On A Generalization of “Eight Blocks to Madness” Puzzle*

Kazuya Haraguchi†

Abstract

We consider a puzzle such that a set of colored cubes is given as an instance. Each cube has unit length on each edge and its surface is colored so that what we call the Surface Color Condition is satisfied. Given a palette of six colors, the condition requires that each face should have exactly one color and all faces should have different colors from each other. The puzzle asks to compose a $2 \times 2 \times 2$ cube that satisfies the Surface Color Condition from eight suitable cubes in the instance. Note that cubes and solutions have 30 varieties respectively. In this paper, we give answers to three problems on the puzzle: **(i)** For every subset of the 30 solutions, is there an instance that has the subset exactly as its solution set? **(ii)** Create a maximum sized infeasible instance (i.e., one having no solution). **(iii)** Create a minimum sized universal instance (i.e., one having all 30 solutions). We solve the problems with the help of a computer search. We show that the answer to (i) is no. For (ii) and (iii), we show examples of the required instances, where their sizes are 23 and 12, respectively. The answer to (ii) solves one of the open problems that were raised in [E. Berkove et al., “An Analysis of the (Colored Cubes)^3 Puzzle,” *Discrete Mathematics*, **308** (2008) pp. 1033–1045].

keywords: Combinatorial puzzle, MacMahon’s colored cube puzzle, Eight Blocks to Madness, Computer-assisted proof

1 Introduction

Various kinds of mathematical puzzles have been invented, which provide us with not only recreation but also interesting research problems. Many types of puzzles such as pencil puzzles [2, 10, 16, 20, 31] and even video games [1, 9, 29] have been shown to be NP-hard, which seems to be a common property that attractive puzzles should have. Hearn and Demaine collected major complexity results in [17].

Puzzle instance creation is one of the possible future research directions in mathematical puzzles. Ordinary people do not have interest in computational complexity. They just like to play more and more addictive and challenging puzzle instances. Hence it is meaningful to exploit how to create puzzle instances that have desired properties. Nevertheless, fairly little literature deals with puzzle instance creation (e.g., Futoshiki [14] and BlockSum [15]). This is due to its difficulty. Creating a puzzle instance is harder than just solving it in general because creation process involves solving. To create an instance, we need not only to check whether a candidate instance is solvable but also to examine whether it has desired properties (e.g., solution uniqueness) or even to search the instance space for better candidates.

With these in mind, we explore how to create an instance for simple puzzles. In this paper, we take up a certain generalization of a classical puzzle called “Eight Blocks to Madness.” We present a collection of our research results on this puzzle. Specifically, we give answers to three problems named Existence, MaxInfeasible, and MinUniversal that are described below.

In the puzzle that we consider, we deal with colored cubes. Suppose that a palette of six colors is given. A puzzle solver is given a set of m colored cubes as an *instance*. Each cube has unit length on each edge and the surface is colored so that the following *Surface Color Condition* is satisfied.

*This is a preprint version of the paper of the same title that appears in *Discrete Mathematics*, **339** (2016), pp. 1400–1409. DOI information: 10.1016/j.disc.2015.12.014

†Department of Information and Management Science, Faculty of Commerce, Otaru University of Commerce, Midori 3-5-21, Otaru, Hokkaido, 0478501 Japan. E-mail: haraguchi@res.otaru-uc.ac.jp

Surface Color Condition: *Each face has exactly one color and all faces have different colors from face to face.*

Hence, 30 *varieties* are possible for colored cubes. The instance may contain multiple cubes for some varieties or may contain no cubes for other varieties. The puzzle asks to compose a $2 \times 2 \times 2$ cube by using eight suitable cubes in the instance so that the $2 \times 2 \times 2$ cube satisfies the Surface Color Condition. We call a $2 \times 2 \times 2$ cube whose surface is colored in that way a *solid*. Of course there are 30 varieties of solids. Once an instance is given, the solids that can be constructed are uniquely determined. Note that all the 30 varieties do not necessarily appear.

To create a puzzle instance, we fix the solution set at first and then search for an instance that has this solution set, as has been done in creation of pencil puzzle instances [14, 15]. Let us denote the set of all the 30 varieties by V_{all} and an arbitrary subset of V_{all} by V. An instance is *V-generatable* if a solid of any variety in V can be composed and no solid of any variety out of V can be composed anyhow. In our creation process, we first decide a subset V of V_{all} and then construct a V-generatable instance. We come up with the following problem.

Problem Existence Is there a V-generatable instance for all possible $V \subseteq V_{\text{all}}$?

We show that, unfortunately, the answer is no. This suggests that a puzzle creator should choose the solution set V appropriately so that a V-generatable instance exists.

