which is a reconfiguration problem on a graph with initial and target arrangements of tokens. Different versions of token reconfiguration problems on various graph structures have been studied widely in computer science. Token Swapping Problem was introduced by Yamanaka et al. . It was shown that 2-Colored Token Swapping can be solved in polynomial time and 3-Colored Token Swapping is NP-complete even for planar bipartite graphs of maximum degree 3. It also showed that  $c$ -Colored Token Swapping is  $O(n^{c+2})$ -time solvable for graphs of maximum degree at most 2, and  $c$ -Colored Token Swapping is fixed-parameter tractable for complete graphs if  $c$  is the parameter. In Sequentially Swapping Colored Tokens on graphs, inapproximability of the Sequential Token Swapping problem was demonstrated and the positive results for trees, complete graphs, and cycles were presented.

Token Shifting Problem was introduced by Sai et al. and it was shown that the Labeled Token Shifting Problem is solvable in polynomial time on a large class of graphs while solving the  $k$ -Colored Token Shifting Problem in the minimum number of moves is NP-hard even for  $k = 2$ .

In this thesis, we investigate a new variation of a token reconfiguration problem on graphs using the cyclic shift operation. A colored or labeled token is placed on each vertex of a given graph, and a “move” consists of choosing a cycle in the graph and shifting tokens by one position along its edges. Given a target arrangement of tokens on the graph, our goal is to find the shortest sequence of moves that will re-arrange the tokens as in the target arrangement. The novelty of our model is that tokens are allowed to shift along any cycle in the graph, as opposed to a given subset of its cycles. We focus our investigation on graph classes with high connectivity so as to reflect the potential applications in areas such as logistics and telecommunications.

We first present the efficient algorithms for solving the token shifting problem on special graph classes and then give the hardness result of the problem.

- We give linear-time algorithms for the shortest shift sequence for both the 2-Colored and the Labeled Token Shifting Problem for complete graphs.
- We also show that the shortest shift sequence for the Labeled Token Shifting Problem on standard barbell graphs, and then on generalized barbell graphs with more than one connecting edge can be constructed in  $O(n)$  time.
- We then describe the procedure of solving the 2-Colored Token Shifting Problem for block graphs in  $O(n^2)$  time.
- Finally, we prove that, in the 2-Colored Token Shifting Problem, the shortest sequence of moves is NP-hard to approximate within a factor of  $1/2 + \varepsilon$ , even for planar graphs with a maximum degree of 4 by reduction from the NP-complete problem of Hamiltonian cycle on grid graph.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Preliminaries</b>	<b>4</b>
<b>3</b>	<b>Token Shifting on Complete Graphs</b>	<b>6</b>
3.1	2-Colored Token Shifting on Complete Graphs . . . . .	6
3.2	Labeled Token Shifting on Complete Graphs . . . . .	7
<b>4</b>	<b>Token Shifting on Barbell Graphs and Their Generalizations</b>	<b>9</b>
4.1	Token Shifting on Barbell Graphs . . . . .	9
4.2	Token Shifting on Generalized Barbell Graphs with Two Bars	14
4.3	Token Shifting on Generalized Barbell Graphs with $k \geq 2$ Bars	17
<b>5</b>	<b>2-Colored Token Shifting on Block Graphs</b>	<b>19</b>
<b>6</b>	<b>Hardness of 2-Colored Token Shifting</b>	<b>26</b>
<b>7</b>	<b>Conclusion</b>	<b>28</b>
7.1	Summary . . . . .	28
7.2	Remaining Work . . . . .	28

# List of Figures

2.1	Token Shift $(v_1, v_3, v_2)$ on a simple graph . . . . .	5
3.1	2-colored token shifting on a complete graph: (a) an initial token arrangement $f_0$ , (b) a target token arrangement $f_t$ , and (c) an optimal shift cycle . . . . .	7
3.2	Labeled Token Shifting on a complete graph:(a) An initial token arrangement $f_0$ , (b) a target token arrangement $f_t$ , (c) the conflict graph $D(f_0, f_t)$ , and (d) the conflict graph $D(f_1, f_t)$ of a complete graph . . . . .	8
4.1	An example of Labeled Token Shifting on a barbell graph (a) initial arrangement $f_0$ (b) conflict graphs $D_A(f_0, f_t)$ and $D_B(f_0, f_t)$ . . . . .	11
4.2	An example of Labeled Token Shifting on a barbell graph showing token arrangements after each exchange across the bar . . . . .	13
4.3	An example of Labeled Token Shifting on a generalized barbell graph with 2 bars: (a) Initial arrangement $f_0$ (b)conflict graphs $D_A(f_0, f_t)$ and $D_B(f_0, f_t)$ and direction for token exchange between A and B shown in dotted arrows . . . . .	15
4.4	An example of Labeled Token Shifting on a generalized barbell graph with 2 bars . . . . .	16
4.5	Representation of token shifting on (a) a barbell graph (b) a generalized barbell graph with 2 bars (c) a generalized barbell graph with $k > 2$ bars . . . . .	18
5.1	2-colored Token Shifting on a block graph: (a) Initial arrangement $f_0$ , (b) target arrangement $f_t$ , and (c) tree representation $T(G)$ of the block graph $G$ with positive values over nodes that need black tokens, negative values over nodes that need white tokens, $\text{diff}(e)$ values over each edge $e$ . . . . .	23

5.2	An example of 2-colored Token Shifting on a block graph $G$ : $T(G)$ with dashed rectangle marking the feasible cliques for the shift (left) and the resulting token arrangement after the shifts (right) . . . . .	24
5.2	An example of 2-colored Token Shifting on a block graph $G$ : $T(G)$ with dashed rectangle marking the feasible cliques for the shift (left) and the resulting token arrangement after the shifts (right) (Cont.) . . . . .	25
6.1	2-colored Token Shifting on a grid graph: (a) Initial arrangement $f_0$ and (b) target arrangement $f_t$ . . . . .	27

# Chapter 1

## Introduction

Reconfiguration is concerned with relationships among solutions to a problem instance, where the reconfiguration of one solution to another is a sequence of steps such that each step produces an intermediate feasible solution. The solution space can be represented as a reconfiguration graph, where two vertices representing solutions are adjacent if one can be formed from the other in a single step. Reconfiguration arises in countless problems that involve movement and change, including problems in computational geometry such as morphing graph drawings and polygons, and problems relating to games and puzzles, such as the 15-puzzle, a topic of research since 1879 [1].

Reconfiguration problems also model real-life dynamic situations in which we seek to transform a solution into a more desirable one, maintaining feasibility during the process [2]. For instance, the optimization concept of token movement on graphs can naturally extend to problems such as motion planning, network design and vehicle routing.

The 15-puzzle can be considered as a token reconfiguration problem: the problem of re-configuring tokens around a given graph, where a token can be moved along an edge to an empty vertex. Previously studied token reconfiguration problems include the Token Swapping Problem, where pairs of tokens can be swapped along the edges of a graph. The Token Swapping Problem is proved to be NP-complete, and there are many special classes of graphs on which the Token Swapping Problem can be solved via exact polynomial-time algorithms, including cliques [3], paths [4], cycles [5], stars [6, 7, 8], brooms [9, 10], complete bipartite graphs [11], and complete split graphs [12] (see, e.g., [13] for comprehensive surveys).

Recently, the Token Shifting Problem was introduced by Sai et al. in [14], inspired by puzzles based on cyclic shift operations. The input of the problem is a graph with a distinguished set of cycles  $\mathcal{C}$ , and an initial and a final arrangement of colored tokens on the vertices of the graph. The basic oper-



ation is called “shift” along a cycle  $C \in \mathcal{C}$ , and it moves each token located on a vertex of  $C$  into the next vertex along  $C$ . The problem asks for a sequence of shift operations that transforms the initial configuration into the final configuration. We can further distinguish between the Labeled Token Shifting Problem, where all tokens are distinct, and the  $k$ -Colored Token Shifting Problem, where tokens come in  $k$  different colors, and same-colored tokens are indistinguishable.

It was shown in [14] that the Labeled Token Shifting Problem is solvable in polynomial time on a large class of graphs while solving the  $k$ -Colored Token Shifting Problem in the minimum number of moves is NP-hard, even for  $k = 2$ .

In this thesis, we study a variation of the Token Shifting Problem where the set of cycles  $\mathcal{C}$  consists of *all* cycles in the graph (as opposed to a subset of them). On one hand, our choice makes the problem description more natural and compact; on the other hand, proving hardness results is now more challenging. Indeed, previous NP-hardness proofs for variations of the Token Shifting Problem crucially relied on the fact that only shifts along certain cycles were allowed.

We remark that in [15], Amano et al. proved that a 2-Colored Token Shifting Problem called *Torus Puzzle* is NP-hard to solve in the minimum number of shifts. This puzzle consists of two arrays of horizontal and vertical cycles arranged in a grid, which yields a planar graph of maximum degree 4. However, in this puzzle the number of moves is measured differently: any number  $k > 0$  of consecutive shifts along the same cycle is counted as only one move, while in our model (as well as in [14]) we count them as  $k$  moves. Because of this, the NP-hardness reduction in [15] does not work in our model. In addition, the majority of cycles in the graph of the Torus Puzzle are forbidden from shifting (such as, for example, the 4-cycle determined by any cell in the grid). However, as already remarked, in our model we insist on allowing shifts along any cycle.

