# PAPER NP-Hard and k-EXPSPACE-Hard Cast Puzzles\*

Chuzo IWAMOTO<sup>†a)</sup>, Member, Kento SASAKI<sup>†</sup>, Kenji NISHIO<sup>††</sup>, Nonmembers, and Kenichi MORITA<sup>†</sup>, Member

SUMMARY A disentanglement puzzle consists of mechanically interlinked pieces, and the puzzle is solved by disentangling one piece from another set of pieces. A cast puzzle is a type of disentanglement puzzle, where each piece is a zinc die-casting alloy. In this paper, we consider the generalized cast puzzle problem whose input is the layout of a finite number of pieces (polyhedrons) in the 3-dimensional Euclidean space. For every integer  $k \ge 0$ , we present a polynomial-time transformation from an arbitrary k-exponential-space Turing machine M and its input x to a cast puzzle  $c_1$  of size k-exponential in |x| such that M accepts x if and only if  $c_1$ is solvable. Here, the layout of  $c_1$  is encoded as a string of length polynomial (even if  $c_1$  has size k-exponential). Therefore, the cast puzzle problem of size k-exponential is k-EXPSPACE-hard for every integer  $k \ge 0$ . We also present a polynomial-time transformation from an arbitrary instance fof the SAT problem to a cast puzzle  $c_2$  such that f is satisfiable if and only if c<sub>2</sub> is solvable.

key words: computational complexity, NP-hard, k-EXPSPACE hard, cast puzzle

## 1. Introduction

Disentanglement puzzles are one of the most fundamental and popular playthings. They consist of mechanically interlinked pieces, and a puzzle is solved by disentangling one piece from another set of pieces. Disentanglement puzzles are classified into two categories, wire puzzles and cast puzzles.

A wire puzzle consists of two or more entangled stiff wires. Wires may or may not be closed loops, and they have complex shapes. Normally, wire puzzles are solved by disentangling one piece from another set of pieces without cutting or bending the wires. Sometimes, one of the pieces is a looped string.

On the other hand, pieces of cast puzzles are zinc diecasting alloys. Therefore, people who take them in hand feel their solidness and heaviness. See the official site [1] for the cast puzzles, produced by Hanayama Co.,Ltd. In this site, cast puzzles are classified into six levels (i.e., Easiest, Easy, Medium, Fairly hard, Hard, and Very hard). However, from the point of view of a computer scientist, they should be classified according to computational complexities (i.e., P,

a) E-mail: chuzo@hiroshima-u.ac.jp

DOI: 10.1587/transinf.E93.D.2995

NP, PSPACE, EXPTIME, 2-EXPSPACE, and so on).

We will define a cast puzzle as a set of simple polyhedrons. We consider the generalized cast puzzle problem whose input is the layout of a finite number of pieces (polyhedrons) in the 3-dimensional Euclidean space. Since all cast puzzles in this paper are constructed by polynomialtime transformations, the layout of a cast puzzle is encoded as a string of length polynomial (even if the cast puzzle has size *k*-exponential).

In this paper, for every integer  $k \ge 0$ , we present a polynomial-time transformation from an arbitrary *k*exponential-space Turing machine *M* and its input *x* to a cast puzzle  $c_1$  of size *k*-exponential in |x| such that *M* accepts *x* if and only if  $c_1$  is solvable. Therefore, the cast puzzle problem of size *k*-exponential is *k*-EXPSPACE-hard for every integer  $k \ge 0$ . We also present a polynomial-time transformation from an arbitrary instance *f* of the SAT problem to a cast puzzle  $c_2$  such that *f* is satisfiable if and only if  $c_2$  is solvable.

Flake and Baum proved that some PSPACE-complete problem can be reduced to the Rush hour problem (a sliding block puzzle on a board) [6]. From this result, it is not difficult to construct a PSPACE-hard cast puzzle by making sure that the pieces cannot use the third direction. Note that the *k*-EXPSPACE-hardness for a set of cast puzzles implies the NP-hardness and PSPACE-hardness for the *same* set of cast puzzles. Our NP-hard cast puzzle, transformed from the SAT problem, can be solved *by hand* in n + 1 steps with *n* guesses, where *n* is the number of variables of the SATinstance. On the other hand, a PSPACE-hard cast puzzle constructed by the idea of [6] cannot be solved in polynomial steps with polynomial guesses unless NP = PSPACE.

There has been a huge amount of literature on the computational complexities of games and puzzles. For example, Tetris [4], Solitaire [12], Minesweeper [11],  $(n \times n)$ -extension of the 15-puzzle [16], the Slither Link Puzzle [19], and Sokoban (a transport puzzle in a maze) [5], Puyo puyo (also known as Puyo pop) [14], and Hashiwokakero [3] are known to be NP-hard. Uehara proved that deciding whether a given pop-up book can be opened (or closed) is NP-hard [18]. As for higher complexity classes, Othello [10] is known to be PSPACE-hard; Chess [7], Shogi (Japanese chess) [2], and Go [17] are EXPTIME-hard. More information on games and puzzles can be found in [9].

In Sect. 2, we introduce the cast puzzle problem, where each piece is defined as a simple polyhedron. The main re-

Manuscript received July 8, 2010.

<sup>&</sup>lt;sup>†</sup>The authors are with the Graduate School of Engineering, Hiroshima University, Higashihiroshima-shi, 739–8527 Japan.

<sup>&</sup>lt;sup>††</sup>The author is with Sharp Corporation, Osaka-shi, 545–8522 Japan.

<sup>\*</sup>A preliminary version of the proof of Theorem 1 appeared in Proc. ISAAC 2009 (LNCS5878). This research was supported in part by Scientific Research Grant, Ministry of Japan.

sults are also given in that section. In Sects. 3 and 4, we will show transformations from Turing machines and SAT-instances to cast puzzles, respectively.