Next, we observe two extreme cases: $V = \emptyset$ and $V = V_{\text{all}}$. Let us call an $\emptyset$-generatable instance *infeasible* and a V_{all}-generatable instance *universal*. It is easy to construct examples of these types of instances. For example, an instance containing seven or fewer cubes is infeasible. An instance that contains eight or greater cubes for all the 30 varieties is universal. It is interesting to explore a larger infeasible instance and a smaller universal instance since, intuitively, the more cubes and the more varieties an instance contains, the more varieties of solids are expected to be composable. The following problems arise.

Problem MaxInfeasible Create an infeasible instance of the maximum size.

Problem MinUniversal Create a universal instance of the minimum size.

We give answers to these problems by showing examples of a maximum infeasible instance and a minimum universal instance, where their sizes are 23 and 12, respectively. We emphasize that MaxInfeasible solves one of the open problems in [6].

To tackle these problems, we utilize the power of a computer search. Our approach can be regarded as "computer-assisted proof," which has been used to solve various significant problems in discrete mathematics; e.g., Four color theorem [3, 4, 5], Sudoku critical set [25]. We formulate the problems by *Constraint Programming* (*CP*) or by *Integer Programming* (*IP*). Once the models are established, we may be able to solve the problems by using state-of-the-art computational softwares.

The paper is organized as follows. We review previous work related to the puzzle in Section 2 and prepare terminologies and notations in Section 3. We present the answers to the three problems in Section 4, along with their CP or IP formulations. We conclude the paper in Section 5, describing open problems.

2 Related Work

The original version of "Eight Blocks to Madness" was invented by Irishman Eric Cross in 1960's and was issued by Austin Enterprises of Ohio [26]. The puzzle instance has exactly eight cubes that are of different varieties from each other. Thus a solver has only to arrange the eight given cubes into a solid. Kahan [19] showed that the solid variety composable from this original instance is unique. Sobczyk [27] studied a slight generalization of the original puzzle so that an instance can be a set of *any* eight cubes, repetition of the same variety cubes being allowed. He showed that at most six varieties of solids are composable from such an instance. Our puzzle is a generalization of these previous puzzles in the sense that an instance is a set of any m cubes.

Berkove et al. [6] already studied a further generalization. In their puzzle, which they call "(Colored Cubes)^3 Puzzle," a puzzle solver is given a set of m cubes as an instance and is asked to compose an $n \times n \times n$ solid, where $m \geq n^3$ is assumed. They prove that a solid is always composable when $n \geq 3$. The scenario of the proof is described as follows: they show that, when $n \geq 3$, an $n \times n \times n$ solid is composable iff the instance contains such a subset of eight cubes that can be arranged into a $2 \times 2 \times 2$ solid. They show that every instance of no less than $3^3 = 27$ cubes has such a subset. Such a subset is significant because, to compose an $n \times n \times n$ solid, it is most "difficult" to find eight cubes from the instance that are assigned to the corners of the solid; these cubes form a $2 \times 2 \times 2$ solid. We use the term "difficult" to mean that cubes that can be assigned to corners are most restricted since the three faces are exposed to the outside. On the other hand, it is comparatively easy to find cubes to be assigned to other parts of the solid. For example, we can assign any cube to the inside since no face is exposed to the outside. Thus, the $n = 2$ case plays a significant role in the analysis of the $n \geq 3$ case. This is why we concentrate on the $n = 2$ case.

Colored cube puzzles appear to have been introduced to recreational mathematics by P.A. MacMahon in 1920's [23, 24]. He invented the first puzzle that deals with colored cubes. In his puzzle, a puzzle solver is given a set of 29 cubes of different varieties except a certain variety, say v. The solver is asked to compose a $2 \times 2 \times 2$ solid of the variety v so that the *Domino Condition* is satisfied, i.e., in the inside of the solid, two faces touching each other should have the same color. The research history of this puzzle is summarized in M. Gardner's famous "Fractal Music, Hypercards and More..." [12]. The most outstanding way to solve the puzzle was developed by J.H. Conway. He arranged the 30 varieties of cubes in the non-diagonal entries of a 6×6 grid, as in Figure 1. When v is the variety in the row i and in the column j, we can compose the desired solid from eight cubes, one each of the varieties in row j and column i, not including the (j, i)-variety.

Our puzzle is different from MacMahon's in three points: in our puzzle, an instance is a multiset of m cubes, the solid variety to be constructed is not specified to the solver, and the Domino Condition is not imposed on the inside of the solid. Nevertheless, we will use Conway's table in the analysis of our puzzle since it has interesting properties that let us understand the analysis more easily.

Among other types of colored cube puzzles, *Instant Insanity*[1] must be the most well-known puzzle. Recently Demaine et al. analyzed the computational complexity of its variations in [8], where the research history is well-covered.