In the following chapters, we will present the topics below:

- **Chapter 2 Preliminaries:** Introducing the notions that will be used throughout the thesis.
- **Chapter 3 Token Shifting on Complete Graphs:** Describing optimal algorithms for solving 2-Colored and Labeled Token Shifting on complete graphs.
- **Chapter 4 Token Shifting on Barbell Graphs and Their Generalizations:** Describing optimal algorithms for solving Labeled Token

Shifting on barbell graphs, generalized barbell graphs with 2 bars, and generalized barbell graphs with  $k > 2$  bars.

- **Chapter 5 Token Shifting on Block Graphs:** Describing the procedure and the bounds for solving 2-Colored Token Shifting on block graphs.
- **Chapter 6 Hardness of 2-Colored Token Shifting:** Proving the NP-hardness of 2-Colored Token Shifting.

# Chapter 2

## Preliminaries

Let  $G = (V, E)$  be an undirected graph, where  $V$  is the vertex set and  $E$  is the edge set, and let  $\text{Col} = \{1, 2, \dots, c\}$  be the color set for tokens, where  $c$  is a constant. A *token arrangement* (or *configuration*) is a function  $f: V \rightarrow \text{Col}$ , where  $f(v)$  represents the color of the token located on the vertex  $v \in V$ .

The *token shift operation* can be defined as follows. Let  $C = (v_1, v_2, \dots, v_k)$  be a cycle of vertices of  $G = (V, E)$ , where  $\{v_i, v_{i+1}\} \in E$  for all  $1 \leq i < k$  and  $\{v_k, v_1\} \in E$ . Then, a token shift along  $C$  will transform any arrangement  $f$  into the arrangement  $f'$ , which coincides with  $f$  on all vertices except the ones in  $C$ . Specifically, for  $v_i \in \{v_1, v_2, \dots, v_{k-1}\}$ , we have  $f'(v_{i+1}) = f(v_i)$ , and  $f'(v_1) = f(v_k)$ . All cycles in  $G$  are eligible for token shift, and the length of the cycle can range from 2 to  $|V|$ . Note that we consider each edge of  $G$  as a cycle of length 2; in this case, the result of the shift operation will be equivalent to a token swap along that edge. In the following chapters, we will denote the token shift along cycle  $C = (v_1, v_2, \dots, v_k)$  as the shift  $(v_1, v_2, \dots, v_k)$ . A simple demonstration of token shift operation is shown in Figure 2.1.

The *Token Shifting Problem* takes as input a connected graph  $G = (V, E)$ , a color set  $\text{Col}$ , an initial arrangement  $f_0$ , and a final arrangement  $f_t$ . The problem asks to determine the shortest sequence of shift operations  $\text{OPT}$  that transforms  $f_0$  into  $f_t$ , assuming that such a sequence exists.

Note that, since swaps along edges are allowed, it is possible to transform  $f_0$  into  $f_t$  if and only if they have the same number of tokens of each color, which is checkable in the linear time given  $f_0$  and  $f_t$ . Thus, without loss of generality, we may assume that there is always a sequence of shift operations that transforms  $f_0$  into  $f_t$ , and our goal is to find the shortest one. Furthermore, it is easy to prove that  $|\text{OPT}| \leq |V|(|V| - 1)/2$  (this bound is obtained by using swap operations only; cf. [11, Theorem 1]). Since we have a polynomial upper bound of the number of shift operations, the Token

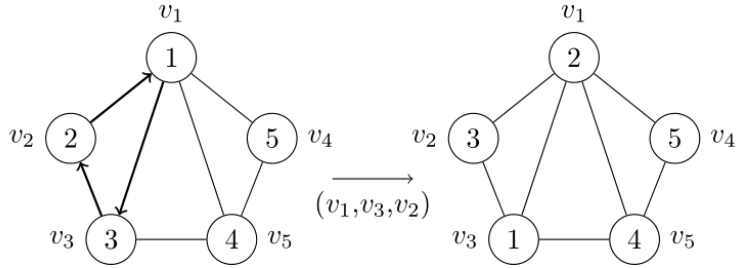


Figure 2.1: Token Shift  $(v_1, v_3, v_2)$  on a simple graph

Shifting Problem is in NP.

We distinguish between the *k-Colored Token Shifting Problem*, where the size of  $\text{Col}$  is a fixed constant  $k$ , and the *Labeled Token Shifting Problem*, where  $\text{Col} = V$ , and  $f_0$  and  $f_t$  are permutations of  $V$  (that is, all tokens have distinct labels). In this thesis, we will mostly focus on the 2-Colored Token Shifting Problem (i.e., where  $\text{Col} = \{c_1, c_2\}$ ) and the Labeled Token Shifting Problem.

We will also introduce the notion of conflict graph [11] constructed from two token arrangements on a graph which will be used in later chapters. We define the *conflict graph*  $D(f_a, f_b) = (V', E')$  of graph  $G = (V, E)$  for two arrangements  $f_a$  and  $f_b$  as follows [11]:

$$V' = \{v \in V \mid f_a(v) \neq f_b(v)\} \text{ and}$$

$$E' = \{e = (v_i, v_j) \mid f_a(v_i) = f_b(v_j) \text{ and } v_i, v_j \in V'\}.$$

$D(f_a, f_b)$  is a digraph that includes vertices that hold different tokens in the token arrangements  $f_a$  and  $f_b$  and there is an arc from  $v_i$  to  $v_j$  if the token on  $v_i$  needs to be moved to  $v_j$ .

# Chapter 3

## Token Shifting on Complete Graphs

### 3.1 2-Colored Token Shifting on Complete Graphs

In this section, we show that for the 2-Colored Token Shifting Problem on complete graphs, an optimal shift sequence can be constructed in linear time.

**Theorem 1.** *The 2-Colored Token Shifting Problem on a complete graph  $G = (V, E)$  can be solved in linear time by a single shift operation.*

*Proof.* Let  $\text{Col} = \{c_1, c_2\}$  be the color set and let  $f_0$  and  $f_t$  be the initial and target token arrangements, respectively. We can construct two sets  $V_1$  and  $V_2$  of vertices as follows:

$$V_1 = \{v \in V \mid f_0(v) = c_1 \text{ and } f_t(v) = c_2\} \text{ and}$$

$$V_2 = \{v \in V \mid f_0(v) = c_2 \text{ and } f_t(v) = c_1\}.$$

Given that  $f_0$  is re-configurable to  $f_t$ ,  $|V_1| = |V_2| = m$  for a complete graph with  $2m$  misplaced tokens. Thus, we can construct a cycle of length  $2m$  that visits each vertex in  $V_1$  and  $V_2$  alternately. For  $V_1 = \{x_1, x_2, \dots, x_m\}$  and  $V_2 = \{y_1, y_2, \dots, y_m\}$ , the shift  $(x_1, y_1, x_2, y_2, \dots, x_m, y_m)$  transforms  $f_0$  into  $f_t$ .  $\square$

For example, in Figure 3.1,  $V_1 = \{v_5, v_8\}$  and  $V_2 = \{v_2, v_4\}$ . From  $V_1$  and  $V_2$  the shift cycle  $(v_2, v_5, v_4, v_8)$  can be constructed, which transforms  $f_0$  into  $f_t$ .

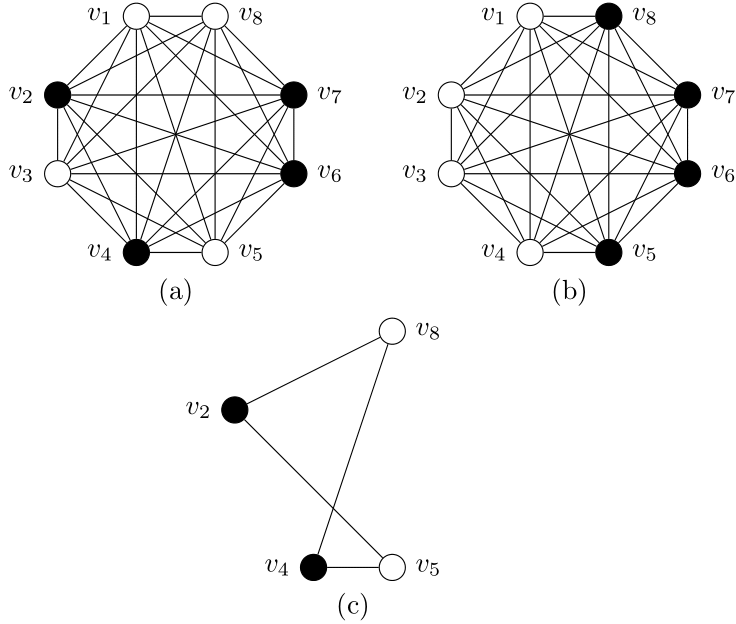


Figure 3.1: 2-colored token shifting on a complete graph: (a) an initial token arrangement  $f_0$ , (b) a target token arrangement  $f_t$ , and (c) an optimal shift cycle

### 3.2 Labeled Token Shifting on Complete Graphs

In this section, we show that the Labeled Token Shifting Problem on a complete graph can be solved by at most two shift operations.