## 2. Definitions and Main Results

In our model, all pieces are defined as simple polyhedrons in the 3-dimensional Euclidean space  $E^3$ . The definitions of polygons and polyhedrons are mostly from [15].

In  $E^2$ , a *polygon* is defined by a finite set of segments such that every segment extreme is shared by exactly two edges and no subset of edges has the same property. The segments are the *edges* and their extremes are the *vertices* of the polygon.

In  $E^3$ , a *polyhedron* is defined by a set of plane polygons such that every edge of a polygon is shared by exactly one other polygon (adjacent polygon) and no subset of polygons has the same property. The vertices and the edges of the polygons are the *vertices* and the *edges* of the polyhedron, and the polygons are the *facet* of the polyhedron.

A polyhedron is said to be *simple* if there is no pair of non-adjacent facets sharing a point. A simple polyhedron partitions the space into two disjoint domains, the *interior* (bounded) and the *exterior* (unbounded). In this paper, the term polyhedron is used to denote the union of the boundary and of the interior.

A *cast puzzle* is a finite set of simple polyhedrons, called *pieces*, embedded in  $E^3$ . One of the pieces is called the *target piece*. The input of the *cast puzzle problem* is the layout of a finite number of pieces. A cast puzzle is said to be *solvable* if the target piece can be disentangled from another set of pieces without deforming, whittling, or breaking the pieces. In this paper, for any piece p, no vertex of any other piece is in the interior of p. A vertex of a piece may touch the surface of another piece. The surface of any piece is frictionless.

In this paper, we assume that each vertex of any piece of a cast puzzle has integral coordinates. Hence, a layout of polyhedrons can be represented as a string as follows. Suppose  $P_0$  is a polyhedron such that one of  $P_0$ 's vertices is on the origin of coordinates. Polyhedron  $P_0$  is represented by a set of  $P_0$ 's faces  $f_0, f_1, \ldots, f_{m-1}$ , where one of  $f_0$ 's vertices is on the origin of coordinates. Each face  $f_i$  is represented by a sequence of  $f_i$ 's vertices. Let  $\langle P_0, (x_1, y_1, z_1) \rangle$ denote the polyhedron embedded in  $E^3$ , obtained by *translating*  $P_0$  from (x, y, z) to  $(x', y', z') = (x, y, z) + (x_1, y_1, z_1)$ (which describes a transformation where each point is subject to a fixed displacement  $(x_1, y_1, z_1)$ ).

For example, let  $P'_0$  be a polyhedron fitted inside a unit cube. Set  $S = \{\langle P'_0, (2i, 0, 0) \rangle | 0 \le i \le h(n') - 1\}$  represents a sequence of h(n') polyhedrons placed at regular intervals, where n' is an integer, and h is a function of n'. Note that such well-regulated polyhedrons can be encoded as a string code(S) over  $\{0, 1\}$  of length  $O(\log n')$  when polyhedron  $P'_0$  and function h can be encoded as strings of length constant.

Let *n* be the length of the string representing the layout

of a cast puzzle. The cast puzzle is said to have *size* s(n) if the convex hull of all pieces is fitted inside a cuboid of size  $s_1(n) \times s_2(n) \times s_3(n)$  such that  $s_1(n) + s_2(n) + s_3(n) \le s(n)$ .

Let g(0, p(n)) = p(n) and  $g(k, p(n)) = 2^{g(k-1,p(n))}$  for every  $k \ge 1$ , where p(n) is an arbitrary polynomial function. A TM is said to be *k*-exponential-space bounded if, for every accepted input x of length n, M halts with an accepting state in g(k, p(n)) space.

The cast puzzle problem of size *k*-exponential is said to be *k*-EXPSPACE-hard if there is a polynomial-time transformation from an arbitrary *k*-exponential-space TM *M* and its input *x* to a cast puzzle  $c_1$  of size *k*-exponential in |x|such that *M* accepts *x* if and only if  $c_1$  is solvable. Since any cast puzzle in this paper is constructed by a polynomial-time transformation, the layout is encoded as a string of length polynomial (even if the cast puzzle has size *k*-exponential).

**Theorem 1:** For every integer  $k \ge 0$ , there is a polynomialtime transformation from an arbitrary *k*-exponential-space TM *M* and its input *x* to a cast puzzle  $c_1$  of size *k*exponential in |x| such that *M* accepts *x* if and only if  $c_1$ is solvable.

**Corollary 1:** For every integer  $k \ge 0$ , the cast puzzle problem of size *k*-exponential is *k*-EXPSPACE-hard.

**Theorem 2:** There is a polynomial-time transformation from an arbitrary instance f of the SAT problem to a cast puzzle  $c_2$  such that f is satisfiable if and only if  $c_2$  is solvable.

The proofs of Theorems 1 and 2 are given in Sects. 3 and 4, respectively.

Flake and Baum proved that some PSPACE-complete problem can be reduced to the Rush hour problem (a sliding block puzzle on a board) [6]. From this result, it is not difficult to construct a PSPACE-hard cast puzzle by making sure that the pieces cannot use the third direction. On the other hand, from Theorem 2, we can construct an NP-hard cast puzzle, which is different from their PSPACE-hard cast puzzle in the following sense.

Suppose that we have a polynomial amount of zinc and two layouts of NP-hard and PSPACE-hard cast puzzles. Our NP-hard cast puzzle can be solved by hand in n+1 steps with n guesses, although we do not prove the NP-completeness for our puzzles. On the other hand, their PSPACE-hard cast puzzle cannot be solved in polynomial steps with polynomial guesses unless NP = PSPACE. Here, a *one-step move* of a piece of NP-hard and PSPACE-hard cast puzzles is defined as a continuous translation from (x, y, z) to  $(x', y', z') = (x, y, z) + (x_1, y_1, z_1)$  such that no vertex of the piece is the interior of any other piece during the translation and  $x_1, y_1, z_1$  are integers.