3 Preliminaries

3.1 Our Puzzle

A cube has six faces, 12 edges and eight corners. The variety of a cube is specified by how the six colors are assigned to its faces. We index the 30 varieties by means of Conway's table in Figure 1. We denote the variety in the row i and in the column j by (i, j). For a natural number n, let $[n] = \{1, 2, \ldots, n\}$. We denote the set of all the 30 varieties by V_{all}, i.e., $V_{\text{all}} = \{(i, j) \in [6] \times [6] \mid i \neq j\}$. We call a cube of variety (i, j) an (i, j)*-cube*.

In order to explain the properties of Conway's table, we introduce the notion of corner triple. For each corner of cube, we define a *corner triple* as an ordered triple of three colors around the corner such that the head element is set to the color of the lexicographically smallest letter among them and the colors are taken in the clockwise order around the corner. A cube (and thus a variety) has eight corner triples. We denote the set of eight corner triples of variety (i, j) by $T_{i,j}$. We show examples as follows.

$$\begin{aligned} T_{1,2} =& \{(p, q, t), (p, s, u), (p, t, s), (p, u, q), (q, r, t), (q, u, r), (r, s, t), (r, u, s)\}, \\ T_{2,1} =& \{(p, t, q), (p, u, s), (p, s, t), (p, q, u), (q, t, r), (q, r, u), (r, t, s), (r, s, u)\}. \end{aligned}$$

[1] "Instant Insanity" was originally trademarked by Parker Brothers in 1967. The trademark is now owned by Winning Moves, Inc.

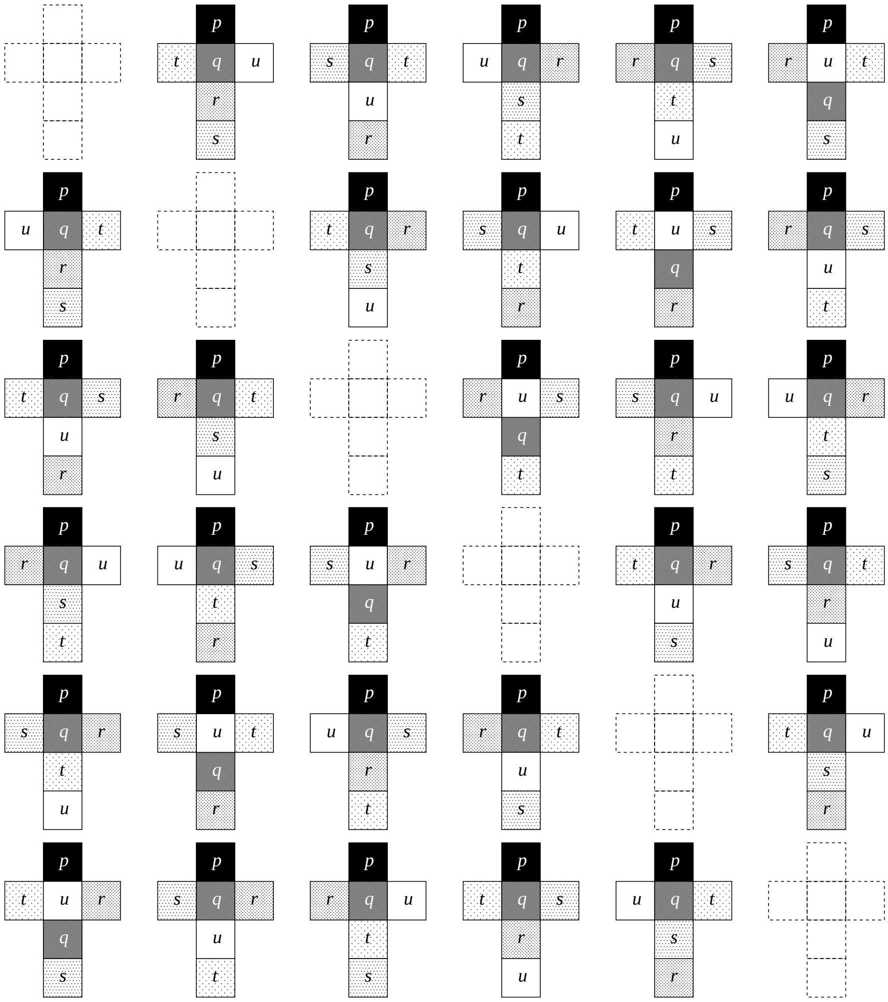


Figure 1: Conway's table of the 30 cube varieties. Each variety is represented by how the 6 colors are assigned to the development of a cube. The six colors are denoted by p, q, r, s, t, u.

Observe that 40 corner triples are possible in all. Two corner triples are the *mirror images* of each other if both contain the same three colors but are different ordered triples. For example, (p, q, t) and (p, t, q) are the mirror images of each other. Two varieties (i, j) and (i', j') are the *mirror varieties* of each other if, for each triple in $T_{i,j}$, its mirror image belongs to $T_{i',j'}$. For example, $(1, 2)$ and $(2, 1)$ are the mirror varieties of each other.

