On the Complexity of Lattice Puzzles

Yasuaki Kobayashi
Graduate School of Informatics, Kyoto University, Yoshida-honmachi, Sakyo-ku,
Kyoto 606-8501 Japan
kobayashi@iip.ist.i.kyoto-u.ac.jp

Koki Suetsugu
National Institute of Informatics, Hitotsubashi, Chiyoda-ku, Tokyo 101-8430 Japan
suetsugu.koki@gmail.com

Hideki Tsuiki
Graduate School of Human and Environmental Studies, Kyoto University, Yoshida Nihonmatsu-cho,
Sakyo-ku, Kyoto 606-8501 Japan
tsuiki.hideki.8e@kyoto-u.ac.jp

Ryuhei Uehara
School of Information Science, Japan Advanced Institute of Science and Technology, Asahidai,
Nomi, Ishikawa 923-1292, Japan
uehara@jaist.ac.jp

Abstract

In this paper, we investigate the computational complexity of lattice puzzle, which is one of the traditional puzzles. A lattice puzzle consists of $2n$ plates with some slits, and the goal of this puzzle is to assemble them to form a lattice of size $n \times n$. It has a long history in the puzzle society; however, there is no known research from the viewpoint of theoretical computer science. This puzzle has some natural variants, and they characterize representative computational complexity classes in the class NP. Especially, one of the natural variants gives a characterization of the graph isomorphism problem. That is, the variant is GI-complete in general. As far as the authors know, this is the first non-trivial GI-complete problem characterized by a classic puzzle. Like the sliding block puzzles, this simple puzzle can be used to characterize several representative computational complexity classes. That is, it gives us new insight of these computational complexity classes.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness

Keywords and phrases Lattice puzzle, NP-completeness, GI-completeness, FPT algorithm

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2019.32

Related Version An early version was presented orally at JCDCGGG 2017 `http://www.jcdcgg.u-tokai.ac.jp/past_conferences/2017.html`.

Funding This work is partially supported by KAKENHI grant numbers 15K00015, 17H06287, and 18H04091, and JST CREST JPMJCR1401.

Acknowledgements The authors thank Prof. Shuji Yamada for his introduction of his original designed lattice puzzle to the first three authors. The authors also thank JCDCGGG 2017 for giving a chance to give an oral presentation of the early work on this topic, which motivated the last author to join this research. The last author thanks Hisayoshi Akiyama and Mineyuki Uyematsu who kindly taught him the history of the lattice puzzle.

1 Introduction

In history of theoretical computer science, some puzzles and games play important roles for giving reasonable characterization to computational complexity classes. For example, Conway's game of life is universal [2], and it essentially has the same computational power of the Turing machine. Later, "pebble game" was proposed as a classic model that gives some complexity classes in a natural way (see, e.g., [6]). The "constraint logic" is a recent model that succeeds to solve a long standing open problem due to Martin Gardner that asks the computational complexity of sliding block puzzles [5]. Such puzzles and games have been giving us some intuitive insight for some computational complexity classes.

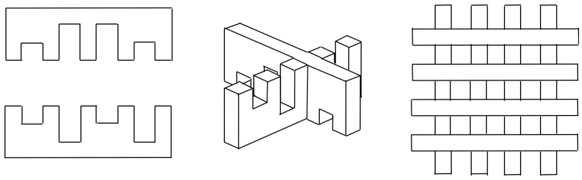

Figure 1 Illustration of a lattice puzzle.

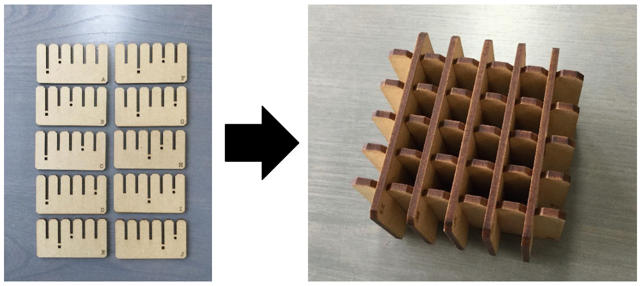

Figure 2 The first lattice puzzle invented by T. Betsumiya (crafted by M. Uyematsu).

In this paper, we investigate the lattice puzzle and its variants. Typical one is illustrated in Figure 1: we are given some plates with slits, and the goal is to assemble them into the form of a lattice. There are many variants of them. In Japanese puzzle society, they say that the first one was called "cross block", and invented by a Japanese puzzle designer, Toshiaki Betsumiya, in 1992 [1] (see Figure 2). This puzzle consists of ten plates. Each plate has five slits; three slits have depth $1/2$, one slit has depth $1/4$, and the last slit has depth $3/4$ (we normalize it to 1 for simplicity). This puzzle has a beautiful mathematical design ($\binom{5}{2} = 10$), and it is reasonably difficult (it has 20 solutions, however, it is difficult to find one). Since then, many variants are invented and are on the market.

It is natural to consider two variants of the cross block. First, the slits are on the same side of a plate, we call it *one-sided*. We can consider *two-sided* plates whose slits are on both sides as Figure 3. In the cross block, when two plates are assembled, there is no gap at the crossing point. We call it *fit*. That is, two slits of depth p and q can be assembled only if $p + q = 1$ in the fit model. In the *loose* model, we permit to assemble when $p + q > 1$. Some readers may think that the two-sided model should be loose; otherwise we may not be able to assemble/disassemble the plates as Figure 3(b). Actually, the two-sided fit model was invented in the puzzle society (see Figure 4). In this commercial product, each piece is made by rubber, and the puzzle itself is in the fit model. However, we can still consider the problem of assembling/disassembling. Even if the final form is feasible in the loose model, we may not assemble when the plates are rigid. In this paper, we focus on the problem that asks whether the assembled form is feasible or not. This assembling problem is another problem, which is not dealt with in this paper.