#### 3. Transformation from TMs to Cast Puzzles

In this section, we will prove Theorem 1. We show a polynomial-time transformation from an arbitrary kexponential-space TM and its input to the layout of a cast



Fig. 1 (a) A target piece. (b) Lateral view. (c) Top view.



**Fig.2** (a) A board with a cross-shaped hole. (b) A cross-section view between C and D.

puzzle such that the TM accepts its input if and only if the cast puzzle is solvable.

The definition of a TM is mostly from [13]. A one-tape two-symbol TM is a system defined by  $M = (Q, \Sigma, q_0, q_{m-1}, R)$ , where  $Q = \{q_0, q_1, \dots, q_{m-1}\}$  is a finite set of states,  $\Sigma = \{0, 1\}$  is a set of tape symbols,  $q_0$  (resp.  $q_{m-1}$ ) is the unique initial (resp. final) state, and *R* is a set of transition rules, where  $R \subseteq ((Q - \{q_{m-1}\}) \times \Sigma \times \Sigma \times Q) \cup ((Q - \{q_{m-1}\}) \times \{/\} \times \{-1, +1\} \times Q)$ .

Each transition rule in *R* is of the form  $[q_i, a, a', q_{i'}]$  or  $[q_i, /, d, q_{i'}]$ , where  $q_i \in Q - \{q_{m-1}\}, q_{i'} \in Q$ ;  $a, a' \in \Sigma$ ; and  $d \in \{-1, +1\}$ . Rule  $[q_i, a, a', q_{i'}]$  means that if *M* reads symbol *a* in state  $q_i$ , then *M* writes *a'* and enters state  $q_{i'}$ . Rule  $[q_i, /, d, q_{i'}]$  means that if *M* is in state  $q_i$ , then *M* moves the head to the right (resp. left) when d = +1 (resp. d = -1) and enters state  $q_{i'}$ .

Let  $\alpha_1 = [q_i, a, a', q_{i'}]$  and  $\alpha_2 = [q_j, b, b', q_{j'}]$  be two transition rules in *R*. We say that  $\alpha_1$  and  $\alpha_2$  overlap in domain if  $(q_i = q_j \text{ and } a = b)$  or  $(q_i = q_j \text{ and } (a = / \text{ or } b = /))$ . A rule  $\alpha$  is said to be *deterministic* in *R* if there is no other rule in *R* with which  $\alpha$  overlaps in domain. A TM *M* is called deterministic if every rule in *R* is deterministic. In the rest of this paper, all TMs are one-tape two-symbol deterministic TMs. A TM defined in [13] is a reversible TM. Our TM is a deterministic TM, but it does not have to be a reversible TM. The reason why we use the definition of [13] is to separate transition rules into those of the forms  $[q_i, a, a', q_{i'}]$  and  $[q_i, /, d, q_{i'}]$ .

We first construct the target piece (see Fig. 1). It is composed of a square pole of size  $1 \times 1 \times (l + 2)$ , two rectangular poles of size  $1 \times 2 \times l$ , and two square poles of size  $1 \times 1 \times l$ . The length *l* will be fixed later. The four poles of length *l* are welded to the pole of length l + 2 so that they form a single polyhedron.

Consider a 3-thick board (see Fig. 2). The board has



**Fig. 3** (a) A board having four corridors, three octagonal spaces, and one quadrilateral space. (b) A cross-section view between D and E. (c) Diagrammatic sketch of (a). (d) Unimportant octagonal spaces are sometimes omitted.

a cross-shaped hole. There is a groove on the inside wall of the hole (see Fig. 2 (b)) so that the target piece cannot be taken out from the board. This cross-shaped hole is composed of horizontal and vertical *corridors*. In the horizontal corridor, the target piece can move to the right, and it can go back to the original position. The vertical corridor connects positions A and B. It should be noted that the target piece cannot move to position A or B from the current position. (Strictly speaking, this is not a cross-shaped *hole* because there is a narrow gap which connects the hole to the exterior so that the board is a polyhedron.)

Consider a board shown in Fig. 3 (a), which has four corridors, three octagonal space, and one quadrilateral space. Suppose that the target piece is in the right-upper octagonal space. The target piece can move to position A, take a 90-degree turn using the octagonal space, and the target piece can reach position C. At this position, we can take out the target piece from the board. In the following, such octagonal spaces and corridors are represented as nodes and arcs (see Fig. 3 (c)). A quadrilateral space (position C) is represented as a double circle. (Unimportant nodes are sometimes omitted as shown in Fig. 3 (d).) On this board, there is a *bridge* so that the two pieces form a single polyhedron (see Fig. 3 (b); octagonal space B and two corridors connected to B are for this explanation only).

We illustrate the transformation from a *k*-exponentialspace TM *M* to a cast puzzle. Since *M* is *k*-exponentialspace bounded, there is a polynomial p(n) such that *M* uses at most g(k, p(n)) cells on its tape, where g(0, p(n)) = p(n)and  $g(k, p(n)) = 2^{g(k-1, p(n))}$  for every integer  $k \ge 1$ . Each tape cell  $c_i$ ,  $1 \le i \le g(k, p(n))$ , is simulated by a *block* shown in Figs. 4 (a) and 4 (b). The size of each block de-



**Fig.4** (a) A block simulating a cell  $c_i$  of M, from a top view. (b) The same block from a lateral view. (c) Blocks are pierced by two L-shaped square poles, which are welded to another square pole.



