Title	パイプパズルに関する研究
Author(s)	白山，卓夢
Citation	
Issue Date	2018-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/15180
Rights	
Description	Supervisor:上原　隆平，先端科学技術研究科，修士（情報科学）

Research on Computational Complexity of Pipe Puzzle
1610095 Takumu Shirayama

Research on computational complexity of games and puzzles has a long history. This topic has brought many contributions to not only characterization of computational complexity classes, but also development of techniques of algorithms.

Tiling puzzles form a large category in classic puzzles, and there are many variants according to their matching rules of tiling. One of the representative and well-known tiling puzzles is a jigsaw puzzle. In a jigsaw puzzle, a set of pieces is given. Each piece is almost square, and each edge has its own concave or convex curve. According to the curve, each edge has its matching pair. We have to put the pieces with adjacent matching pairs. The goal is to put all pieces into a rectangular shape, which is not given explicitly. In an ordinary jigsaw puzzle, each edge has its own unique matching pair. Therefore, it has a unique solution, and the resulting rectangular shape is uniquely built. However, in research on computational complexity, the jigsaw puzzle is generalized so that each edge can be matched to two or more edges in general, which drastically changes the computational complexity of this puzzle. As a variant of tiling puzzle, another category of puzzles is known as edge matching puzzles. They are as popular and classic as jigsaw puzzles, and many different kinds of edge matching puzzle are on the market. In fact, there was an edge matching puzzle such that a $2 million prize was offered for the first solver. In edge matching puzzles, each edge of the tile has a color or pattern. If the tile has a color on its edge, we will match the same colored edge and place tiles. If the tile has a pattern on the edge, tiles are placed with collecting the edges so that the pattern can be put together. The goal is to collect the edges with the same colors (or the pattern can be put together) and fit all the tiles in a rectangle of desired size.

In this paper, we investigate one new kind of edge matching puzzle. This puzzle is called *pipe puzzle*. We are given a set of cards that drawn pictures of pipes. Some joints of pipes appear at the centers of the cards, and pipes are cut at the edges of cards. When we place the cards, pipes should be matched at each edge so that pipes are seamlessly joined there. The goal of the pipe puzzle is different from the other previously investigated tiling puzzles. We are also given two points, which are placed before starting the puzzle. The goal of the pipe puzzle is to joint these two endpoints by all cards without leaking water. We investigate the computational complexity of this pipe puzzle.

First, we generalize pipe puzzles. Generalized pipe puzzles consist of a set of square cards, a board to place cards and two endpoints on the board. All of the cards are classified into pipe cards and blank cards. The pipe cards are the cards which are drawn pipes, and the blank cards are the cards which are not drawn any pipe. In this paper, we assume that pipes in pipe cards are unbranched pipes. Each pipe card has joints of pipe on its edges. Each joint has a type which means the radius of the pipe. We have to connect the same type of joints. The cards are put into a given board. The board has a rectangular shape and consists of cells. One endpoint is placed on the left edge and the other endpoint is placed on the right edge of the board.

For a given set of cards, we place each of them on a cell of the board one by one. When we place the next card, it has to satisfy one of the following two conditions: Two adjacent edges have joints of the same type when both of the edges are joints. Two adjacent edges are non-joint. We call the conditions placement rules. We call a placement valid placement if all cards in the board satisfy the placement rules. On a valid placement, we assume that two given endpoints on the board are joined by a sequence of the pipe cards. Then we call this sequence a path between two these endpoints. For a given instance of the pipe puzzle, when two endpoints are joined by a path, and all pipe cards appear exactly once on this path, this valid placement of the cards is called a solution of the pipe puzzle problem.

We proved that the generalized pipe puzzle problem is NP-complete. The generalized pipe puzzle problem is clearly in NP. Therefore, we prove that the generalized pipe puzzle problem is NP-hard. We use the reduction from the 3-partition problem to the generalized pipe puzzle problem. We construct a board, endpoints and cards from the given the 3-partition problem instance in such a way that a constructed pipe puzzle represents the original 3-partition problem instance.

Now we turn to the cases that the pipe puzzle problem can be solved in polynomial time. We investigate the pipe puzzle problem that has only one type of joint. We consider two cases; the first case is the pipe puzzle problem of a constant height, and the second case is the pipe puzzle problem of infinite height. (In the latter case, we assume that the number of pipe cards is finite, while the number of blank cards is infinite.) For the first case, we show a polynomial time algorithm for solving that. The algorithm is based on dynamic programming technique. That is, the algorithm does not store the whole information of placement of cards to reduce its running time and memory. It stores placement states, which only consists of the numbers of placed cards and the conditions of the edges on the boundary of these placed cards and empty area on the board. For the second case, we investigate the property of the solutions in an inductive way. As a result, we can check if the given set of cards has a solution or not in linear time without placement.