We here note that there is an old application of this problem. In the classic Japanese wood craft, called "kumiki" which means "assembling wood", there is a tricky method to combine bars so that they seem to be "impossible" to assemble. Such a kumiki is actually

(a) (b)

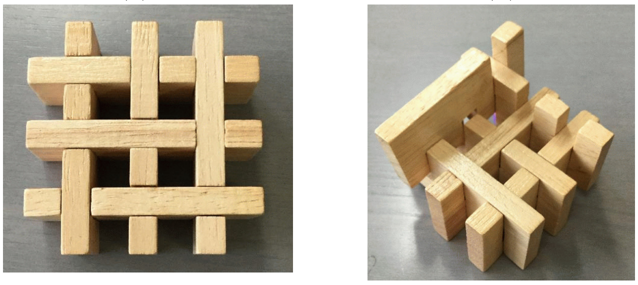

Figure 3 Two-sided lattice puzzle.

Figure 4 Two-sided lattice puzzle with no gap and one depth.

the puzzle which can be seen as the two-sided loose model (Figure 5). This kumiki pattern is known as "chidori-goshi", which means "thousand-of-birds lattice". For example, this craft can be found at the Kiyomizu-temple in Kyoto, which is one of the most famous temples in Japan (Figure 6). This kumiki method is too old and traditional to find the inventor[1]. As a mark of respect for the inventor, we name this puzzle *lattice puzzle*.

As seen in Figure 6, we can consider the lattice puzzle of size $n \times m$ in general. When $n \neq m$, we can distinguish horizontal plates and vertical plates. However, if the lattice is square of size $n \times n$, we can consider a variant that asks us to partition the given set of $2n$ plates into two sets of n plates. We call this variant a *square lattice puzzle*. We also consider two variants based on the set of operations we use to solve a puzzle. To obtain a solution, one needs to order the plates in a sequence in each set. We call this process the permutation process. One also needs to flip the pieces so that they have correct directions and correct upsides. Note that each plate can be flipped to change its direction in the one-sided model, and to change its direction and its upside in the two-sided model. If the puzzle has a hint of the direction and the upside by, for example, coloring one of the endpoints and one of the sides of each plate, then one can only permute the pieces. We call this variant *permutation lattice puzzle*. If the puzzle has a hint of the order of each set of plates by, for example, assigning numbers to each set of plates, then one can only flip the pieces. We call this variant *flip lattice puzzle*.

[1] We found one in a literature published in 1770 at `http://www.wul.waseda.ac.jp/kotenseki/html/i16/i16_00875/index.html`.

Figure 5 Traditional wooden craft in Japan (Kumiki).

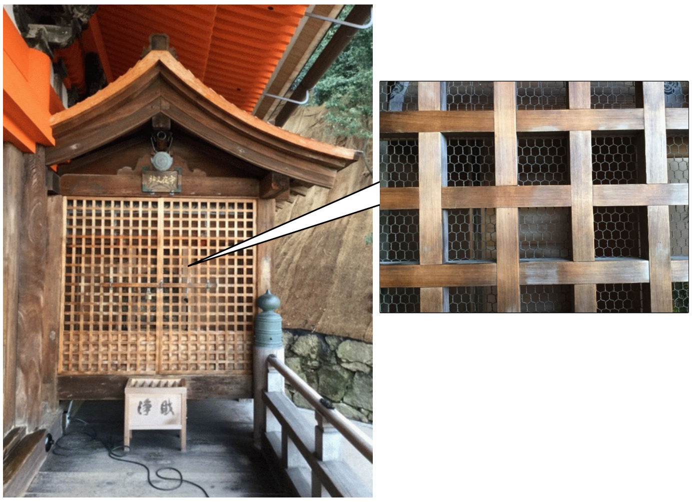

Figure 6 Large (im)possible wooden lattice at Kiyomizu temple in Kyoto.

We here summarize the table of our results in Table 1. It is easy to observe that the lattice puzzle is in the class NP since we can check the feasibility of a given solution in polynomial time. We show in Section 3 that the one-sided permutation loose model with 3 depths is NP-complete. This implies that the puzzle is NP-complete in general in the most flexible variant; two-sided, loose, and the number of different depths is not bounded.

On the other hand, we show in Section 4 that when we turn to the one-sided permutation fit model with 2 depths, this puzzle is GI-complete in general. The graph isomorphism problem (GI) is a decision problem that, given two graphs, decides whether they are isomorphic or not. This problem is one of the most well-studied problems in computational complexity and some related areas, and is believed to be in neither P nor NP-complete. There are several work to seek problems that are equivalent to GI. A problem is said to be GI-complete if the problem is as hard as the GI problem for general graphs. (Precisely speaking, a problem P is GI-complete if and only if the graph isomorphism problem for general graphs can be reducible to P and vice versa under polynomial time reduction.) As far as the authors know, there is no known characterization of the GI problem with such a simple puzzle.

Table 1 Summary of results.

square/ colored	one/two sided	operations	#depths	rule	complexity	note
any	one-sided	permutation	3	loose	NP-complete	Theorem 2
colored	one-sided	permutation	2	fit	GI-complete	Theorem 3
any	one-sided	both	3	fit	GI-hard	Corollary 4
colored	one-sided	flip	unbounded	any	poly	Theorem 5
$n \times k$ (k:fixed)	any	both	unbounded	any	FPT for k	Theorem 6

Finally, we show in Section 5 that when we turn to the one-sided flip fit model, the problem can be reduced to the 2SAT problem, which can be solved in linear time for the size of the puzzle. We also consider the case that the lattice size is bounded as $n \times k$ for a fixed constant k. In this case, we show that the problem is fixed parameter tractable for k. That is, this problem can be solved in $f(k)p(n)$ time, where $p(n)$ is a polynomial function of n.