**Theorem 2.** *The Labeled Token Shifting Problem on a complete graph  $G = (V, E)$  can be solved with a minimum shift sequence  $|\text{OPT}| \leq 2$  in linear time.*

*Proof.* For a complete graph  $G = (V, E)$  with initial token arrangement  $f_0$  and target token arrangement  $f_t$ , construct a conflict graph  $D(f_0, f_t)$ . Each vertex in  $D(f_0, f_t)$  must belong to a directed cycle for  $f_0$  to be configurable to  $f_t$ . A simple example is given in Figure 3.2. One way to transform  $f_0$  to  $f_t$  would be to perform a token shift along each directed cycle in  $D(f_0, f_t)$ ; if there are only 1 or 2 cycles, this strategy is optimal. However, it is not optimal when the number of cycles is more than 2.

The disjoint cycles in  $D(f_0, f_t)$  are analogous to the notion of permutation cycles. In Figure 3.2(c) we have the three disjoint cycles  $(v_1, v_4)$ ,  $(v_2, v_6, v_3, v_7)$ , and  $(v_5, v_8)$ , which collectively correspond to the permutation  $(14)(2637)(58)$ .

Suppose there are  $m$  disjoint cyclic permutations involving  $n$  elements; the product of these  $m$  disjoint cycles and a length- $m$  cycle consisting of one

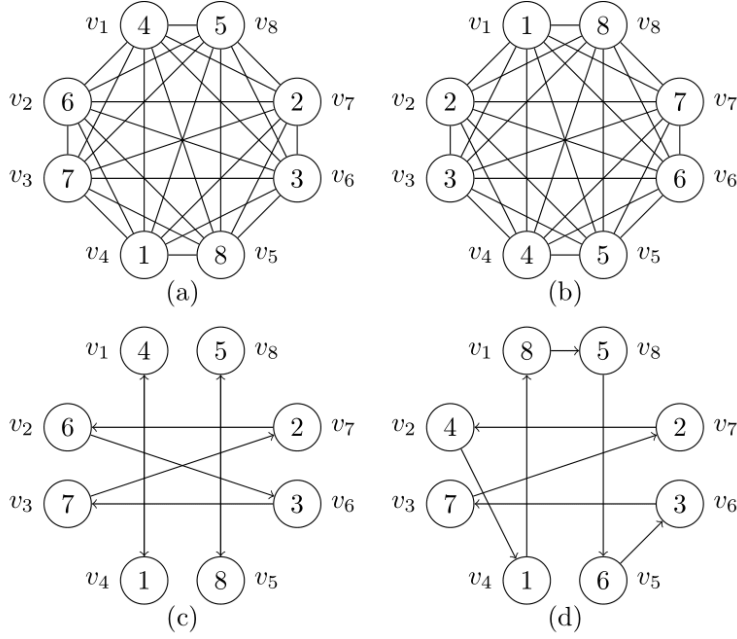


Figure 3.2: Labeled Token Shifting on a complete graph:(a) An initial token arrangement  $f_0$ , (b) a target token arrangement  $f_t$ , (c) the conflict graph  $D(f_0, f_t)$ , and (d) the conflict graph  $D(f_1, f_t)$  of a complete graph

element from each disjoint cycle is a single length- $n$  cycle which includes all  $n$  elements. For example,  $(14)(2637)(58)(521) = (18563724)$ .

Accordingly, we construct a first shift cycle including one vertex from each cycle in  $D(f_0, f_t)$ . That will result in an arrangement  $f_1$  whose conflict graph  $D(f_1, f_t)$  consists of a single directed cycle (see Figure 3.2(d)). We can then perform a second shift along this cycle to obtain the target token arrangement  $f_t$ .  $\square$

**Corollary 2.1.** *For the  $k$ -Colored Token Shifting Problem on a complete graph  $G = (V, E)$ , we have  $|\text{OPT}| \leq 2$ .*

*Proof.* Let  $f_0$  and  $f_t$  be the initial and final arrangements, respectively. Let  $\text{Col}' = V$ , and let us define  $f'_0$  as an arbitrary bijection  $f'_0: V \rightarrow \text{Col}'$ . We then define  $f'_t: V \rightarrow \text{Col}'$  as a bijection that, for all  $v_i, v_j \in V$ , satisfies  $f'_0(v_i) = f'_t(v_j) \implies f_0(v_i) = f_t(v_j)$ . Essentially, we assign unique labels to tokens in a way that is consistent with their colors. Thus, we obtain an instance of the Labeled Token Shifting Problem, which we can solve by Theorem 2. The same sequence of moves also solves the original instance, by construction.  $\square$

# Chapter 4

## Token Shifting on Barbell Graphs and Their Generalizations

In this chapter, we consider the Labeled Token Shifting Problem on barbell graphs and their generalization. A *barbell graph* is a simple graph obtained by connecting two complete graphs by an edge, which is called its *bar*. Our goal is to find the minimum shift sequence between initial and final token arrangements  $f_0$  and  $f_t$  on a barbell graph. Then we extend our result to generalized barbell graphs that have two or more bars.

### 4.1 Token Shifting on Barbell Graphs

We first show that we can find the minimum shift sequence on a barbell graph in linear time. Let  $G$  be a barbell graph composed of two cliques  $A$  and  $B$ , each of size  $n$ , connected by a single edge which is called the bar.

The two cliques  $A$  and  $B$  of size  $n$  containing vertices  $v_1$  to  $v_n$  and from  $v_{n+1}$  to  $v_{2n}$ , respectively. The two vertices joined by the bar will be referred as *gate* vertices. Furthermore, we subdivide the tokens into two types, based on their matching vertices in the target arrangement: *local* tokens and *foreign* tokens, as follows. Tokens on vertices in a clique whose target vertices are in the other clique are referred to as *foreign* tokens. Let  $\text{foreign}(A)$  be the set of foreign tokens in  $A$  in  $f_0$  and  $\text{foreign}(B)$  be the set of foreign tokens in  $B$  in  $f_0$  as follows:

$$\text{foreign}(A) = \{v_i \in V \mid f_0(v_i) = f_t(v_j) \text{ where } v_i \in A \text{ and } v_j \in B\},$$

$$\text{foreign}(B) = \{v_i \in V \mid f_0(v_i) = f_t(v_j) \text{ where } v_i \in B \text{ and } v_j \in A\}.$$



Let  $F = |\text{foreign}(A)| = |\text{foreign}(B)|$ . Next, we will prove that  $3F - 2 \leq |\text{OPT}| \leq 3F + 4$ . Note that  $|\text{foreign}(A)| = |\text{foreign}(B)| = F$  must hold in order for  $f_0$  to be re-configurable to  $f_t$ . Let  $S_F$  be an optimal shift sequence for moving all  $2F$  foreign tokens to their matching vertices (possibly leaving some non-foreign tokens on incorrect vertices).

**Lemma 3.** *In the Labeled Token Shifting Problem on a barbell graph, we have  $3F - 2 \leq |S_F| \leq 3F + 2$ .*

*Proof.* To transform  $f_0$  to  $f_t$ , it is required for every foreign token on  $A$  and  $B$  to cross the bar at least once. Note that we can move two foreign tokens by performing a token exchange across the bar. In the worst case, a foreign token needs to be moved three times: from the current vertex to the nearest gate vertex, then across the bar to the gate vertex of the target clique, and then to the target vertex. Firstly, a foreign token on each clique must be moved to the gate vertex of that clique, which takes 2 shifts in total. Then, the actual exchange of tokens on gate vertices in a shift cycle  $(v_n, v_{n+1})$  of length 2 occurs. Next, in each clique, the token on the gate vertex, say  $v_n$ , is moved to its target vertex  $v_i$ , while a new foreign token is moved from  $v_j$  to the gate. This is done with the single cycle  $(v_n, v_i, v_j)$ . After the  $F$ th exchange, we need one more shift in each clique to move the token from the gate vertex to its target vertex. Therefore, in the worst case, we do  $F$  exchanging shifts and  $2F + 2$  local shifts, which is  $3F + 2$  shifts in total. However, we also need to consider the following special cases.

**Condition 1.** *A gate vertex already holds a foreign token in the initial arrangement  $f_0$ .*

If a gate vertex already holds a foreign token in the initial arrangement, then the initial shift for moving a foreign token to that gate vertex is not necessary as in  $A$  of Figure 4.1(b). Hence, in the cases where  $A$  or  $B$  (or both) satisfy Condition 1, we need one (or two) fewer shifts than  $2F + 2$ .

**Condition 2.** *The target token of a gate vertex (i.e., the token that is on a gate vertex in  $f_t$ ) is in the opposite clique in  $f_0$ .*

If this condition is satisfied, we can move that gate's final token across the bar in the  $F$ th exchange. In this way, it is already in place when it enters the clique, and we can spare the final shift in that clique. In Figure 4.1(b), both  $A$  and  $B$  satisfied Condition 2. Thus, in the extreme case where both gate vertices satisfy Conditions 1 and 2, and only  $3F - 2$  shifts are necessary.  $\square$

As for the local tokens, their target vertices are within the same clique. Hence, by Theorem 2, at most 2 shifts are necessary to solve the problem in each clique.