We say that two different varieties are *compatible* (resp., *incompatible*) with each other if they share at least one corner triple (resp., no corner triple). For convenience, we may say that two cubes are compatible (resp., incompatible) when their varieties are compatible (resp., incompatible) with each other. For a variety (i, j), let $V_{i,j}$ (resp., $\bar{V}_{i,j}$) denote the set of varieties compatible (resp., incompatible) with (i, j). Note that neither $V_{i,j}$ nor $\bar{V}_{i,j}$ contains the variety (i, j) itself. Every variety in $V_{i,j}$ can be obtained by changing the color assignment of (i, j) by either the "swap" operation or the "rotation" operation [6]. By the swap operation, we mean the exchange of two colors of adjacent faces that share one edge with each other. By the rotation operation, we mean the rotation of colors of the three faces around a corner so that the corresponding corner triple is unchanged. The former yields 12 varieties distinct from (i, j) and the latter yields eight varieties distinct from (i, j). The variety set $\bar{V}_{i,j}$ consists of the mirror variety of (i, j) and the mirrors of the eight varieties that are generated by performing the rotation operation on (i, j).

Proposition 1 (Proof of Lemma 2.7 in [6]) *We have $|V_{i,j}| = 20$ and $|\bar{V}_{i,j}| = 9$ for any variety (i, j).*

Moreover, whenever two varieties are compatible, they share exactly two corner triples.

Proposition 2 (Lemma 2.7 in [6]) *Two different varieties share either zero or two corner triples with each other.*

Now we present the properties of Conway's table as follows.

Property 1 *The mirror variety of (i, j) is (j, i).*

Property 2 *The five varieties on the same row or on the same column are incompatible with each other.*

From $|\bar{V}_{i,j}| = 9$, we have

$$\bar{V}_{i,j} = \{(j, i)\} \cup \{(i, j') \mid j' \in [6] \setminus \{i, j\}\} \cup \{(i', j) \mid i' \in [6] \setminus \{i, j\}\}.$$

This implies that, to compose a $2 \times 2 \times 2$ cube of variety (i, j), we cannot use cubes of the mirror variety (j, i) or cubes of the varieties on the row i or on the column j, except (i, j) itself.

An *instance* is a multi-set of colored cubes. We denote an instance by I. We represent the distribution of varieties in I by a 6×6 matrix, which we denote by its calligraphic style $\mathcal{I}$. Each value $\mathcal{I}_{i,j}$ in the matrix represents the number of (i, j)-cubes in the instance. Note that we let $\mathcal{I}_{i,i} = 0$ for any $i \in [6]$. We call $|I|$ the *size of* I. Clearly we have

$$|I| = \sum_{(i,j) \in V_{\text{all}}} \mathcal{I}_{i,j}.$$

A *solid* is a $2 \times 2 \times 2$ cube that is composed of eight $1 \times 1 \times 1$ cubes and that satisfies the Surface Color Condition. We call a solid *an (i, j)-solid* when its variety is (i, j). We say that an (i, j)-solid is *composable from I* if I contains a subset of eight cubes that can be arranged into an (i, j)-solid. An instance I is *V-generatable* if an (i, j)-solid is composable from I when and only when $(i, j) \in V$. An instance is *universal* if it is V_{all}-generatable. An instance is *infeasible* if it is $\emptyset$-generatable.

3.2 Composability Conditions

We introduce two necessary and sufficient conditions for an (i, j)-solid to be composable from a given instance I, using graph terminology. We also derive corollaries from these conditions, which we will use to solve the three problems introduced in Section 1.

The first condition is rather straightforward. We consider a bipartite graph $B_{i,j}$ such that nodes on one side are cubes in I and nodes on the other side are corner triples in $T_{i,j}$. Since I is a multi-set, it may contain multiple cubes for a single variety, and all of them appear as distinct nodes. A pair (c, τ) in $I \times T_{i,j}$ is joined by an edge whenever c has corner triple τ. The existence of edge (c, τ) indicates that c can be allocated to the corner having τ. Let the variety of c be v. From Proposition 2, the degree of c is zero if $v \in \bar{V}_{i,j}$, it is two if $v \in V_{i,j}$, and it is eight if $v = (i, j)$. A *matching* in a graph is a subset of edges such that no two edges in the subset have an endpoint in common. A matching in $B_{i,j}$ represents admissible allocation of cubes to corners, where each cube in I is allocated to at most one corner, and each corner is assigned at most one cube. Since $|T_{i,j}| = 8$, we have the following condition.