**Fig. 5** (a) A block has a hole  $h(q_i, a)$  for every pair of state  $q_i \in Q$  and symbol  $a \in \Sigma$ . (b) Solid (resp. dotted) arcs correspond to transition rules of the form  $[q_i, a, a', q_{i'}]$  (resp.  $[q_i, /, d, q_{i'}]$ ). L-shaped square poles are omitted in this figure.

pends only on the number of M's states. All blocks are arranged in a row, and they are pierced by two L-shaped square poles, which are welded to another square pole (see Fig. 4 (c)). These square poles form a single polyhedron. (A groove in Fig. 4 (b) is explained later using Fig. 6.)

Let  $Q = \{q_0, q_1, \dots, q_{m-1}\}$  be the set of *M*'s states, where  $q_0$  and  $q_{m-1}$  are unique initial and final states, respectively. Recall that tape symbols are 0 and 1 only. Each block has 2m holes, say,  $h(q_0, 0), h(q_0, 1); h(q_1, 0), h(q_1, 1);$  $\dots; h(q_{m-1}, 0), h(q_{m-1}, 1)$  (see Fig. 5 (a) for a four-state TM). For every transition rule of the form  $[q_i, a, a', q_{i'}], a, a' \in \Sigma$ , we add an arc from node  $h(q_i, a)$  to  $h(q_{i'}, a')$  (see solid arcs in Fig. 5 (b)). Furthermore, for every transition rule of the form  $[q_i, /, d, q_{i'}], d \in \{-1, +1\}$ , we add  $2 \times 2$  dotted arcs from nodes  $h(q_i, 0), h(q_i, 1)$  in the current block to  $h(q_{i'}, 0), h(q_{i'}, 1)$  in the right (resp. left) adjacent block if d = +1 (resp. d = -1) (see dotted arcs in Fig. 5 (b)). All blocks are the same polyhedron except for the first and last blocks. The first (resp. last) block does not have arcs which is to or from the left (resp. right) adjacent block.

Figure 5 (b) is constructed according to a four-state deterministic TM  $M = (Q, \Sigma, q_0, q_3, R)$  such that  $Q = \{q_0, q_1, q_2, q_3\}, \Sigma = \{0, 1\}$ , and

$$R = \{ [q_0, /, +1, q_1], [q_1, 0, 1, q_0], [q_1, 1, 0, q_3], [q_2, 0, 1, q_3], [q_2, 1, 0, q_0] \}.$$
(1)

Figure 6 illustrates a cast puzzle simulating a configuration of TM *M* in Fig. 7 (a). It is composed of a sequence of g(k, p(n)) blocks (see also Fig. 5 (b)), two boards 1, 2 behind them, and g(k, p(n)) sets of octagonal cylinders  $S_1, S_2, \dots, S_{g(k,p(n))}$ . See Fig. 6 (b) for a lateral view of an octagonal cylinder. (Due to space limitation, Fig. 6 contains only three blocks, say, blocks 1, 2, 3, and three sets of octagonal cylinders  $S_1, S_2, S_3$ .)

Figure 6 (b) is a lateral view from the cross section between K' and K". Each of boards 1, 2 is welded to an Lshaped square pole (see Fig. 6 (b)). The other end of the L-shaped square pole is welded to square pole 1, which is further welded to square poles 2, 3 and a rectangular pole. It should be noted that boards 1, 2 are connected via two L-shaped square poles and a rectangular pole; if board 1 is moved to the right, then board 2 is also moved to the right simultaneously (see (6) in Fig. 6 (a)).

On block *i*, there is a set of octagonal cylinders  $S_i$ , where  $S_i$  contains *m* octagonal cylinders. Those *m* octagonal cylinders are connected by a rod so that they cover all holes in either  $\{h(q_i, 0) | 0 \le j \le m-1\}$  or  $\{h(q_i, 1) | 0 \le j \le m-1\}$ . The other end of the rod is connected to rectangular solid *i*, which has two (square) holes 1, 2 and one rectangular hole (see Fig. 6 (b)). Square poles 2, 3 pierce either holes 1, 2 or hole 2 and the rectangular hole. If square poles 2, 3 pierce holes 1, 2, then the set of octagonal cylinders close holes  $\{h(q_i, 1) \mid 0 \le j \le m - 1\}$  (and holes  $\{h(q_i, 0) \mid 0 \le j \le m - 1\}$ are opened; see  $S_2$  in Fig. 6 (a)), which implies the tape cell has symbol 0. Similarly, if square poles 2, 3 pierce hole 2 and the rectangular hole, then the tape cell has symbol 1 (see  $S_1, S_3$ ). There is a gap between the left square poles 2, 3 and the right square poles 2, 3. The gap distance is the same as the width of each rectangular solid. Therefore, there exists at most one set of octagonal cylinders which we can move vertically.

The rectangular pole always pieces all rectangular solids. We dig a ditch on every block (see Fig. 6(c)) and we fit the corresponding set of octagonal cylinders into the ditch so that they cannot move in the horizontal direction. On each board, there is a monorail which is fitted in the groove of each block so that the block does not move in the vertical direction.

Since the target piece has height 5 (see Fig. 1 (b)), the top and bottom of the target piece protrude outside the block (see Fig. 2 (b)). Therefore, the target piece must always be between boards 1 and 2. If the target piece reaches one of the "accepting" holes  $h(q_{m-1}, 0), h(q_{m-1}, 1)$ , then we can take it out from the block.

Consider a configuration of the TM *M* at step *t* (see Fig. 7 (a)). In Fig. 6 (a), the target piece is in the hole  $h(q_1, 0)$  in the second block, and holes  $\{h(q_j, 0) | 0 \le j \le m-1\}$  (resp.  $\{h(q_j, 1) | 0 \le j \le m-1\}$ ) are opened in the second block (resp. in the first and third blocks). From Eq. (1), there is a solid arc from hole  $h(q_1, 0)$  to hole  $h(q_0, 1)$  (see Figs. 6 (a)



**Fig. 6** (a) A cast puzzle simulating the configuration of a TM in Fig. 7 (a). (b) A lateral view from the cross section between K' and K''. (c) A lateral view of a block and an octagonal cylinder from point C of (b).