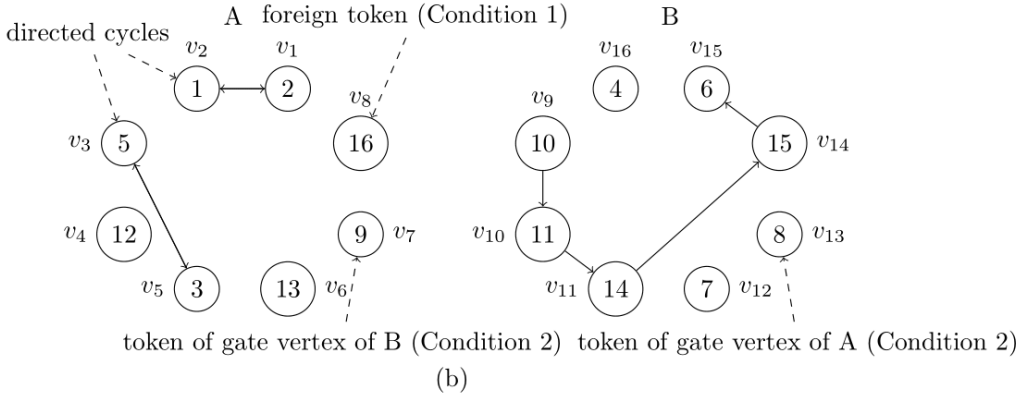
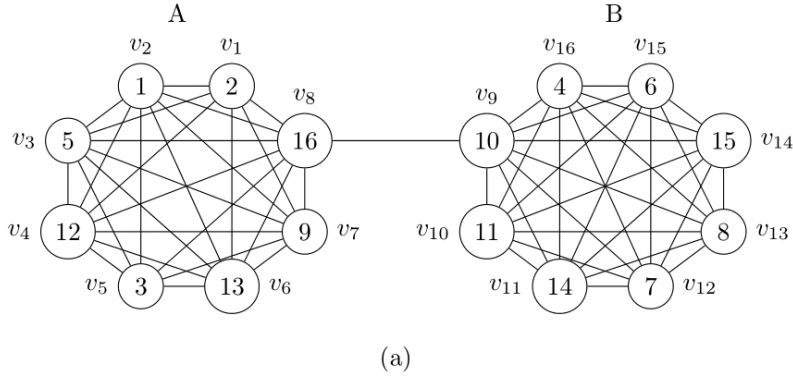


Figure 4.1: An example of Labeled Token Shifting on a barbell graph (a) initial arrangement  $f_0$  (b) conflict graphs  $D_A(f_0, f_t)$  and  $D_B(f_0, f_t)$

**Theorem 4.** *The Labeled Token Shifting Problem on a barbell graph  $G = (V, E)$  can be solved with an optimal shift sequence in linear time, satisfying  $3F - 2 \leq |\text{OPT}| \leq 3F + 4$ .*

*Proof.* We can classify each vertex into one of three types by constructing conflict graphs for  $A$  and  $B$ . A vertex either

1. already holds its target token,
2. belongs to a directed cycle such as  $(v_i, v_j, \dots, v_k)$  where  $v_k$  holds token  $i$ ,  $v_i$  holds token  $j$ , or
3. belongs to a chain of vertices that cannot form a cycle such as  $v_i, v_j, \dots, v_k$  where  $v_i$  holds token  $j$ ,  $v_k$  holds a foreign token  $j$ , and token  $i$  belongs to another clique.

Type-1 vertices need no consideration. As for type-3 vertices, they can be solved while exchanging foreign tokens. Since token  $i$  must reach gate  $v_n$  after an exchange at some point, we can then perform the cyclic shift  $(v_i, v_j, \dots, v_k, v_n)$ . This will move the token  $i$  to  $v_i$ , token  $j$  to  $v_j$ , token  $k$  to  $v_k$  and lastly the foreign token on  $v_k$  to  $v_n$ . This not only matches the vertices  $v_i, v_j, \dots, v_k$  with their tokens but also moves a foreign token to  $v_n$  for the next exchange. We now consider how to deal with the type-2 vertices. As they are isolated from the type-3 vertices, they cannot be solved while exchanging foreign tokens. Hence, to avoid additional shifts, we connect the directed cycles to a chain of type-3 tokens. We can do that by performing a shift that includes a type-2 vertex from each directed cycle and a type-3 vertex while moving a foreign token to the gate vertex. In the example described in Figure 4.1 with  $F=4$  with Condition 1 on  $A$  and Condition 2 on both cliques, we can compute that  $|OPT| = |S_F| = (3F + 2) - 3 = 11$ . In Figure 4.2, we can see how the problem is solved in 11 shifts. Since  $v_8$  already holds foreign token 16 in  $f_0$ , we only need 2 shifts before the first exchange. The directed cycles  $(v_1, v_2)$  and  $(v_3, v_5)$  are connected to type-3 token  $v_4$  while moving the foreign token 12 to  $v_8$  by the shift  $(v_1, v_3, v_4, v_8)$ . We get the target arrangement in  $f_{11}$  as the 4th exchange was for tokens 8 and 9 that belong to the gate vertices. In this way, we can handle the local tokens while exchanging foreign tokens and  $|OPT| = |S_F|$ . However, this is true only when  $|S_F| \geq 5$ .

Let us now discuss the exceptional case where the minimum shift sequence required for exchanging foreign tokens satisfies  $|S_F| < 5$ . In this case, fewer than two shifts are performed on one of the cliques during the exchange. When  $F = 0$ , the problem becomes two independent token shifting problems on two complete graphs, which may need 4 shifts in total. Thus,  $3F + 2$  is no longer an upper bound.

We can conclude that the shortest shift sequence for token shifting on a barbell graph is  $|OPT| = |S_F| = 3F + 2$  in the general case without Conditions 1, 2, and excluding the exceptional case discussed above. We can now compute optimal bounds on the minimum shift sequence from the extreme cases as follows. For a case with  $F = 0$ , we need at most 4 shifts for solving the problem on two complete graphs independently, and so  $|OPT| = 3F + 4$  holds. For a case with Conditions 1 and 2 on both sides, we have the minimum sequence of  $|OPT| = 3F - 2$ . We can easily determine whether those conditions hold in linear time.  $\square$

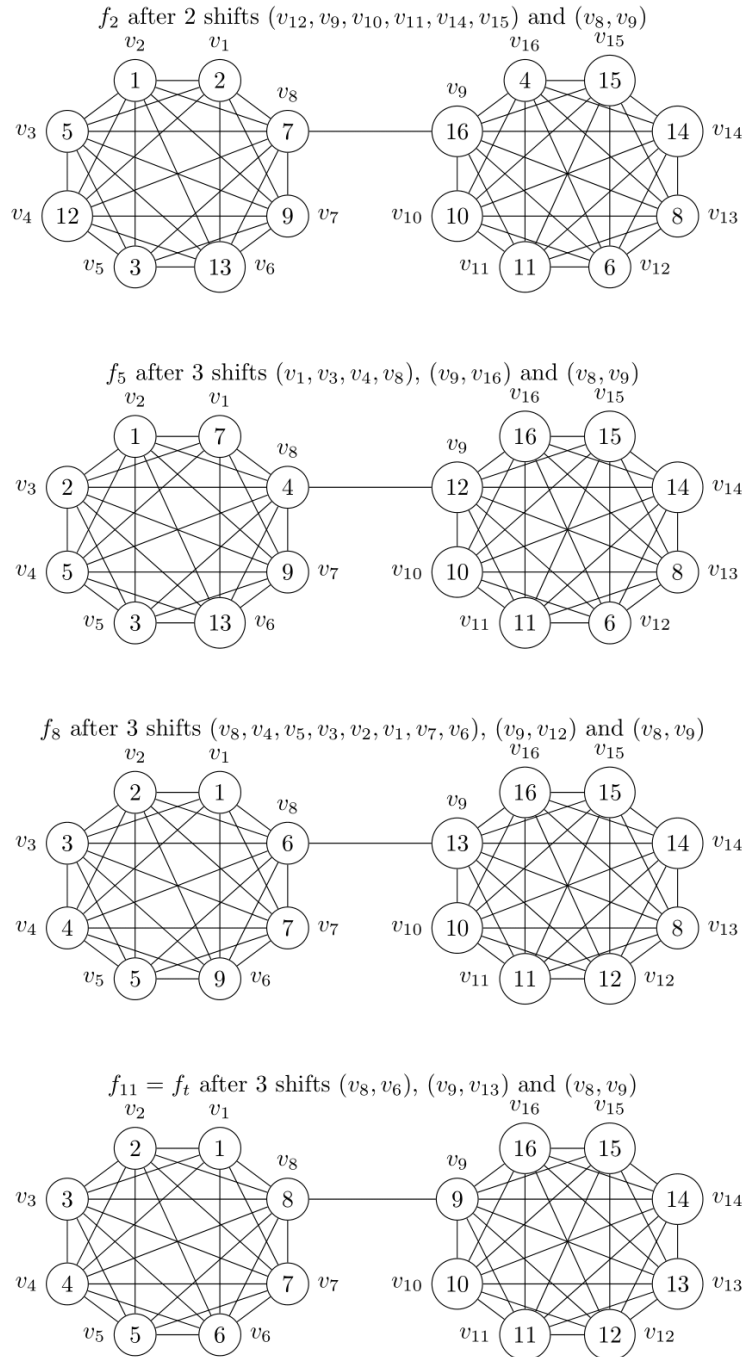


Figure 4.2: An example of Labeled Token Shifting on a barbell graph showing token arrangements after each exchange across the bar

## 4.2 Token Shifting on Generalized Barbell Graphs with Two Bars

In this section, we extend our previous result to generalized barbell graphs. That is, we join two cliques by two bars instead of one, and this allows us to more effectively exploit the cyclic shift operation.