Proposition 3 *An (i, j)-solid is composable from an instance I iff there is an 8-size matching in the bipartite graph $B_{i,j}$.*

Clearly, if an 8-size matching exists, then it is a maximum cardinality matching. For any subset T of $T_{i,j}$, let $I(T)$ denote the subset of cube nodes that are adjacent to at least one node in T. That is, $I(T)$ is the subset of cubes in I that have a corner triple in T. From the Marriage Theorem [21], the following corollary is immediate.

Corollary 1 *An (i,j)-solid is composable from I iff $|T| \le |I(T)|$ holds for all possible subsets T of $T_{i,j}$.*

The second condition is based on a certain multi-graph that is obtained by changing the structure of $B_{i,j}$ as follows; remove all cube nodes of degrees 0 and 8 and the connecting edges, and then shrink every cube node of degree 2 and its connecting edges into a new edge. As a result, only $T_{i,j}$ remains as the node set. Each edge is associated with a cube compatible with (i,j) and multi-edges may appear. We denote this multi-graph by $G_{i,j} = (T_{i,j}, E_{i,j})$. Since all compatible cubes in I appear as edges, we have $|E_{i,j}| = \sum_{(i',j') \in V_{i,j}} \mathcal{I}_{i',j'}$.

Proposition 4 *An (i,j)-solid is composable from an instance I iff $\mathcal{I}_{i,j}$ is no less than the number of tree components in the graph $G_{i,j}$.*

Proof: For the necessity, let us denote the number of tree components by k. Suppose $\mathcal{I}_{i,j} < k$. We denote the set of all nodes in the k tree components by T and the set of all cubes appearing in the components as edges by I'. We have $|I'| = |T| - k$. The set T of corner triples is a node subset in the bipartite graph $B_{i,j}$. We have $I(T) = I' \cup \{(i,j)\text{-cubes in } I\}$, where the second cube set in the right side may be a multi-set. It follows that:

$$|I(T)| = |I'| + \mathcal{I}_{i,j} = |T| - k + \mathcal{I}_{i,j} < |T|.$$

By Corollary 1, we cannot compose an (i,j)-solid in any way.

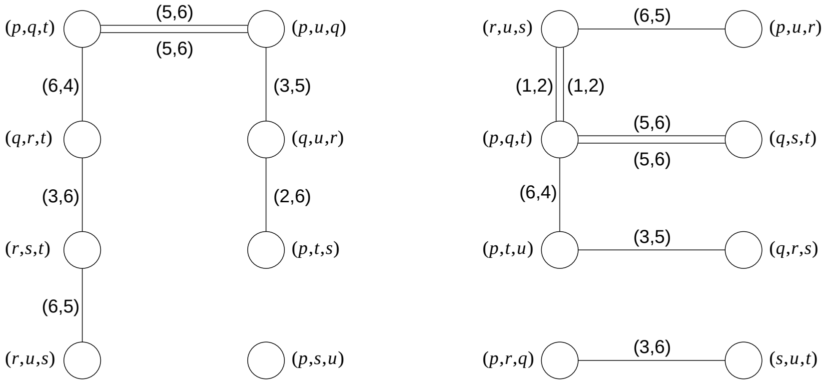
For the sufficiency, if an (i,j)-solid is not composable, then there is a subset T of $T_{i,j}$ such that $|T| > |I(T)| = \mathcal{I}_{i,j} + |I'|$, where I' is the subset of compatible cubes in I that are adjacent to a node in T. Consider the subgraph of $G_{i,j}$ induced by T. The number of edges in the subgraph is equal to $|I'|$. Denoting the number of tree components in the subgraph by k, we have $|I'| \ge |T| - k$. Since $|T| > \mathcal{I}_{i,j} + |T| - k$, we have $k > \mathcal{I}_{i,j}$. $\square$

We give an example of how Proposition 4 works. In Figure 2, we show graphs $G_{1,2}$ and $G_{2,3}$ for the instance I with nine cubes that has the following matrix representation:

$$\mathcal{I} = \begin{pmatrix} 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}. \tag{1}$$

In the figure, a node (resp., an edge) is labeled with its corresponding corner triple (resp., cube variety). We see that a $(1,2)$-solid is composable since there is only one tree component in $G_{1,2}$ (i.e., an isolated point (p,s,u)) and $\mathcal{I}_{1,2} = 2$. On the other hand, a $(2,3)$-solid is not composable since there is a tree component in $G_{2,3}$ but $\mathcal{I}_{2,3} = 0$.