**Fig.** 7 Configurations of a TWI at steps i, i + 1, and i + 1

and 8 (a)), and there is a pair of dotted arcs from  $h(q_0, 1)$  to  $h(q_1, 0)$  and  $h(q_1, 1)$  in the right adjacent block, where the hole  $h(q_1, 0)$  is closed. Figure 8 (b) is the detailed drawing of Fig. 8 (a).

(1) Suppose that the target piece is in the hole  $h(q_1, 0)$ . (2) We can carry the target piece forward until it reaches position B (see Fig. 8 (b)). At this position, we can take a 45-degree turn. (3) During the evacuation of the target piece, we can move the set of octagonal cylinders  $S_2$  upward. (In Fig. 8 (b), octagonal cylinder H is just moving upward.) Since rectangular solid 2 is in the gap of square poles 2, 3 (see Fig. 6 (a)), we can freely move  $S_2$  vertically (as far as the rectangular pole in the rectangular hole permits). (4) After the movement of octagonal cylinders, the hole  $h(q_0, 1)$  at position C is opened, and  $h(q_1, 0)$  at position A is closed. We can carry the target piece to position C.



**Fig. 8** (a) Solid arc from  $h(q_1, 0)$  to  $h(q_0, 1)$ , and dotted arcs from  $h(q_0, 1)$  to  $h(q_1, 0)$  and  $h(q_1, 1)$ . (b) Detailed drawing of (a).

At this time, the edges of boards 1, 2 are on borderlines J,J', respectively (see Figs. 6 (a) and 8 (b)).

Currently, the target piece is in position C and the edges of boards 1 and 2 are on borderlines J and J', respectively. (5) We can carry the target piece to position D by moving board 2 to the right simultaneously. (6) Square poles 2, 3 are connected to boards 1, 2; thus the square hole 2 and the rectangular hole of rectangular solid 2 are pierced by square poles 2 and 3, respectively (see Fig. 6(a)). (7) Then we take a 90-degree turn at D, a downward movement from D to F, and a 90-degree turn at F. Again, we carry the target piece to position G by moving board 2 to the right. Finally, we move board 2 to the right so that octagonal cylinders  $S_3$ are placed in the middle between boards 1 and 2, by which square poles 2, 3 are pulled out of rectangular solid 3. Now the set of octagonal cylinders  $S_3$  can move vertically, and the target piece can also move according to the solid arc in the third block.

The target piece cannot reach hole  $h(q_1, 0)$  via dotted arc (5') in Fig. 8 (b) because of the following reason. In general, the set of octagonal cylinders  $S_i$  can move vertically if and only if the distance between I and J is the same as the distance between I' and J' (see Fig. 6 (a)).

Now we fix the length l of the target piece so that l is strictly longer than the distance between I and J. For such an l, the target piece gets stuck between the octagonal cylinder on  $h(q_1, 0)$  and board 1 (see position G' in Fig. 8 (b)).

Initially, the first *n* sets of octagonal cylinders  $S_1, S_2, \ldots, S_n$  are placed according to the input string  $x \in \{0, 1\}^n$ , and the remaining  $S_{n+1}, S_{n+2}, \ldots$  are placed so that holes  $h(q_j, 0)$  are opened on every block. The target piece is initially in the hole  $h(q_0, 0)$  or  $h(q_0, 1)$  in the first block. The layout of such polyhedrons can be represented by a string of length polynomial in *n*.

Since TM M is deterministic, the accepting computation is a sequence of configurations in which the last configuration is accepting. Such a sequence belongs to a directed *accepting tree* such that (i) every node is a configuration, and (ii) the root node is the unique accepting configuration in the tree. Note that there is a path from every node in the tree to the accepting node (i.e., root node). Therefore, Maccepts input x if and only if the initial configuration of Mbelongs to an accepting tree. Moving the target piece in our cast puzzle corresponds to walking round nodes in the tree. The root node of this tree is an accepting configuration if and only if the target piece can be taken out of a block. By this construction, the TM M accepts x if and only if the cast puzzle is solvable. This completes the proof of Theorem 1.

## 4. Transformation from SAT-Instances to Cast Puzzles

In this section, we will prove Theorem 2. We show a polynomial-time transformation from an arbitrary instance f of the 3-SAT problem (3-SAT) [8] to a cast puzzle such that f is satisfiable if and only if the cast puzzle is solvable.

Let  $U = \{x_1, x_2, ..., x_n\}$  be a set of Boolean variables.



**Fig.9** (a) A polyhedron made by one square pole and four flanges. (b) The same polyhedron from a lateral view.



**Fig. 10** (a) A board with large and small holes connected by a channel. (b) A flange is smaller than the large hole, but it is larger than the small hole. The pole can move back and forth between large and small holes through the channel. (c) The lateral view of the board when the pole is in the large hole.

Boolean variables take on values 0 (false) and 1 (true). If x is a variable in U, then x and  $\overline{x}$  are *literals* over U. The value of  $\overline{x}$  is 1 (true) if and only if x is 0 (false). A *clause* over U is a set of literals over U, such as  $\{\overline{x_1}, x_3, x_4\}$ . It represents the disjunction of those literals and is *satisfied* by a truth assignment if and only if at least one of its members is true under that assignment.