Let  $G$  be a generalized barbell graph with  $2n$  vertices, with cliques  $A$  and  $B$  consisting of vertices from  $v_1$  to  $v_n$  and  $v_{n+1}$  to  $v_{2n}$ , respectively. Two bars  $e_1$  and  $e_2$  connect  $A$  and  $B$  such that  $e_1$  is incident to  $v_n$  and  $v_{n+1}$ , and  $e_2$  is incident to  $v_{n-1}$  and  $v_{n+2}$ . Let  $F = |\text{foreign}(A)| = |\text{foreign}(B)|$ , defined as in the previous section.

**Theorem 5.** *The Labeled Token Shifting Problem on a generalized barbell graph  $G = (V, E)$  with 2 bars can be solved with an optimal shift sequence in linear time, satisfying  $F \leq |\text{OPT}| \leq F + 4$ .*

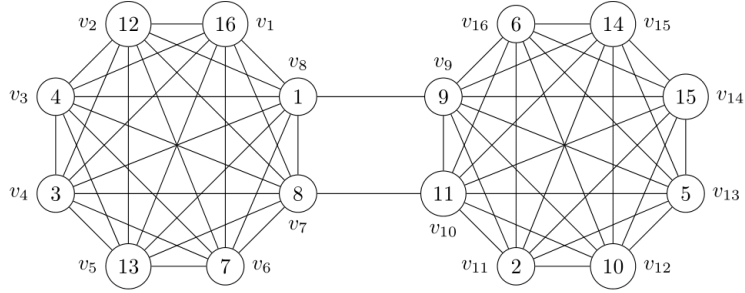
*Proof.* As discussed before, an exchange needs two steps: moving foreign tokens on each clique to the gate vertices and the actual exchange of tokens on gate vertices. In a barbell graph with 2 bars, we can combine the two steps into one by exchanging foreign tokens and bringing the foreign tokens to the gate vertices for the next exchange in a single shift. Since each clique now has two gate vertices, one vertex acts as the entry gate vertex where the incoming tokens pass through and another acts like the exit gate vertex through which the foreign tokens leave. Between the cliques  $A$  and  $B$ , the two bars  $e_1$  and  $e_2$  act like two lanes going in opposite directions.

Let  $v_n$  and  $v_{n-1}$  be the gate vertices of  $A$  and  $v_{n+1}$  and  $v_{n+2}$  be the gate vertices of  $B$  such that  $v_n$  and  $v_{n+1}$  are connected by  $e_1$  and  $v_{n-1}$  and  $v_{n+2}$  are connected by  $e_2$ . In a single shift, we can move a foreign token  $b_3$  inside  $A$  to  $v_n$  (exit of  $A$ ),  $b_2$  on  $v_n$  to  $v_{n+1}$  (entry of  $B$ ), and  $b_1$  on  $v_{n+1}$  to  $v_{n+2}$  inside  $B$ . Also, move a foreign token  $a_3$  inside  $B$  to  $v_{n+2}$  (exit of  $B$ ),  $a_2$  on  $v_{n+2}$  to  $v_{n-1}$  (entry of  $A$ ), and  $a_1$  on  $v_{n-1}$  to  $v_{n-2}$  inside  $A$  (see Figure 4.5(b)).

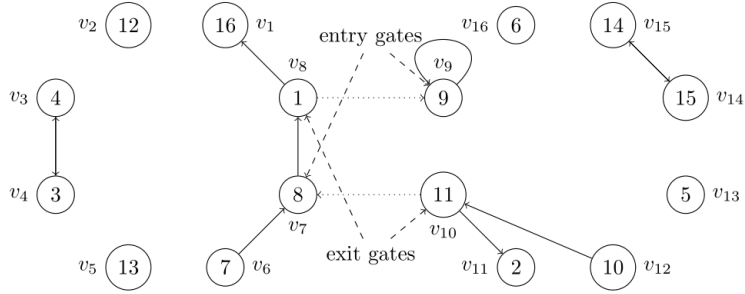
Therefore,  $|S_F|$  is reduced to  $F + 4$  ( $F$  exchanging shifts, 2 pre-exchange shifts, and 2 post-exchange shifts). The generalized barbell graphs with 2 bars also have two exceptional conditions, corresponding to Conditions 1 and 2 in Lemma 3.

In this case of Condition 1, we need one less shift than  $F + 4$ . If the gate vertices  $v_a$  of  $A$  and  $v_b$  of  $B$  are not adjacent and both vertices hold foreign tokens, we can start exchanging tokens immediately and need 2 fewer shifts.

In the case of Condition 2, the target token of a gate vertex lies in the opposite clique. This token can be exchanged last to save 1 shift. If both  $A$  and  $B$  satisfy Conditions 1 and 2, then  $|S_F| = F$ .



(a)



(b)

Figure 4.3: An example of Labeled Token Shifting on a generalized barbell graph with 2 bars: (a) Initial arrangement  $f_0$  (b) conflict graphs  $D_A(f_0, f_t)$  and  $D_B(f_0, f_t)$  and direction for token exchange between A and B shown in dotted arrows

We can deal with the local tokens in a similar way as in Section 4.1, so that no additional shift is necessary for moving local tokens when  $|S_F| \geq 2$ . Figure 4.4 shows an example with  $F=3$  and without Condition 1 or 2 solved in  $|\text{OPT}|=|S_F|=F+4=7$  shifts.

In the exceptional case where the minimum shifts required for exchanging foreign tokens is  $|S_F| < 2$ , local tokens cannot be handled by  $S_F$ .

We can now work out exact bounds on the minimum shift sequence from the extreme cases as follows. For a case with  $F = 0$ , we need at most 4 shifts for solving the two cliques separately, and  $|\text{OPT}| = F + 4$  holds. For a case with Conditions 1 and 2 on both sides, we have the minimum shift sequence of  $|\text{OPT}| = F$ .  $\square$

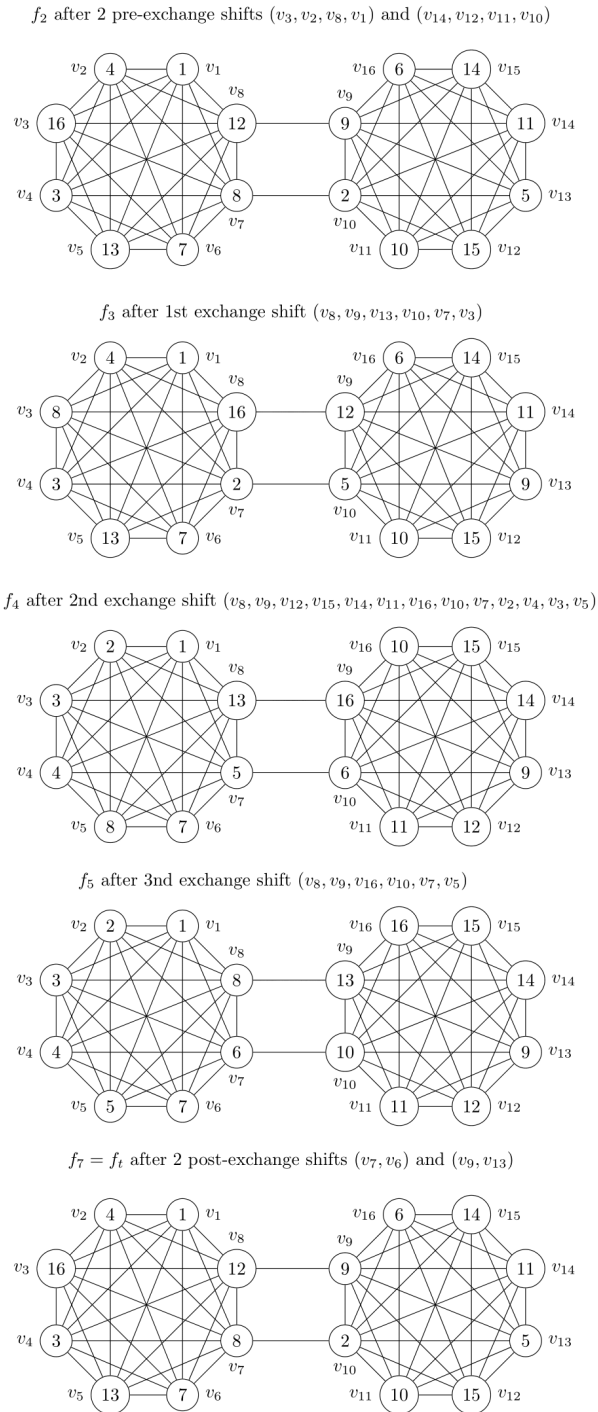


Figure 4.4: An example of Labeled Token Shifting on a generalized barbell graph with 2 bars

### 4.3 Token Shifting on Generalized Barbell Graphs with $k \geq 2$ Bars

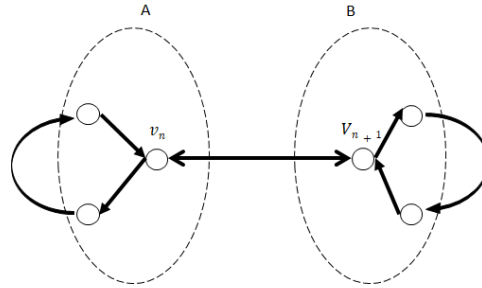
For the next step, we discuss the Labeled Token Shifting Problem on generalized barbell graphs with  $k > 2$  bars. Here,  $G$  is a graph consisting of two equal cliques  $A$  and  $B$  connected by  $k$  edges, called *bars*, such that no two bars are incident to the same vertex. Let  $F = \text{foreign}(A) = \text{foreign}(B)$ , defined as usual.