2 Definition of lattice puzzles and preliminaries

We first explain the one-sided model. We assume that each instance of a lattice puzzle P is given by two sets $X = \{x_1, \dots, x_n\}$ and $Y = \{y_1, \dots, y_m\}$ of *plates*. Each plate in X and Y is a rectangle of size $1 \times (m+1)$ and $1 \times (n+1)$, respectively. Let x be a plate in X. On x, we have m *slits* uniformly spaced on the long side. In the one-sided model, we denote the depth of the ith slit from the left by $d_i(x)$ with $0 < d_i(x) < 1$ (since the plate is disconnected if $d_i(x) = 1$, and we have no feasible solution if $d_i(x) = 0$). We define $d_j(y)$ in the same manner for each $y \in Y$. We distinguish a plate and the one obtained by flipping it, and denote by $\text{flip}(x)$ the plate obtained by flipping x. That is, if n is the number of slits of x, $d_i(\text{flip}(x)) = d_{n+1-i}(x)$ for every $1 \leq i \leq n$. We say that two plates x and y *fit* at point (i, j) if $d_i(x) + d_j(y) = 1$, and *weakly fit* if $d_i(x) + d_j(y) \geq 1$.

A solution of the puzzle is an arrangement of the plates in X and Y so that they form a lattice as shown in Figure 1 and Figure 2. More precisely, a solution is a pair of lists of plates $[x'_1, \dots, x'_n]$ and $[y'_1, \dots, y'_m]$ such that every x'_i is x or $\text{flip}(x)$ for some different $x \in X$ and every y'_j is y or $\text{flip}(y)$ for some different $y \in Y$ and x'_i and y'_j (weakly) fit at (j, i) for $1 \leq i \leq n$ and $1 \leq j \leq m$. That is, we have $d_j(x'_i) + d_i(y'_j) = 1$ in the fit model, and $d_j(x'_i) + d_i(y'_j) \geq 1$ in the loose model. We consider three different models based on the operations we can use. The above model is called the *all-operations model*. In the *permutation model*, neither $\text{flip}(x)$ nor $\text{flip}(y)$ do not appear in the list. In the *flip model*, the sets X and Y are ordered from the beginning and we are given two lists $[x_1, \dots, x_n]$ and $[y_1, \dots, y_m]$, and x'_i is x_i or $\text{flip}(x_i)$ for $1 \leq i \leq n$ and y'_j is y_j or $\text{flip}(y_j)$ for $1 \leq j \leq m$.

We here note for the special case $|X| = |Y| = n$. In this case, we cannot distinguish the plates of X and Y from their shapes. Thus, we can consider a variant of the lattice puzzle that a set of $2n$ plates is given and one divides it into two sets X and Y of n plates. We call this variant an *$n \times n$ square lattice puzzle*. Note that when we simply say an $n \times n$ lattice puzzle, we consider that two sets X and Y of plates are given. In this case, we sometimes say it is *colored* (as Figure 5) to emphasis the model.

In the two-sided model, we select one side and call it the positive side and the other one the negative side. If the ith slit of x is on the negative side, then we define $d_i(x)$ as $-1 < d_i(x) < 0$ according to the length of the slit. (That is, we do not allow to have two slits on both sides at the same position.) In this model, we allow two kinds of flip operations. That is, for a plate x, the plate $\text{flip}(x)$ such that $d_i(\text{flip}(x)) = d_{n+1-i}(x)$ for every i and the plate sflip(x) such that $d_i(\text{sflip}(x)) = -d_i(x)$ for every i. We say that two plates x and y *fit* at point (i, j) if $d_i(x) + d_j(y) = \pm 1$ and *weakly fit* if $|d_i(x) + d_j(y)| \geq 1$. The solution in the two-sided model can be defined in a similar way.

By default, we consider non-square one-sided all-operations 2-depth fit lattice puzzle and we omit these adjectives. We use adjectives square, two-sided, permutation, flipping, n-depth, any number of depth, and loose if they are.

For any given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ with $|V_1| = |V_2|$, a bijection $\phi : V_1 \to V_2$ is said to be an *isomorphism* when $\{u, v\} \in E_1$ if and only if $\{\phi(u), \phi(v)\} \in E_2$. When there is an isomorphism between G_1 and G_2, G_1 is *isomorphic* to G_2. The graph isomorphism problem (GI) is a decision problem that, given two graphs G_1 and G_2, decides whether G_1 is isomorphic to G_2. A problem is *graph isomorphism complete* (GI-complete) if the problem is as hard as the graph isomorphism problem on general graphs. (GI-hardness is defined in the same manner of NP-hardness.) It is known that the graph isomorphism problem is GI-complete even if the input graphs are bipartite graphs (see, e.g., [7]).

In this paper, we did not give a definition of *fixed parameter tractability*; see e.g., [3] for the details. In our context, when an instance of a lattice puzzle P is given by two sets X and Y with $|X| = n$ and $|Y| = k$ for any fixed positive constant k, we say that the lattice puzzle P is fixed parameter tractable if there is an algorithm that solves P in $f(k)p(n)$ time, where f is any function of k, and p is a polynomial function of n.

3 NP-completeness

We first show the following key lemma:

▶ **Lemma 1.** *The lattice puzzle of size $n \times m$ is NP-complete in the loose model with linear number of depths.*

Proof. We reduce the following positive 1-IN-3SAT problem, which is one of the well known NP-complete problems [4], to one-sided lattice puzzle (hence two-sided lattice puzzle immediately follows):

Input: A collection of clauses $C_1, \ldots, C_m$ of variables $x_1, \ldots, x_n$ such that each C_j is a disjunction of exactly three positive literals.