An instance of 3-SAT is a collection  $f = \{c_1, c_2, \ldots, c_m\}$  of clauses over U such that  $|c_i| = 3$  for  $1 \le i \le m$ . The 3-SAT problem asks whether there exists some truth assignment for U that simultaneously satisfies all the clauses in f. For example,  $U = \{x_1, x_2, \ldots, x_5\}$ ,  $f = \{c_1, c_2, c_3, c_4\}$ , and  $c_1 = \{\overline{x_1}, x_3, x_4\}$ ,  $c_2 = \{x_1, x_2, x_4\}$ ,  $c_3 = \{\overline{x_2}, x_4, \overline{x_5}\}$ ,  $c_4 = \{\overline{x_3}, \overline{x_4}, \overline{x_5}\}$  provide an instance of 3-SAT. For this instance, the answer is "yes", since there is a truth assignment  $(x_1, x_2, x_3, x_4, x_5) = (0, 1, 0, 1, 0)$  satisfying all clauses.

In order to transform the instance of 3-SAT, we define several pieces, such as a *square pole*, *flanges*, and *boards* (see Figs. 9 and 10). Flanges are horseshoe-shaped pieces (see Fig. 9 (a)), which straddle the square pole at regular intervals. Flanges are welded to the pole; namely, a pole with flanges at regular intervals form a single polyhedron.

A board has large and small *holes*, which are connected by *a channel* (see Fig. 10 (a)). A flange is smaller than the large hole, but it is larger than the small hole (Fig. 10 (b)). The pole can move back and forth between large and small holes through the channel. (Strictly speaking, there is a narrow gap which connects the small hole to the exterior so that the board is a polyhedron. Such gaps can also be found in Figs. 12, 15, and 16.)

Suppose that a square pole with two flanges is in the large hole, where the board is between two flanges (Fig. 10(c)). In this case, we can pull the pole out of the



**Fig. 11** (a) Relationship among eight assignments for a clause on three literals. (b) We assign a small box to node 000 and large boxes to the remaining nodes.



**Fig. 12** Octagonal board with control rods simulating a clause  $c_1 = \{\overline{x_1}, x_3, x_4\}$ .

board, since the large hole is larger than flanges. On the other hand, if the pole is in the small hole, we cannot pull it out of the board.

All the possible assignments for three literals are  $(0, 0, 0), (0, 0, 1), \ldots, (1, 1, 1)$ . We represent the relationship among those eight assignments by a four-level graph shown in Fig. 11 (a). For example, node 000 in the figure corresponds to assignment (0, 0, 0).

We assign a small box to node 000, and large boxes to the remaining nodes in the graph (see Fig. 11 (b)). Now the board corresponding to each clause is constructed as described in Fig. 12. The board for clause  $c_j$  is called *octagonal board*  $c_j$ . Note that all octagonal boards have the same shape for all clauses  $c_1, c_2, \ldots, c_m$ .

The octagonal board has seven large holes and one small hole at the positions where the corresponding nodes in Fig. 11 (b) are in large boxes and a small box, respectively. In Fig. 12, the small hole is called *hole* 000, and large holes are called *holes* 001, 010, ..., 111, which correspond to nodes 001, 010, ..., 111 in Fig. 11 (b), respectively. Two holes in Fig. 12 are connected by a channel if the corresponding nodes are connected by an edge in Fig. 11 (b).

We construct a square pole with m + 1 flanges (see Fig. 9). We get the pole through a hole of every octagonal board such that flanges and octagonal boards appear alternately. Suppose that the first clause is  $c_1 = \{\overline{x_1}, x_3, x_4\}$ . This clause is unsatisfied if and only if  $(x_1, x_3, x_4) = (1, 0, 0)$ . In this sense, we get the pole through hole 100. The posi-



Fig. 13 (a) The lateral view of an octagonal board with bridges. Five bridges appear as one, since they are overlapping in space from the lateral view. (b) Square pole with m branches. Flanges are at the end of branches.

tion of the pole corresponds to the assignment for variables  $(x_1, x_3, x_4) = (0, 0, 0)$  (i.e., for literals  $(\overline{x_1}, x_3, x_4) = (1, 0, 0)$ ).

Furthermore, we fix three *control rods* on the lateral side of every octagonal board. The gradient and length of each rod are the same as those of the corresponding channel (see Fig. 12). If we push or pull a rod in the direction of the arrow, the square pole moves from a hole to another hole through a channel. One can see that pulling (resp. pushing) a rod implies changing the assignment of a literal from 0 to 1 (from 1 to 0). Note that the assignment for literals  $(\overline{x_1}, x_3, x_4)$  is (0, 0, 0) if and only if the pole is in the small hole 000.

Here, it should be noted that the octagonal board of Fig. 12 is not a single piece but eight disjoint pieces. So, we connect the eight pieces with seven *bridges* so that they form a single polyhedron (see Fig. 13 (a)). Since bridges may disrupt the movement of the square pole, we change the shape of the pole. The new square pole has *m branches*, where each branch has a flange at the end (see Fig. 13 (b)). The detailed drawing of the end of each branch will be given in Fig. 21, where we will use the free-floating gadget (see also Fig. 17).

Now, we construct a connection between literals and clauses. For simplicity of exposition, suppose  $f = \{c_1, c_2, c_3, c_4\}$  and

$$c_1 = \{\overline{x_1}, x_3, x_4\}, \quad c_2 = \{x_1, x_2, x_4\}, \\ c_3 = \{\overline{x_2}, x_4, \overline{x_5}\}, \quad c_4 = \{\overline{x_3}, \overline{x_4}, \overline{x_5}\}.$$

Figure 14 illustrates control rods of octagonal boards  $c_2, c_3$ , and  $c_4$ .

Since literal  $x_1$  appears in clause  $c_2$ , the control rod for  $x_1$  is connected to the second octagonal board (see Fig. 13 (b)). Since literal  $x_4$  appears in clauses  $c_1, c_2$ , and



**Fig. 14** (a)  $c_2 = \{x_1, x_2, x_4\}$ , (b)  $c_3 = \{\overline{x_2}, x_4, \overline{x_5}\}$ , (c)  $c_4 = \{\overline{x_3}, \overline{x_4}, \overline{x_5}\}$ .