**Theorem 6.** *The Labeled Token Shifting Problem on a generalized barbell graph  $G = (V, E)$  with  $k \geq 2$  bars can be solved with an optimal shift sequence that satisfies  $F/\lfloor k/2 \rfloor \leq |\text{OPT}| \leq F/\lfloor k/2 \rfloor + 4$ .*

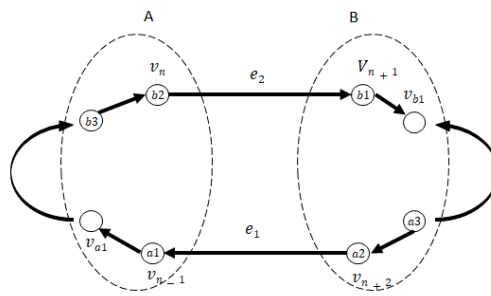
*Proof.* In the previous section, we proved that token shifting on a barbell graph with 2 connecting edges for  $2F$  foreign tokens uses  $F + 4$  shifts: 2 local shifts for moving foreign tokens on gate vertices at the start,  $F$  shifts for exchanging foreign tokens between cliques, and 2 local shifts to rearrange tokens within cliques. Now, while the number of local shifts remains the same, the number of exchanging shifts decreases as  $k$  increases.

Half of the  $k$  edges can be used to move the foreign tokens from  $A$  to  $B$  and another half of the  $k$  edges can be used to move foreign tokens from  $B$  to  $A$ . In one shift, we can exchange  $k$  tokens for even  $k$  and  $k - 1$  tokens for odd  $k$  (see Figure 4.5). Thus, for  $F$  tokens, we only need  $F/\lfloor k/2 \rfloor$  shifts.  $\square$

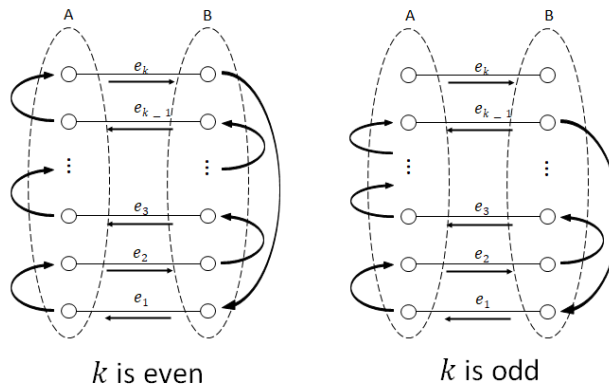




(a)



(b)



(c)

Figure 4.5: Representation of token shifting on (a) a barbell graph (b) a generalized barbell graph with 2 bars (c) a generalized barbell graph with  $k > 2$  bars

# Chapter 5

## 2-Colored Token Shifting on Block Graphs

In this chapter, we discuss the 2-Colored Token Shifting Problem on block graphs. A block graph (or a clique tree) is a graph in which every bi-connected component (block) is a clique (see Figure 5.1).

**Definitions.** In order to state this section's result, we need some definitions. Given a block graph  $G = (V, E)$ , where a block is a maximal clique, an *articulation point* is a vertex that belongs to more than one block. Let  $P \subseteq V$  be the set of articulation points of  $G$ , and let  $K$  be the set of blocks of  $G$ . We define the *tree representation* of  $G$  (see [16]) as the undirected graph  $T(G) = (V', E')$ , where  $V' = P \cup K$  and

$$E' = \{\{k, p\} \mid \text{the articulation point } p \in P \text{ lies in the block } k \in K\}.$$

When referring to  $T(G)$ , the nodes in  $P$  are called *articulation nodes*, and the nodes in  $K$  are called *clique nodes*. Figure 5.1(c) shows an example of a tree representation. For a clique node  $k \in K$ , we write  $I(k)$  to indicate the vertices of  $G$  that are in the block  $k$  but are not articulation points, i.e.,  $I(k) = k \setminus P$ . Note that  $I(k)$  induces a (possibly empty) clique in  $G$ .

Now, let  $G = (V, E)$  be a block graph with  $n$  vertices, let  $\text{Col} = \{c_1, c_2\}$  be the color set, and let  $f_0$  and  $f_t$  be the initial and target token arrangements on  $G$ . We say that an articulation node  $p \in P$  *holds* color  $c \in \text{Col}$  if  $f_0(p) = c$ . Also, if  $f$  is an arbitrary arrangement, we write  $n_c(f(p)) = 1$  if  $f(p) = c$ , and  $n_c(f(p)) = 0$  otherwise. Similarly, for a clique node  $k \in K$ , let  $n_c(f(k))$  be the number of  $c$ -colored tokens in  $I(k) \subseteq V$  in the arrangement  $f$ . Then, we say that a clique node  $k$  of  $T(G)$  *holds* color  $c$  if  $n_c(f_0(k)) > n_c(f_t(k))$ .

For each node  $x$  in  $T(G)$ ,  $x$  has a *value* of  $n_{c_1}(f_0(x)) - n_{c_1}(f_t(x))$ . For each edge  $e$  in  $E'$  connecting two nodes  $k \in K$  and  $p \in P$ , we define the

number  $\text{diff}(e)$  as follows (cf. [17]). Let  $T_k$  be the subtree including node  $k$  resulted by the removal of  $e$  from  $T(G)$ .  $n_{c_1}(f(T'))$  is the number of  $c_1$  tokens on the set of vertices of  $G$  represented by  $T'$  in arrangement  $f$ . Then,  $\text{diff}(e) = n_{c_1}(f_t(T_k)) - n_{c_1}(f_0(T_k))$ , i.e., the difference in the number of  $c_1$  tokens on  $T'$  between  $f_0$  and  $f_t$ . For simplicity,  $\text{diff}(e)$  can be defined as the number of  $c_1$  tokens (and, symmetrically, also  $c_2$  tokens) that we must move along  $e$  to transform  $f_0$  into  $f_t$ . If  $\text{diff}(e) = d > 0$ , it means we need to move  $d$  tokens of color  $c_1$  to  $k$ . If  $\text{diff}(e) = -d < 0$ , it means we need to move  $d$  tokens of color  $c_2$  to  $k$ .

Finally, we define  $E'_k \subseteq E'$  to be the set of the edge of  $T(G)$  that are incident to the clique node  $k$ .

**Theorem 7.** *For the 2-Colored Token Shifting Problem on a block graph  $G = (V, E)$ , we have*

$$\sum_{k \in K} \max_{e \in E'_k} \{|\text{diff}(e)|\} \leq |\text{OPT}| \leq \sum_{k \in K} \max \left\{ \sum_{\substack{e \in E'_k \\ \text{diff}(e) > 0}} \text{diff}(e), \sum_{\substack{e \in E'_k \\ \text{diff}(e) < 0}} |\text{diff}(e)|, 1 \right\},$$

and a shift sequence within these bounds can be computed in  $O(n^2)$  time.

*Proof.* For the upper bound, we will give a procedure for finding a shift sequence. We first construct the tree representation  $T(G)$  in  $O(n^2)$  time. From  $T(G)$ , we determine the sequence of shifts by deciding on which clique the shift must be performed in each step (note that, in a block graph, every cycle is included in a single clique).

For a clique  $k$  with an excess of  $c_1$  tokens connected to an articulation vertex  $p$ , some  $c_1$  tokens in  $k$  must be moved out and some  $c_2$  tokens must be moved in through  $p$ . We need to perform a shift that moves the extra  $c_1$  token in  $k$  to the articulation vertex  $p$  and the  $c_2$  tokens on  $p$  to the target vertex in  $k$ . On  $T(G)$ , it will be a token exchange between a clique node  $k$  that holds color  $c_1$  and the articulation node  $p$  that holds color  $c_2$  along the edge  $e = \{k, p\} \in E'$ . This exchange will decrease  $|\text{diff}(e)|$  and change the color of  $p$  to  $c_1$ . However, in the case where the  $p$  holds the same color  $c_1$  as  $k$ , it is pointless to perform a shift between them. The same goes for a clique with  $n_{c_2}(f_0(k)) > n_{c_2}(f_t(k))$ . If  $\text{diff}(e) = 0$ , no token needs to be moved across  $e$ , and  $e$  can be removed from  $T(G)$ . For  $G$  to achieve the target arrangement  $f_t$ , all the edges in  $T(G)$  must be removed. Thus, we can construct the shift sequence for  $G$  from  $T(G)$  by determining the clique nodes for an exchange in each step.