Question: Is there a truth assignment to the variables occurring so that exactly one literal is true in each C_j?

We will use $n' = 2n + 1$ plates for variables and $m' = 3m + 1$ plates for clauses and reduce the above 1-IN-3SAT problem to the problem of solving an $n' \times m'$ lattice puzzle(in Figure 7, vertical plates are for variables, and horizontal ones are for clauses). There are two special horizontal plate p_x and vertical plate p_C which are called *ID organizers* for x and C, respectively. We first define $d_1(p_x) = \epsilon_0$ and $d_1(p_C) = 1 - \epsilon_0$ for sufficiently small $\epsilon_0 > 0$. In the following construction, the depth d of all the other slits satisfy $\epsilon_0 < d < 1 - \epsilon_0$. Therefore, p_x and p_C should be assembled as shown in Figure 7 at the *pivot* $(1, 1)$. We let $0 < \epsilon_0 < \epsilon_1 < \cdots < \epsilon_n < \epsilon'_1 < \cdots < \epsilon'_m < 1/4$ for some small distinct values. Then we define $d_2(p_x) = d_3(p_x) = \epsilon_1$, $d_4(p_x) = d_5(p_x) = \epsilon_2$, $\ldots$, $d_{2i}(p_x) = d_{2i+1}(p_x) = \epsilon_i$, $\ldots$, $d_{2n}(p_x) = d_{2n+1}(p_x) = \epsilon_n$. For these slits, we prepare $2n$ variable plates $x_1^+, x_1^-, x_2^+, x_2^-, \ldots, x_n^+, x_n^-$. For each i with $1 \leq i \leq n$, we let $d_1(x_i^+) = d_1(x_i^-) = 1 - \epsilon_i$. As we will see, the depths of other vertical slits are $1/4$, $1/2$, or $3/4$. Therefore, it is easy to see that each pair of variable plates x_i^+ and x_i^- should be assembled at points $(2i, 1)$ and $(2i+1, 1)$ (or $(2i+1, 1)$ and $(2i, 1)$). In a similar way for the plate p_C with small values $\epsilon'_1, \ldots, \epsilon'_m$, the ID organizer for C organizes

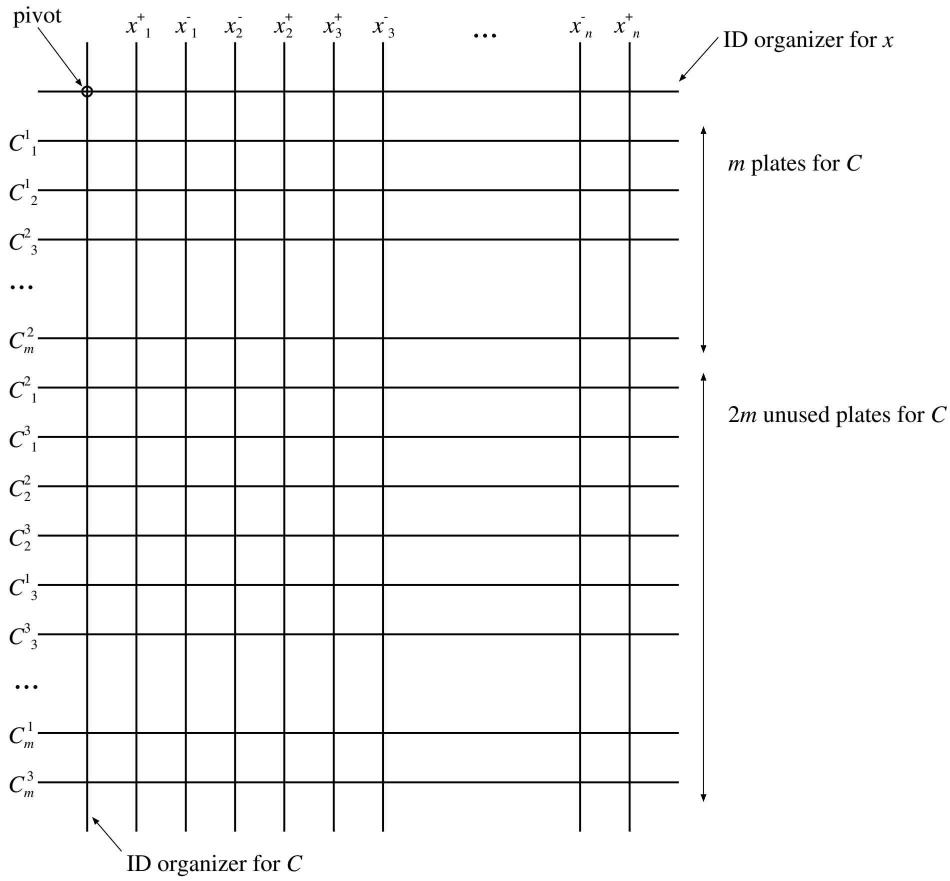

Figure 7 Reduction from 1-IN-3SAT to lattice puzzle.

the clause plates as follows. We first define $d_2(p_C) = \epsilon'_1$, $d_3(p_C) = \epsilon'_2$, …, $d_{m+1}(p_C) = \epsilon'_m$. Then we further define $d_{m+2}(p_C) = d_{m+3}(p_C) = \epsilon'_1$, $d_{m+4}(p_C) = d_{m+5}(p_C) = \epsilon'_2$, …, $d_{m+2j}(p_C) = d_{m+2j+1}(p_C) = \epsilon'_j$, …, $d_{3m}(p_C) = d_{3m+1}(p_C) = \epsilon'_m$. For these slits, we prepare $3m$ clause plates C^1_j, C^2_j, C^3_j $(1 \leq j \leq m)$. For each j with $1 \leq j \leq m$, we let $d_1(C^1_j) = d_1(C^2_j) = d_1(C^3_j) = 1 - \epsilon'_j$. Thus, one of C^1_j, C^2_j, C^3_j is assembled at $(1, j+1)$ and the other two plates are assembled at $(1, m+2j)$ and $(1, m+2j+1)$.