As a corollary to Proposition 4, we derive a sufficient condition for an (i^*, j^*)-solid to be composable from a given I. Let us consider composing an (i^*, j^*)-solid from only (i^*, j^*)-cubes and cubes whose varieties are in a single row $i_0 \ne i^*$ (or a single column $j_0 \ne j^*$ analogously) of Conway's table. In the row i_0, exactly four varieties are compatible with (i^*, j^*) from Property 2. Observe a subgraph of G_{i^*,j^*} such that the node set is T_{i^*,j^*} and the edge set is a subset of those coming from cubes of the four varieties. Since the four varieties are pairwise incompatible by Property 2, edges coming from different varieties do not touch with each other. Each tree component in the subgraph is an isolated point or consists of two nodes connected by just one edge. The number of tree components of the former kind is $2|\{(i_0, j) \in V_{i^*,j^*} \mid \mathcal{I}_{i_0,j} = 0\}|$ and that



$G_{1,2}$ $G_{2,3}$

Figure 2: Graphs $G_{1,2}$ and $G_{2,3}$ for the instance I of (1)

of the latter kind is $|\{(i_0, j) \in V_{i^*,j^*} \mid \mathcal{I}_{i_0,j} = 1\}|$. Hence, the total number of tree components is:

$$\begin{aligned}
&2|\{(i_0, j) \in V_{i^*,j^*} \mid \mathcal{I}_{i_0,j} = 0\}| + |\{(i_0, j) \in V_{i^*,j^*} \mid \mathcal{I}_{i_0,j} = 1\}| \\
&+ 0|\{(i_0, j) \in V_{i^*,j^*} \mid \mathcal{I}_{i_0,j} \geq 2\}| \\
= &\sum_{(i_0,j)\in V_{i^*,j^*}} (2 - \min\{\mathcal{I}_{i_0,j}, 2\}) \\
= &8 - \sum_{(i_0,j)\in V_{i^*,j^*}} \min\{\mathcal{I}_{i_0,j}, 2\}.
\end{aligned}$$

By Proposition 4, if $\mathcal{I}_{i^*,j^*}$ is no less than the above number, then we can compose an (i^*, j^*)-solid. More generally, if $\mathcal{I}_{i^*,j^*} + \sum_{(i_0,j)\in V_{i^*,j^*}} \min\{\mathcal{I}_{i_0,j}, 2\} \geq 8$ holds in at least one row $i_0 \neq i^*$, or analogously, if $\mathcal{I}_{i^*,j^*} + \sum_{(i,j_0)\in V_{i^*,j^*}} \min\{\mathcal{I}_{i,j_0}, 2\} \geq 8$ holds in at least one column $j_0 \neq j^*$, then we can compose an (i^*, j^*)-solid. We define $f(i^*, j^*)$ as follows:

$$f(i^*, j^*) = \mathcal{I}_{i^*,j^*} + \max\{f_{\text{row}}(i^*, j^*), f_{\text{col}}(i^*, j^*)\}, \tag{2}$$

where f_{row} and f_{col} are defined as:

$$f_{\text{row}}(i^*, j^*) = \max_{i_0\in[6]\setminus\{i^*\}}\{\sum_{(i_0,j)\in V_{i^*,j^*}} \min\{\mathcal{I}_{i_0,j}, 2\}\}, \tag{3}$$

$$f_{\text{col}}(i^*, j^*) = \max_{j_0\in[6]\setminus\{j^*\}}\{\sum_{(i,j_0)\in V_{i^*,j^*}} \min\{\mathcal{I}_{i,j_0}, 2\}\}. \tag{4}$$

Corollary 2 *If $f(i^*, j^*) \geq 8$, then an (i^*, j^*)-solid is composable.*

4 Answers to Existence, MaxInfeasible and MinUniversal

In this section, we present answers to the three problems, Existence, MaxInfeasible and MinUniversal. To draw the answers, we utilize the power of a computer search; we formulate the problems by CP and IP and then solve them using computational packages.

Let us begin with description of CP and IP. The CP technique that we use is *Constraint Satisfaction Problem* (*CSP*). In a general setting of CSP, we are given a set of *variables*, each of which has its *domain*, and a set of *constraints* restricting the values that the variables can simultaneously take. We are asked to assign a value from its domain to every variable, in such a way that every constraint is satisfied. The Global Constraint Catalog [13] summarizes constraint types that are typically used in CSP. The key techniques to solve CSP include *backtracking*, *constraint propagation*, and so on [7].

Concerning IP, we employ *Integer Linear Programming* (*ILP*) among various models; given a linear function of integer variables called an *objective* and a set of linear equalities and/or inequalities representing constraints, we are asked to assign integers to the variables so that the objective value is minimized (or maximized) and that, at the same time, all the constraints are satisfied. The key techniques to solve ILP include *linear programming*, *branch-and-bound*, *cutting plane*, and so on [30].