We now discuss how to choose a feasible clique node for token exchange. There are three types of clique nodes in  $T(G)$ :

1. *leaf node*: a clique node with an articulation node, the removal of which will disconnect the clique node from the other clique nodes in  $T(G)$ . When we look for a clique for token exchange, we start with the leaf nodes and go up the tree  $T(G)$ . A leaf node  $k$  connected to node  $p$  by edge  $e$  is feasible for an exchange if  $k$  and  $p$  hold different colors and  $|\text{diff}(e)| > 0$ .
2. *non-leaf node*: a clique node with more than one articulation nodes connecting it to other clique nodes in  $T(G)$ . In non-leaf nodes, we can exchange one or more pairs of different color tokens in one shift. For a non-leaf node  $k$  with  $m$  articulation nodes  $p_1, p_2, \dots, p_m$ ,  $k$  is feasible for an exchange (1) if there are one or more edges  $e = (k, p)$  with  $|\text{diff}(e)| > 0$ , and  $k$  and  $p$  hold different colors, where  $p \in \{p_1, p_2, \dots, p_m\}$  and  $k$  has non-zero value or (2) if  $k$  is connected to one or more pairs of articulation nodes  $p_i$  and  $p_j \in \{p_1, p_2, \dots, p_m\}$  where  $p_i$  and  $p_j$  hold different colors, and  $\text{diff}(e_i = \{k, p_i\})$  and  $\text{diff}(e_j = \{k, p_j\})$  have opposite sign (one positive, one negative).
3. *isolated node*: a clique node that is disconnected from other clique nodes in  $T(G)$  as the amount of both  $c_1$  and  $c_2$  tokens in it is the same for  $f_0$  and  $f_t$ . For each isolated node  $k$  with no edge in  $T(G)$ , if  $f_0(k) \neq f_t(k)$ , then one shift suffices to reach the target arrangement as  $n_c(f_0(k)) = n_c(f_t(k))$ ,  $c \in \{c_1, c_2\}$ .

Each shift exchanges at least a pair of different color tokens and thus decreases a pair of  $+\text{diff}(e)$  and  $-\text{diff}(e)$  value (if the exchange is between two  $p$  nodes) or decreases a  $+\text{diff}(e)$  or  $-\text{diff}(e)$  value (if the exchange is between the  $p$  and  $k$  node). Including the possible case of isolated clique which needs 1 shift, we get the upper bound for the number of shift in a clique corresponding to  $k$  as:

$$\max \left\{ \sum_{\substack{e \in E'_k \\ \text{diff}(e) > 0}} \text{diff}(e), \sum_{\substack{e \in E'_k \\ \text{diff}(e) < 0}} |\text{diff}(e)|, 1 \right\}$$

As for the lower bound, we observe that, for each clique node  $k$ , we can only move one token to or from each articulation point in a shift and decrease the  $|\text{diff}(e)|$  of each edge by one. Therefore, if  $k$  is incident to an edge  $e$  with  $|\text{diff}(e)| = d$ , then at least  $d$  shifts must be performed in the clique corresponding to  $k$ . Thus, to remove all the edges incident to a clique node  $k$  in  $T(G)$ , at least  $\max_{e \in E'_k} \{|\text{diff}(e)|\}$  shifts are necessary. Figures 5.1 and

5.2 show a step-by-step example of 2-Colored Token Shifting on a block graph.  $\square$

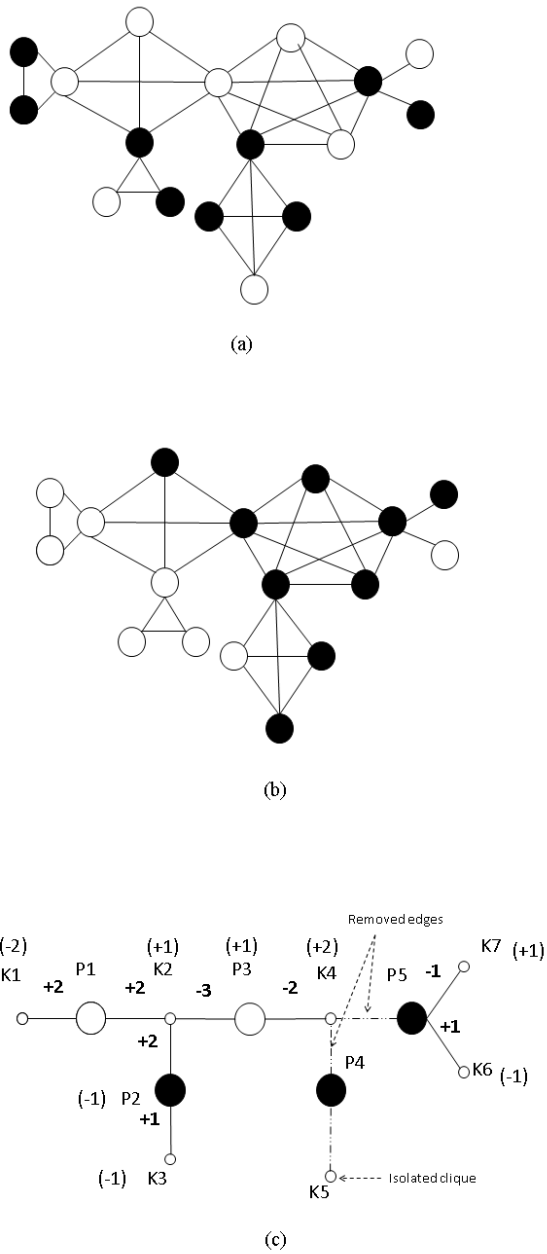
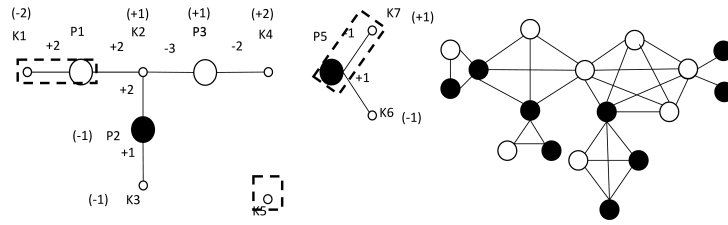
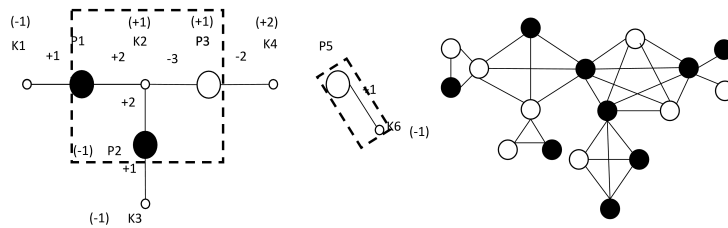


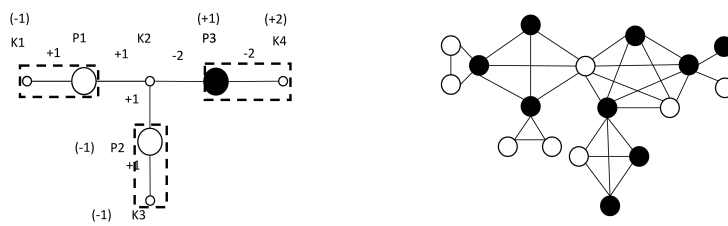
Figure 5.1: 2-colored Token Shifting on a block graph: (a) Initial arrangement  $f_0$ , (b) target arrangement  $f_t$ , and (c) tree representation  $T(G)$  of the block graph  $G$  with positive values over nodes that need black tokens, negative values over nodes that need white tokens,  $\text{diff}(e)$  values over each edge  $e$



(a)

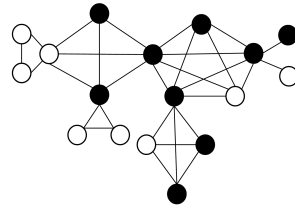
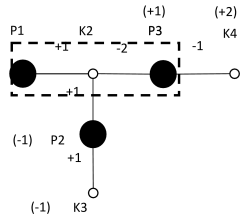


(b)

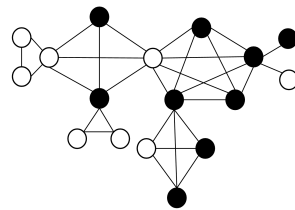
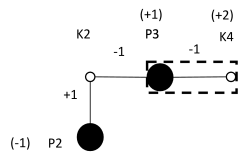


(c)

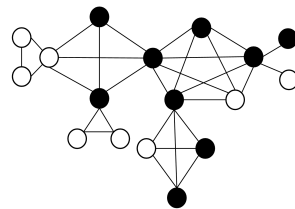
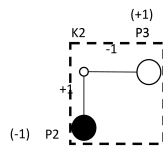
Figure 5.2: An example of 2-colored Token Shifting on a block graph  $G$ :  $T(G)$  with dashed rectangle marking the feasible cliques for the shift (left) and the resulting token arrangement after the shifts (right)



(d)



(e)



(f)

Figure 5.2: An example of 2-colored Token Shifting on a block graph  $G$ :  $T(G)$  with dashed rectangle marking the feasible cliques for the shift (left) and the resulting token arrangement after the shifts (right) (Cont.)