From the construction, we can observe that all ordering of these plates are fixed except (1) we can exchange x^+_i and x^-_i, and (2) we can exchange C^1_j, C^2_j, and C^3_j. Now we give the assignments of depths for each x^+_i, x^-_i, C^1_j, C^2_j, and C^3_j.

First, we define depths of variable plates x^+_i and x^-_i. As depth, we use three values $3/4, 1/4$, and $1/2$. We set $d_j(x) = 3/4$ for all variable plates x and $j > m+1$. Note that a slit with depth $3/4$ matches with any slit in the loose model. Therefore, for each $1 \leq j \leq m$, one can put any two (unused) clause plates C^k_j $(k \in \{1,2,3\})$ at some jth row for $j > m+1$ as shown in the lower part of Figure 7.

For each $1 \leq j \leq m$, suppose that the clause C_j contains variables x_{i_1}, x_{i_2} and x_{i_3}. Then we set $d_{j+1}(x^+_{i_k}) = 3/4$ for each $k \in \{1,2,3\}$. The depth of other slits of variable plates are set to $1/2$.

Next, we define depths of clause plates. Suppose that C_j contains variables x_{i_1}, x_{i_2}, and x_{i_3}. Let $s^+_k = 2i_k$ and $s^-_k = 2i_k + 1$, which are the two indices of the rows at which $x^+_{i_k}$ and $x^-_{i_k}$ can be placed. We assign depth $1/4$ to the slits at the following positions of C^1_j, C^2_j, and

C_j^3. On C_j^1, we assign at positions s_1^+, s_2^-, and s_3^-. On C_j^2, we assign at positions s_2^+, s_3^-, and s_1^-. On C_j^3, we assign at positions s_3^+, s_1^-, and s_2^-. At all the other positions, of clause plates we assign depth 1/2.

Suppose that a solution of the original instance of the 1-IN-3SAT is given. If x_i is true, then we set the plate x_i^+ at position $2i$ and the plate x_i^- at position $2i+1$. If x_i is false, we exchange these two plates. Then, for each $1 \leq j \leq m$, one can observe that exactly one of C_j^1, C_j^2, or C_j^3 can be placed on the $(j+1)$st line, and we put the other two plates on the lines with indices greater than $m+1$. Thus, we obtain a solution of the lattice puzzle. This is a one-to-one correspondence, and one can construct a solution of the 1-IN-3SAT problem from a solution of this puzzle. ◀

In the proof of Lemma 1, except ID organizers for x and C, we need depths 1/4, 1/2, and 3/4. Moreover, flipping is not required in the proof. These facts lead us to the following stronger result.

▶ **Theorem 2.** *The lattice puzzle of size $n \times m$ and the square lattice puzzle of size $n \times n$ are NP-complete in the loose model with 3 depths even if one-sided model and only permutation is permitted.*

Proof. It is enough to show that we can design a "frame" surrounding the puzzle in the proof of Lemma 1. A brief sketch is given in right down in Figure 8. The gray area of the figure forms a frame, which plays ID organizers in the proof of Lemma 1. The magnification of the left up corner of the frame is depicted at left up in Figure 8. Let $n' = \max\{2n, 3m\}$. Then the frame F is the set of $4n'$ plates. For each $i = 1, 2, \ldots, n'$, we have two copies of f_i and two copies of f_i'. The set of f_is plays the role of the ID organizer of C, and the set of f_i's plays the rule of the ID organizer of x.

In Figure 8, each small circle on f_i corresponds to depth 3/4, and each small circle on f_i' corresponds to depth 1/4. The other intersection of f_i and f_j', the depth is 1/2.

We apply the same manner for each intersection between f_i and C_k^j and each intersection between f_i' and x_j^+ or x_j^-. Precisely, for example, f_2 has depth 3/4 at the points corresponding to C_1^1, C_1^2, and C_1^3, and has depth 1/2 at the other points. In general, f_{i+1} has depth 3/4 at the points corresponding to C_i^1, C_i^2, and C_i^3. Similarly, each f_{i+1}' has depth 1/4 at the points corresponding to x_i^+ and x_i^-, and has depth 1/2 at the other points. The corresponding depths of the plates x_j and C_j are trivial.

We note that they fit without any gap in the gray area of the figure. Since there is no gap at the gray area, we can observe that the shape of the frame is uniquely formed by these $4n'$ plates.

Therefore, combining the reduction in Lemma 1, we prove that the lattice puzzle of size $n \times m$ is NP-complete in the loose model with 3 depths.

It is easy to see that the reductions work even in one-sided model and only permutation is permitted. Addition of extra plates to make the frame square is trivial. ◀

4 GI-completeness

In this section, we show the following theorem:

▶ **Theorem 3.** *The permutation lattice puzzle is GI-complete.*

Proof. We show a correspondence between lattice puzzles and graph isomorphism problems of bipartite graphs. Let $G_1 = (A_1, B_1, E_1)$ and $G_2 = (A_2, B_2, E_2)$ be two bipartite graphs with $|A_1| = |A_2| = n$ and $|B_1| = |B_2| = m$. Let $A_1 = \{a_1, a_2, \ldots, a_n\}$, $B_1 = \{b_1, b_2, \ldots, b_m\}$,