**Fig. 15** (a) 24 octagonal cylinders in a doughnut-shaped hollow tube. Control rods are in ducts. (b) Clockwise rotation.



Fig. 16 A hollow tube with a gap.

 $c_3$ , the control rod for  $x_4$  is connected to the first, second, and third octagonal boards. Connections for the remaining literals are similar. In general, a literal  $x_i$  (resp.  $\overline{x_i}$ ) appears in clause  $c_j$  if and only if the control rod for  $x_i$  (resp.  $\overline{x_i}$ ) is connected to the *j*th octagonal board, where  $1 \le i \le n$  and  $1 \le j \le m$ .

The value of literal  $x_i$  is 1 if and only if  $\overline{x_i}$  is 0; in order to simulate this relation, we use the gadget shown in Fig. 15 (a). First of all, consider a hollow tube such that one of the four faces has a gap (see Fig. 16). We connect eight such tubes so that they form a doughnut shape (see Fig. 15 (a)). The doughnut-shaped tube is filled with 24 octagonal cylinders leaving no space between adjacent cylinders. (The value 24 has no special significance.) Two of the 24 octagonal cylinders in Fig. 15 (a) (see black cylinders) are connected to a pair of control rods in Fig. 13 (b). The gap of the hollow tube (Fig. 16) is the same as the width of rods.

If we rotate all octagonal cylinders two positions in a clockwise direction (see Fig. 15 (b)), the control rods for  $x_i$  and  $\overline{x_i}$  are pulled and pushed, respectively. One can see that both  $x_i$  and  $\overline{x_i}$  do not have the same value simultaneously. Initially, black cylinders are placed so that  $x_i = 0$  and  $\overline{x_i} = 1$  for every  $i \in \{1, 2, ..., n\}$ . Since control rods should always be pushed and pulled *vertically*, we cover them with *ducts* (Fig. 15 (a)).

Finally, all doughnut-shaped tubes are fixed on a *plat*form (see Fig. 17) so that the positional relationship among them may not be changed. Consider octagonal board  $c_1$  in



**Fig. 17** Octagonal board  $c_1$  can make "free-floating" moves in the directions of  $\overline{x_1}$  and  $x_3$ .

Fig. 17. (See also the control rod for  $x_4$  in Fig. 12, and octagonal board  $c_1$  connected to  $x_4$  in Fig. 13 (b).) In Fig. 17, octagonal board  $c_1$  is connected to a rectangular solid, which is in a hollow tube (labeled with  $x_3$ ). (Tubes have "stoppers" at both ends so that the solid do not fall off the tube.) This tube is further connected to another rectangular solid, which is in a hollow tube (labeled with  $\overline{x_1}$ ). This tube is connected to a black cylinder in the doughnut-shaped hollow tube. (Note that welded joints connect between a black cylinder and a hollow tube; a rectangular solid and a hollow tube; a rectangular solid and an octagonal board.) Those two hollow tubes are for the free-floating mechanism as follows.

It should be noted that octagonal board  $c_1$  is also connected with rods  $\overline{x_1}$  and  $x_3$  (see Fig. 12). If 24 cylinders move two positions in a clockwise direction, the square pole in Fig. 12 moves from hole 100 to the right-lower hole 101 through a channel. In this case, the two rectangular solids of  $\overline{x_1}$  and  $x_3$  in Fig. 17 do not move in the left-lower or vertical direction, if the pole moves along the channel from hole 100 to hole 101 (this "no turn" mechanism is explained in the following paragraphs using Figs. 18 through 21). Then, if rod  $\overline{x_1}$  in Fig. 12 is pushed, then the pole moves in the right-upper direction from hole 101 to hole 001. This movement is allowed by the free-floating mechanism in Fig. 17; the rectangular solid in tube  $\overline{x_1}$  is moved to the left-lower direction.

If we use a square pole (see Fig. 9), we can move it from the hole 100 to hole 010 (see Figs. 11 and 12) by taking a left turn at the intersection of two channels (see Fig. 18 (a)). In order to avoid such a movement, we use an



**Fig. 21** (a) A flange shown in Fig. 13 (b). (b) Detailed drawing of (a). A flange and three octagonal poles are connected to a free-floating gadget. This figure is from a worm's-eye view (except the branch).



Fig. 18 (a) Square pole with a flange. (b) Octagonal pole with two ditches.



 $\label{eq:Fig.19} Fig. 19 \qquad (a) \ Octagonal \ pole. \ (b) \ Ditches. \ (c) \ The \ lateral \ view \ of \ the \ octagonal \ pole \ with \ two \ ditches.$ 



Fig. 20 (a) Octagonal pole after 45-degree revolution. (b) The lateral view of (a). (c) Octagonal pole after (-45)-degree revolution. (d) The lateral view of (c).

octagonal pole (see Figs. 18 (b) and 19 (b)).

Consider an octagonal pole of which one of the rectangular faces is gray colored (see Fig. 19 (a)). We dig two ditches as pictured in Figs. 19 (b) and 19 (c). In Fig. 19 (c), the width d of a rectangular face is the same as the width of a channel. Gap distance s is the same as the thickness of the octagonal board of Fig. 12. This octagonal pole with two ditches cannot take the above left turn at the intersection of two channels (see Fig. 18 (b)).

We use two more octagonal poles with ditches, which are placed after 45-degree and (-45)-degree revolutions (see Fig. 20). We connect three poles in Figs. 20 (a), 20 (c), and 19 (b) so that they form a single octagonal pole, called the *new pole*. A flange and a free-floating gadget are fixed on the left and right ends of the new pole (see Fig. 21.) The new pole is smaller than the small hole. If the new pole is in a small or large hole of the octagonal board (see Fig. 12), we can choose one of the three octagonal poles by moving the rectangular solid in the tube in Fig. 21. There is a "stopper" to the left of the hole so that the flange is always between the stopper and the octagonal board, by which the new pole always pierces the octagonal board. The height h of any bridge is larger than the length of the new pole so that the bridge does not disrupt the horizontal movement of the new pole.