The software choice is significant since it has a great influence on computation time. For CSP, we use Sugar (ver. 2.2.1) [28] that solves a CSP model by means of solving a corresponding SAT model. Sugar transforms an input CSP model into an artificial one by preprocessing, encodes it into a SAT model by a certain sophisticated algorithm, and then runs a SAT solver. We use MiniSat (ver. 2.2.1) [11] as the SAT solver. For ILP, we utilize IBM ILOG CPLEX (ver. 12.6) [18]. Both are recognized as excellent software packages.

The computation was conducted on a workstation that carries Intel® Core™ i7-4770 Processor (up to 3.90GHz by means of Turbo Boost Technology) and 8GB main memory. The installed OS is Ubuntu 14.04.1.

We formulate Existence and MaxInfeasible by CSP and MinUniversal by ILP. CSP and ILP are so robust that our three problems can be formulated by either model, but they have different weak points from each other. For example, ILP is not very suitable for formulating disjunctive constraints since we are possibly required to introduce a large number of artificial variables and constraints. Problems Existence and MaxInfeasible contain disjunctive constraints, and in our preliminary study, ILP models for these two problems are not solved within 10^4 seconds; CSP is more suitable for formulating them. On the other hand, MinUniversal does not contain a disjunctive constraint, and the ILP model takes only a couple of seconds to find a minimum universal instance, while the CSP model takes 4.5×10^3 seconds.

4.1 Problem Existence

To say no, it suffices to present a variety subset V of V_{all} for which a V-generatable instance does not exist. Based on our preliminary study, we conjecture that V consisting of five varieties in the same row (or column) should be such a subset.

Let us set $V = \{(1,2),(1,3),(1,4),(1,5),(1,6)\}$. We formulate the problem of finding a V-generatable instance by CSP, expecting the software Sugar to say that such an instance does not exist. In the CSP model, we have 30 non-negative integer variables, denoted by $x_{1,2}, x_{1,3}, \ldots, x_{6,5}$, that represent the distribution of cubes over the 30 varieties, that is, $x_{i,j} = \mathcal{I}_{i,j}$. The domain of each $x_{i,j}$ is set as follows:

$$\begin{array}{ll} x_{i,j} \in \{0,1,\ldots,8\} & \text{if } (i,j) \in V, \\ x_{i,j} \in \{0,1,2\} & \text{otherwise.} \end{array} \tag{5}$$

We assume $x_{i,j} \leq 8$ since a ninth cube is redundant. Furthermore, we assume $x_{i,j} \leq 2$ for any $(i,j) \notin V$ since (i,j)-cubes can be used to compose an (i^*,j^*)-solid ($(i^*,j^*) \in V$) at most twice and thus a third cube is unnecessary. The constraints include:

(I) The instance size is at least eight.

(II) For each composable variety $(i^*,j^*) \in V$, $|T| \leq |I(T)|$ holds for every subset T of T_{i^*,j^*}.

(III) For each non-composable variety $(i^*,j^*) \notin V$, $|T| > |I(T)|$ holds for at least one subset T of T_{i^*,j^*}.

(II) and (III) come from the composability condition of Corollary 1.

CSP deals with a constraint that is represented by a linear inequality of variables. (I) and (II) can be expressed by a set of linear inequalities; For (I),

$$\sum_{(i,j) \in V_{\text{all}}} x_{i,j} \geq 8. \tag{6}$$

For (II),

$$\forall (i^*, j^*) \in V,\ \forall T \subseteq T_{i^*,j^*}, \quad x_{i^*,j^*} + \sum_{(i,j) \in V_{i^*,j^*}:\ T_{i,j} \cap T \neq \emptyset} x_{i,j} \geq |T|, \tag{7}$$

where the left side of the inequality is equal to $|I(T)|$.

(III) is a disjunctive constraint. To express this, we employ OR constraint:

$$\forall (i^*, j^*) \notin V, \quad \bigvee_{T \subseteq T_{i^*,j^*}} \big(x_{i^*,j^*} + \sum_{(i,j) \in V_{i^*,j^*}:\ T_{i,j} \cap T \neq \emptyset} x_{i,j} < |T| \big). \tag{8}$$

Furthermore, to increase the computational efficiency, we reduce the search space by introducing the following constraint that is derived from Corollary 2:

$$\forall (i^*, j^*) \notin V, \quad f(i^*, j^*) \leq 7.$$

Observing (2) to (4), we express the constraint by a set of linear inequalities and MIN constraints as follows:

$$\forall (i^*, j^*) \notin V,\ \forall i_0 \in [6] \setminus \{i^*\}, \qquad x_{i^*,j^*} + \sum_{(i_0,j) \in V_{i^*,j^*}} \min\{x_{i_0,j}, 2\} \leq 7, \tag{9}$$

$$\forall (i^*, j^*) \notin V,\ \forall j_0 \in [6] \setminus \{j^*\}, \qquad x_{i^*,j^*} + \sum_{(i,j_0) \in V_{i^*,j^*}} \min\{x_{i,j_0}, 2\} \leq 7. \tag{10}$$

Finally, our primitive CSP model has 30 integer variables with domains in (5) and constraints from (6) to (10). As a part of preprocessing, Sugar transforms this naïve model into an artificial one that has 4.5×10^4 integer variables, 6.3×10^3 Boolean variables and 7.5×10^5 constraints, and then encodes it into a SAT model that has 5.9×10^5 variables and 1.1×10^7 clauses. After 1.6×10^3-second computation, MiniSat decides that the SAT model is not satisfiable, which means that no V-generatable instance exists.