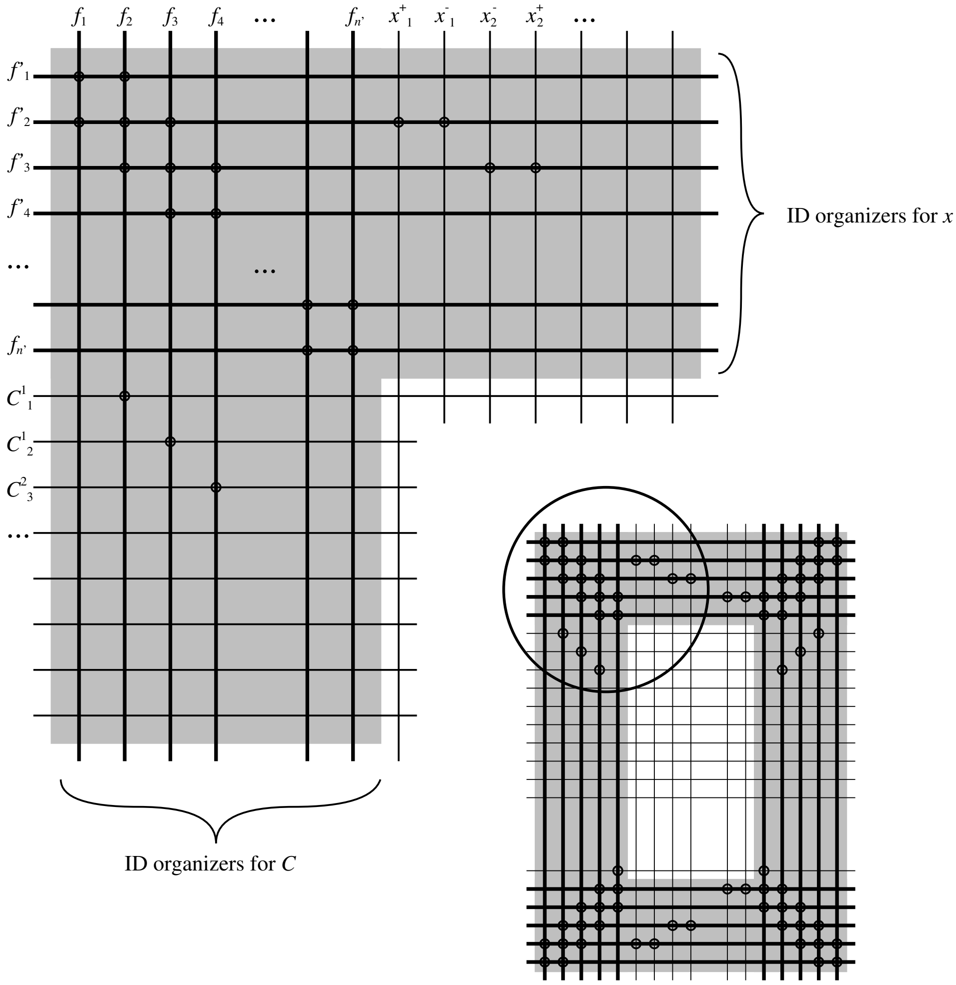

Figure 8 Framing of the lattice puzzle in Lemma 1.

$A_2 = \{a'_1, a'_2, \ldots, a'_n\}$ and $B_2 = \{b'_1, b'_2, \ldots, b'_m\}$. From these graphs, we construct a lattice puzzle as follows. The set $X = \{x_1, \ldots, x_n\}$ of plates is constructed from G_1: the plate x_i of size $1 \times (m+1)$ corresponds to $a_i \in A_1$ and $d_j(x_i) = 2/3$ if $\{a_i, b_j\}$ is in E_1, and $d_j(x_i) = 1/3$ otherwise. The set $Y = \{y_1, \ldots, y_m\}$ of plates is constructed from G_2: the plate y_j of size $1 \times (n+1)$ corresponds to $b'_j \in B_2$ and $d_i(y_j) = 1/3$ if $\{a'_i, b'_j\}$ is in E_2, and $d_i(y_j) = 2/3$ otherwise.

Now we check that a graph isomorphism gives us a solution of the puzzle, and vice versa. Let ϕ_A and ϕ_B be permutations on $\{1, \ldots, n\}$ and $\{1, \ldots, m\}$, respectively. ϕ_A induces a bijection $a_i \mapsto a'_{\phi_A(i)}$ from A_1 to A_2, and ϕ_B induces a bijection $b_j \mapsto b'_{\phi_B(i)}$ from B_1 to B_2. They form a graph isomorphism from G_1 to G_2 if and only if $\{a_i, b_j\} \in E_1 \Leftrightarrow \{a'_{\phi_A(i)}, b'_{\phi_B(j)}\} \in E_2$ for $1 \le i \le n$ and $1 \le j \le m$. On the other hand, ϕ_A induces a permutation on X such that the result of permutation $X' = [x'_1, \ldots, x'_n]$ is given as $x'_i = x_{\phi_A^{-1}(i)}$. In the same way, ϕ_B^{-1} induces a permutation on Y such that the result of permutation $Y' = [y'_1, \ldots, y'_m]$ is given as $y'_j = y_{\phi_B(j)}$.