By the constructions, the pole with *m* branches in Fig. 13 (b) can be pulled out of octagonal boards if and only if the new pole is moved from the initial hole to a large hole in every octagonal board  $c_j \in \{c_1, c_2, \ldots, c_m\}$  simultaneously (if and only if there is a truth assignment for variables  $x_1, x_2, \ldots, x_n$  satisfying all clauses  $c_1, c_2, \ldots, c_m$ ). This completes the proof of Theorem 2.

#### 5. Conclusions

In this paper, we presented a polynomial-time transformation from a *k*-exponential-space TM *M* and its input *x* to a cast puzzle  $c_1$  of size *k*-exponential such that *M* accepts *x* if and only if  $c_1$  is solvable. As a corollary, the cast puzzle problem of size *k*-exponential is *k*-EXPSPACE-hard. We also presented a polynomial-time transformation from an instance *f* of the SAT problem to a cast puzzle  $c_2$  such that *f* is satisfiable *x* if and only if  $c_2$  is solvable.

#### Acknowledgement

We would like to thank anonymous referees for their constructive suggestions on the initial draft of this paper and pointing out an error in the proof of Theorem 2.

#### References

- [1] http://www.castpuzzle.net/english/index.html
- [2] H. Adachi, H. Kamekawa, and S. Iwata, "Shogi on n × n board is complete in exponential time," IEICE Trans. Inf. & Syst. (Japanese Edition), vol.J70-D, no.10, pp.1843–1852, Oct. 1987.

- [3] D. Andersson, "Hashiwokakero is NP-complete," Inf. Process. Lett., vol.109, pp.1145–1146, 2009.
- [4] E.D. Demaine, S. Hohenberger, and D. Lieben-Nowell, "Tetris is hard, even to approximate," Proc. 9th Ann. Int'l Conf. on Theoretical Aspects of Computer Science, LNCS 2697, pp.351–363, Springer, 2003.
- [5] D. Dor and U. Zwick, "Sokoban and other motion planning problems," Comput. Geom., Theory Appl., vol.13, no.4, pp.215–228, 1999.
- [6] G.W. Flake and E.B. Baum, "Rush hour is PSPACE-complete, or why you should generously tip parking lot attendants," Theor. Comput. Sci., vol.270, no.1/2, pp.895–911, 2002.
- [7] A.S. Fraenkel and D. Lichtenstein, "Computing a perfect strategy for *n* × *n* chess requires time exponential in *n*," J. Comb. Theory A, vol.31, no.2, pp.199–214, 1981.
- [8] M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman, New York, 1979.
- [9] R.A. Hearn and E.D. Demaine, Games, Puzzles, and Computation, A K Peters Ltd., 2009.
- [10] S. Iwata and T. Kasai, "The Othello game on an  $n \times n$  board is PSPACE-complete," Theor. Comput. Sci., vol.123, no.2, pp.329–340, 1994.
- [11] R. Kaye, "Minesweeper is NP-complete," Math. Intelligencer, vol.22, no.2, pp.9–15, 2000.
- [12] L. Longpré and P. McKenzie, "The complexity of solitaire," Theor. Comput. Sci., vol.410, pp.5252–5260, 2009.
- [13] K. Morita, A. Shirasaki, and Y. Gono, "A 1-tape 2-symbol reversible Turing machine," IEICE Trans., vol.E72, no.3, pp.223–228, March 1989.
- [14] H. Muta, "PUYOPUYO is NP-complete," IEICE Technical Report, COMP2005-14, 2005.
- [15] F.P. Preparata and M.I. Shamos, Computational Geometry: An Introduction, Springer Verlag, New York, 1985.
- [16] D. Ratner and M. Warmuth, "Finding a shortest solution for the  $(N \times N)$ -extension of the 15-puzzle is intractable," J. Symbolic Computation, vol.10, pp.111–137, 1990.
- [17] J.M. Robson, "The complexity of Go," Proc. 9th World Computer Congress on Information Processing, pp.413–417, 1983.
- [18] R. Uehara and S. Teramoto, "Computational complexity of a popup book," Proc. 18th Canadian Conf. on Computational Geometry, pp.3–6, 2006.
- [19] T. Yato, "On the NP-completeness of the slither link puzzle," IPSJ Signotes Algorithms, vol.74, pp.25–32, 2000.



Kento Sasaki received the B.Eng. degree from Hiroshima University in 2010. He is currently a student at the Graduate School of Engineering, Hiroshima University.



Kenji Nishio received the B.Eng. and M.Eng. degrees from Hiroshima University in 2007 and 2009, respectively. He is currently working in Sharp Corporation.



Kenichi Morita received the B.Eng., M.Eng., and Dr. Eng. degrees from Osaka University in 1971, 1973, and 1978, respectively. From 1974 to 1987, he was a research associate of the Faculty of Engineering Science, Osaka University. From 1987 to 1990 he was an associate professor, and from 1990 to 1993 a professor of the Faculty of Engineering, Yamagata University. Since 1993 he has been a professor of the Graduate School of engineering, Hiroshima University. He has been engaged in the

research of automata theory (especially cellular automata), reversible computing, formal language theory, array grammars, and logic systems for knowledge and language processing.





**Chuzo Iwamoto** received the B.Eng. degree from Yamaguchi University in 1990, and M.Eng. and Dr. Eng. degrees from Kyushu University in 1992 and 1995, respectively. From 1995 to 1997, he was a lecturer at Kyushu Institute of Design. Since 1997 he has been an associate professor in the Graduate School of Engineering, Hiroshima University. His present interests include computational complexity and automata theory.