# Chapter 6

## Hardness of 2-Colored Token Shifting

In this chapter, we show that the shortest shift sequence for the 2-Colored Token Shifting Problem is not only NP-hard to compute but also NP-hard to approximate within a factor of  $1/2 + \varepsilon$ , for any  $\varepsilon > 0$ . This is true even if the graph  $G$  is a grid graph, hence planar and with maximum degree 4. We will prove it by a reduction from the NP-complete problem of deciding if a grid graph has a Hamiltonian cycle, i.e., a cycle involving all vertices [18].

**Theorem 8.** *The optimal shifting sequence for the 2-Colored Token Shifting Problem is NP-hard to approximate within a factor of  $1/2 + \varepsilon$ , for any  $\varepsilon > 0$ , even for grid graphs.*

*Proof.* Let  $G = (V, E)$  be a grid graph (i.e., a vertex-induced finite subgraph of the infinite grid), and let a *checkerboard arrangement* be an arrangement of two-colored tokens on  $G$  such that tokens on any two adjacent vertices have different colors. Note that, for any given  $G$ , there are exactly two different checkerboard arrangements.

Our reduction maps the grid graph  $G$  to the 2-Colored Token Shifting Problem on the same graph  $G$ , where the initial arrangement  $f_0$  and the target arrangement  $f_t$  are the two distinct checkerboard arrangements (see Figure 6.1).

Observe that  $f_0(v) \neq f_t(v)$  for all  $v \in V$ , and thus a sequence of shift operations that transforms  $f_0$  into  $f_t$  must move every token at least once. More precisely,  $f_t$  is reached if and only if every token takes part in an odd number of shift operations. If  $G$  has a Hamiltonian cycle  $C$ , then the shift operation along  $C$  immediately transforms  $f_0$  into  $f_t$ , and hence  $|\text{OPT}| = 1$ . Conversely, if  $|\text{OPT}| = 1$ , the single shift operation that transforms  $f_0$  into  $f_t$  must involve every vertex, and thus it must be a Hamiltonian cycle.

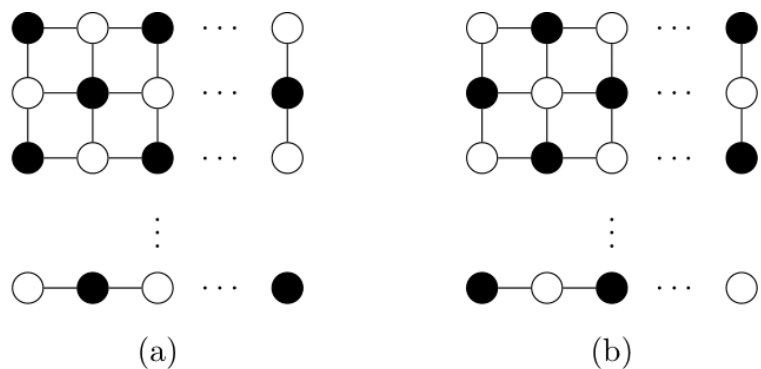


Figure 6.1: 2-colored Token Shifting on a grid graph: (a) Initial arrangement  $f_0$  and (b) target arrangement  $f_t$

We have proved that, if  $G$  has a Hamiltonian cycle, then  $|\text{OPT}| = 1$ , and that if  $G$  does not have a Hamiltonian cycle, then  $|\text{OPT}| \geq 2$ . Thus, if we could compute an approximation of  $|\text{OPT}|$  within a factor of  $1/2 + \varepsilon$  in polynomial-time, we would also be able to decide if  $G$  has a Hamiltonian cycle. Since the latter problem is NP-hard [18], then so is the former problem.  $\square$

# Chapter 7

## Conclusion

### 7.1 Summary

In this thesis, we presented linear time algorithms for optimally solving the 2-Colored Token Shifting Problem on complete graphs as well as the Labeled Token Shifting Problem on complete graphs and variants of barbell graphs. We also gave the  $O(n^2)$  procedure for solving and computing upper and lower bounds of the 2-Colored Token Shifting Problem on block graphs. We then show that, in the 2-Colored Token Shifting Problem, the shortest sequence of moves is NP-hard to approximate within a factor of  $1/2 + \varepsilon$ , even for grid graphs. Notably, our NP-hardness result settles a problem left open in [14], which asked whether the Token Shifting Problem remains NP-hard when restricted to planar graphs or graphs of constant maximum degree.

### 7.2 Remaining Work

In Corollary 2.1, we showed that  $k$ -Colored Token Shifting Problem on a complete graph has  $|\text{OPT}| \leq 2$  and can be solved in 2 shifts. However, for the  $k$ -Colored Token Shifting Problem with  $k > 2$ , we do not have an efficient algorithm to determine when  $|\text{OPT}| = 1$  and when  $|\text{OPT}| = 2$ . We leave this as an open problem. It will also be interesting to extend the problem of Labeled Token Shifting on a generalized barbell graph with  $k > 2$  bars in Chapter 4 and investigate the case where not all  $k$  bars are independent i.e. some of the bars may share a common vertex.

# Bibliography

- [1] Naomi Nishimura. Introduction to reconfiguration. *Algorithms*, 11(4):52, (2018).
- [2] Amer E Mouawad, Naomi Nishimura, Venkatesh Raman, Narges Simjour, and Akira Suzuki. On the parameterized complexity of reconfiguration problems. *Algorithmica*, 78(1):274–297, 2017.
- [3] Arthur Cayley. Lxxvii. note on the theory of permutations. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 34(232):527–529, 1849.
- [4] Donald Ervin Knuth. *The art of computer programming*, volume 3. Pearson Education, 1997.
- [5] Mark R Jerrum. The complexity of finding minimum-length generator sequences. *Theoretical Computer Science*, 36:265–289, 1985.
- [6] Sheldon B. Akers and Balakrishnan Krishnamurthy. A group-theoretic model for symmetric interconnection networks. *IEEE transactions on Computers*, 38(4):555–566, 1989.
- [7] Frederick J Portier and Theresa P Vaughan. Whitney numbers of the second kind for the star poset. *European Journal of Combinatorics*, 11(3):277–288, 1990.
- [8] Igor 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.
- [9] Theresa P Vaughan. Factoring a permutation on a broom. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 30:129–148, 1999.

- [10] Jun Kawahara, Toshiki Saitoh, and Ryo Yoshinaka. The time complexity of the token swapping problem and its parallel variants. In *International Workshop on Algorithms and Computation*, pages 448–459. Springer, 2017.
- [11] Katsuhisa Yamanaka, Erik D Demaine, Takehiro Ito, Jun Kawahara, Masashi Kiyomi, Yoshio Okamoto, Toshiki Saitoh, Akira Suzuki, Kei Uchizawa, and Takeaki Uno. Swapping labeled tokens on graphs. *Theoretical Computer Science*, 27:81–94, June (2015).
- [12] Gaku Yasui, Kouta Abe, Katsuhisa Yamanaka, and Takashi Hirayama. Swapping labeled tokens on complete split graphs. *Inf. Process. Soc. Jpn. SIG Tech. Rep*, pages 1–4, 2015.
- [13] Ahmad Biniiaz, Kshitij Jain, Anna Lubiw, Zuzana Masárová, Tillmann Miltzow, Debajyoti Mondal, Anurag Murty Naredla, Josef Tkadlec, and Alexi Turcotte. Token swapping on trees. *arXiv preprint arXiv:1903.06981*, (2019).
- [14] Kwon Kham Sai, Ryuhei Uehara, and Giovanni Viglietta. Cyclic shift problems on graphs. In *WALCOM: Algorithms and Computation*, pages 308–320. Lecture Notes in Computer Science Vol. 12635, Springer-Verlag, (2021).
- [15] Kazuyuki Amano, Yuta Kojima, Toshiya Kurabayashi, Keita Kurihara, Masahiro Nakamura, Ayaka Omi, Toshiyuki Tanaka, and Koichi Yamazaki. How to solve the torus puzzle. *Algorithms*, 5(1):18–29, (2012).
- [16] Boris Brimkov and Illya V Hicks. Memory efficient algorithms for cactus graphs and block graphs. *Discrete Applied Mathematics*, 216:393–407, (2017).
- [17] Katsuhisa Yamanaka, Takashi Horiyama, David Kirkpatrick, Yota Otachi, Toshiki Saitoh, Ryuhei Uehara, and Yushi Uno. Swapping colored tokens on graphs. In *Workshop on Algorithms and Data Structures*, pages 619–628. Springer, (2015).
- [18] Alon Itai, Christos H Papadimitriou, and Jayme Luiz Szwarcfiter. Hamilton paths in grid graphs. *SIAM Journal on Computing*, 11(4):676–686, (1982).