Now, we look at the point (i, j) where the plates $x'_i = x_{\phi_A^{-1}(i)}$ and $y'_j = y_{\phi_B(j)}$ are crossing. On the plate x'_i, $d_j(x'_i) = 2/3$ if and only if $\{a_{\phi_A^{-1}(i)}, b_j\} \in E_1$ and on the plate y'_j, $d_i(y'_j) = 1/3$ if and only if $\{a'_i, b'_{\phi_B(j)}\} \in E_2$. Therefore, X' and Y' form a solution of the puzzle if and only if $\{a_{\phi_A^{-1}(i)}, b_j\} \in E_1 \Leftrightarrow \{a'_i, b'_{\phi_B(j)}\} \in E_2$ for $1 \le i \le n$ and $1 \le j \le m$. Let $i' = \phi_A^{-1}(i)$. Then, it is equivalent to $\{a_{i'}, b_j\} \in E_1 \Leftrightarrow \{a'_{\phi_A(i)}, b'_{\phi_B(j)}\} \in E_2$ for $1 \le i' \le n$ and $1 \le j \le m$. That is, G_1 and G_2 are isomorphic. ◀

▶ **Corollary 4.**

(1) *The square lattice puzzle is GI-complete.*

(2) *The square permutation lattice puzzle is GI-complete.*

Proof. We reduce a colored permutation puzzle of size $n \times n$ to a square all-operations puzzle of size $(n+4) \times (n+4)$. We denote by 0 a slot with depth 1/3, and by 1 a slot with depth 2/3. Suppose that an $n \times n$ non-square lattice puzzle with the lists of plates $X = [x_1, \ldots, x_n]$, $Y = [y_1, \ldots, y_n]$ is given. From X and Y, we form a new set $\tilde{Z} = \{\tilde{x}_1, \ldots, \tilde{x}_n, \tilde{y}_1, \ldots, \tilde{y}_n, \tilde{z}_1, \ldots, \tilde{z}_8\}$ of plates with $n+4$ slots. The sequences of slots are given as follows. Here, $\overline{x}$ is the sequence of slots of x.

$$
\begin{array}{llll}
\tilde{x}_i & = 1\ 0 \quad \overline{x_i} & 0\ 0 & (1 \le i \le n) \\
\tilde{y}_i & = 0\ 1 \quad \overline{y_i} & 0\ 0 & (1 \le i \le n) \\
\tilde{z}_i & = 1\ 1 \quad 1^n & 0\ 0 & (i = 1, 2, 3, 4) \\
\tilde{z}_i & = 1\ 0 \quad 1^n & 0\ 1 & (i = 5, 6) \\
\tilde{z}_i & = 0\ 1 \quad 0^n & 1\ 0 & (i = 7, 8)
\end{array}
$$

Let $[x'_1, \ldots, x'_{n+4}]$ and $[y'_1, \ldots, y'_{n+4}]$ be the lists of plates which form a solution of this puzzle. We first study $x'_1, x'_2, x'_{n+3}, x'_{n+4}$ and $y'_1, y'_2, y'_{n+3}, y'_{n+4}$. First note that these plates cross at the four 2×2 corners, and each place needs to contain at least one 1-slot. It means we need, in all, 16 1-slots at the positions $1, 2, n+3, n+4$ of the 8 plates. Therefore, these 8 plates must be z_i $(1 \le i \le 8)$. One can see that $[x'_1, x'_2, x'_{n+3}, x'_{n+4}] = [z_1, z_7, \mathrm{flip}(z_2), z_5]$ and $[y'_1, y'_2, y'_{n+3}, y'_{n+4}] = [z_8, \mathrm{flip}(z_3), z_6, z_4]$ form a solution. In this case, for each $3 \le j \le n+2$, the jth slots of $x'_1, x'_2, x'_{n+3}, x'_{n+4}$ form the sequence 0 1 1 1, and those of $y'_1, y'_2, y'_{n+3}, y'_{n+4}$ form the sequence 1 0 1 1. There are other possibilities of the assignments of $x'_1, x'_2, x'_{n+3}, x'_{n+4}, y'_1, y'_2, y'_{n+3}, y'_{n+4}$. However, one can see that, in all of them, we have the same sequences 0 1 1 1 and 1 0 1 1 or their rotation at these slots. It means that $\{x'_i \mid 3 \le i \le n+2\}$ must be $\{\tilde{x}_i \mid 1 \le i \le n\}$ and $\{y'_i \mid 3 \le i \le n+2\}$ must be $\{\tilde{y}_i \mid 1 \le i \le n\}$. Note that $\tilde{x}_i$ and $\tilde{y}_i$ cannot be flipped. Therefore, there is a correspondence between solutions of the original non-square permutation puzzle of size $n \times n$ and this square all-operations puzzle of size $(n+2) \times (n+2)$. ◀

5 Polynomial time algorithms

In this section, we show two variants that can be solved in polynomial time.

5.1 Fixed ordering case

In this variant, we assume that the place of each plate is fixed. That is, the lattice of size $n \times m$ is fixed, and each plate can only be flipped. In such a restricted case, we still have 2^{n+m} possible cases. However, we can solve this variant in linear time:

▶ **Theorem 5.** *The lattice puzzle of size $n \times m$ can be solved in $O(nm)$ time in the flip fit model with 2 depths*[2].

[2] The modification of the number of depths from 2 to any positive integer is straightforward and omitted here.

Proof. Without loss of generality, the set $X = \{x_1, \dots, x_n\}$ of n plates with m slits and the set $Y = \{y_1, \dots, y_m\}$ of m plates with n slits are given, their positions are given by their indices, and the depth of each slit is 1/3 or 2/3. We reduce this puzzle to the 2SAT problem, which can be solved in linear time.

From the set X of n plates, we define a Boolean matrix A of size $n \times m$ which may contain Boolean variables $a_1, \dots, a_n$. For each $1 \le i \le n$ and $1 \le j \le m$, we define

$$A_{i,j} = \begin{cases} T & \text{if } d_j(x_i) = d_{m-j+1}(x_i) = 1/3 \\ F & \text{if } d_j(x_i) = d_{m-j+1}(x_i) = 2/3 \\ a_i & \text{if } d_j(x_i) = 1/3 \text{ and } d_{m-j+1}(x_i) = 2/3 \\ \bar{a_i} & \text{if } d_j(x_i) = 2/3 \text{ and } d_{m-j+1}(x_i) = 1/3. \end{cases}$$

The variable a_i represents the direction of the ith plate. Similarly, the matrix B is defined from the set Y of m plates as

$$B_{i,j} = \begin{cases} T & \text{if } d_i(y_j) = d_{n-i+1}(y_j) = 2/3 \\ F & \text{if } d_i(y_j) = d_{n-i+1}(y_j) = 1/3 \\ b_j & \text{if } d_i(y_j) = 2/3 \text{ and } d_{n-i+1}(y_j) = 1/3 \\ \bar{b_j} & \text{if } d_i(y_j) = 1/3 \text{ and } d_{n-i+1}(y_j) = 2/3, \end{cases}$$

for each $1 \le i \le n$ and $1 \le j \le m$, where $b_1, \dots, b_m$ are Boolean variables. Observe that the plates x_i and y_j can fit at (i, j) if and only if $A_{i,j} = B_{i,j}$.

Now we solve the assignment problem for these variables. This condition can be represented by two clauses as $(\alpha_i^j \wedge \beta_i^j) \vee (\bar{\alpha}_i^j \wedge \bar{\beta}_i^j)$, where α_i^j and β_i^j are the literals appearing in $A_{i,j}$ and $B_{i,j}$, respectively. We consider the 2-CNF formula obtained as their conjunction. Then, this puzzle is solvable if and only if there is a satisfying assignment for this formula.

Suppose that it is satisfied with a variable assignment to $a_1, \dots, a_n, b_1, \dots, b_m$. We obtain a solution of the puzzle with the following procedure: If $a_i = F$, then we flip x_i for $i = 1, \dots, n$. If $b_j = F$, then we flip y_j for $j = 1, \dots, m$. Since the 2SAT problem can be solved in polynomial time, Theorem 5 follows. ◀

5.2 Fixed parameter tractable algorithm

In this variant, we consider the lattice of size $n \times k$ for a fixed constant k. First, we mention that the fit model with one-sided plates is easy to solve by checking all permutations of k plates of size $1 \times (n + 1)$. Considering the flipping, we have $k!2^k$ ways to arrange these k plates. Once we fix one arrangement of k plates, checking of feasibility is straightforward in $O(kn^2)$ time. Therefore, the algorithm runs in $O(k!2^k kn^2)$ time. For the two-sided plates, the number of possible permutations is $k!4^k$, and the checking of feasibility can be done in $O(kn^2)$ time. Therefore, we can solve the lattice puzzle in $O(k!4^k kn^2)$ time. We extend this idea to the loose model:

▶ **Theorem 6.** *The lattice puzzle of size $n \times k$ in the loose model can be solved by a fixed parameter tractable algorithm with parameter k.*

Proof. We first consider the one-sided plates. The basic idea is similar to the algorithm for the fit model: the algorithm checks all $k!2^k$ permutations. Now we fix a permutation. We have n possible places for n plates. Thus, we construct a bipartite graph $G = (X, Y, E)$ as follows. Let $X = \{x_1, x_2, \dots, x_n\}$ be the set of n plates of size $1 \times (k + 1)$, and $Y = \{y_1, y_2, \dots, y_n\}$ be the places produced by k plates of size $1 \times (n + 1)$. E consists of an edge $\{x_i, y_j\}$ if and only if the plate x_i can be assembled to the place y_j. The construction of the graph G

takes $O(kn^2)$ time. Then, it is easy to see that a solution of the lattice puzzle corresponds to a perfect matching on G. It is known that the perfect matching problem on a bipartite graph can be solved in polynomial time $p(|X|+|Y|) = O(\min\{\sqrt{|X|+|Y|}|E|, (|X|+|Y|)^{\omega}\})$, where $\omega < 2.373$ is the matrix multiplication exponent.

Therefore, the lattice puzzle can be solved in $O(k!2^k(kn^2 + p(n+k)))$ time. When the plates are two-sided, in the same way, we can solve it in $O(k!4^k(kn^2 + p(n+k)))$ time. ◀

6 Concluding remarks

In this paper, we propose a general framework of simple lattice puzzles. Using this framework, we can characterize some representative computational complexity classes. Especially, we can characterize the problems in NP-complete and GI-complete. As far as the authors know, there is no such a simple framework.

Although we show several results, we still have many unsolved problems. Especially, computational complexity of the simplest problem on $2n$ plates of size $1 \times (n+1)$ in the one-sided fit model is open. By Theorems 2 and 3, it seems that this problem exists between the NP-complete problem and the GI-complete problem.

We also mention that we focus on the problems that ask if a given set of plates has a feasible state or not in this paper. That is, we do not ask if the feasible state can be assembled even if each plate is rigid. Like the sliding block puzzles, allowing the "movement of pieces", the assembling puzzle of rigid plates can be PSPACE-complete. Is there a variant of the lattice puzzle with movement which gives us a characterization of PSPACE-completeness?

References

1. Hisayoshi Akiyama. *Cube Puzzle Book (in Japanese)*. Shinkigensha, 2004.
2. E. R. Berlekamp, J. H. Conway, and R. K. Guy. *Winning Ways for Your Mathematical Plays*, volume 1–4. A K Peters Ltd., 2001–2003.
3. M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.
4. M.R. Garey and D.S. Johnson. *Computers and Intractability – A Guide to the Theory of NP-Completeness*. Freeman, 1979.
5. R. A. Hearn and E. D. Demaine. *Games, Puzzles, and Computation*. A K Peters Ltd., 2009.
6. Takumi Kasai, Akeo Adachi, and Shigeki Iwata. Classes of pebble games and complete problems. *SIAM Journal on Computing*, 1979.
7. R. Uehara, S. Toda, and T. Nagoya. Graph Isomorphism Completeness for Chordal Bipartite Graphs and Strongly Chordal Graphs. *Discrete Applied Mathematics*, 145(3):479–482, 2004. doi:10.1016/j.dam.2004.06.008.