Theorem 1 *There is a subset V of V_{all} such that a V-generatable instance does not exist.*

The above argument implies that, if all five varieties in a row or a column of Conway's table are composable, then there is a composable variety outside the row or the column. So far, however, we have not gained any insight into which variety it might be.

4.2 Problem MaxInfeasible

It is Berkove et al. [6] who first studied MaxInfeasible. They showed the maximum size of infeasible instance to be at least 22, presenting a concrete example of a 22-size infeasible instance as the certificate. They conjectured the maximum size to be 23, leaving it open.

We show by construction that a 23-size infeasible instance exists. We then show that no 24-size infeasible instance exists by means of CSP. This answers Berkove et al.'s open problem.

Theorem 2 *There is an infeasible instance of size* 23*.*

Proof: We show that the instance N with the following matrix representation is infeasible:

$$\mathcal{N} = \begin{pmatrix} 0 & 7 & 7 & 7 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{11}$$

The size of N is exactly 23. We show that no (i, j) satisfies the composability condition of Proposition 4. Recall Properties 1 and 2 of Conway's table. For every $(1, j)$ with $j \in \{2, \dots, 6\}$, the graph $G_{1,j}$ consists of eight isolated points, while $\mathcal{N}_{1,j}$ is smaller than eight. Hence, a $(1,j)$-solid is not composable.

For $(i,j) = (2,1)$, the edges appearing in $G_{2,1}$ come from the cubes of the varieties $(1,3),\ldots,(1,6)$ since $(1,2)$ is the mirror variety of $(2,1)$. The edges from the same variety form multi-edges, and those coming from different varieties do not touch with each other since they are pairwise incompatible. There are four connected components in $G_{2,1}$, each of which consists of two nodes. Among these, two connected components contain only one edge respectively, which are tree components, but N has no $(2,1)$-cube. Hence, a $(2,1)$-solid is not composable. The remaining cases can be shown in the same way. $\square$

Lemma 1 *There is no infeasible instance of size* 24 *or greater.*

PROOF: Setting $V = \emptyset$, we show that a 24-size V-generatable instance does not exist by utilizing CSP. Here we do not use the inequality constraint (6) that gives a lower bound on the instance size but an equality constraint that fixes the size to 24:

$$\sum_{(i,j)\in V_{\mathrm{all}}} x_{i,j} = 24. \tag{12}$$

The CSP model in this case has 30 integer variables with domains in (5) and constraints from (7) to (10) and (12). We run SUGAR to solve the model; the preprocessed CSP model has 7.4×10^4 integer variables, 7.7×10^3 Boolean variables, and 1.3×10^5 constraints. It is then encoded into a SAT model that has 1.8×10^6 variables and 4.2×10^7 clauses. After 4.4×10^3-second computation, MINISAT reports that no solution exists, meaning that no 24-size infeasible instance exists. Clearly, any larger instance is not infeasible. $\square$

Theorem 3 *The maximum size of infeasible instance is* 23. *The instance with the matrix representation in* (11) *is an example.*

4.3 Problem MINUNIVERSAL

First, we give a lower bound on the size of universal instance. We then show that the bound is tight by presenting a universal instance of that size.

Lemma 2 *A universal instance should contain at least* 12 *cubes.*

PROOF: Let U be a universal instance. For every variety (i,j) in V_{all}, since an (i,j)-solid is composable, the number $|E_{i,j}| + \mathcal{U}_{i,j}$ should be at least eight. The sum of this number over the 30 varieties is at least $30 \times 8 = 240$. On the other hand, by Proposition 1, each cube in U contributes to the sum in exactly 21 varieties; denoting the variety of the cube by (i',j'), once for $\mathcal{U}_{i',j'}$, and 20 times for an edge in $G_{i,j}$ such that $(i',j') \in V_{i,j}$. Hence we have $|U| \geq 240/21 > 11$. $\square$

Theorem 4 *The minimum size of universal instance is* 12. *The instance U with the following matrix representation is an example*:

$$\mathcal{U} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}.$$

Proof: The size of U is exactly 12. One can verify the universality by checking that Proposition 3 or 4 holds for all the 30 varieties. □

We find the minimum universal instance above by solving the following ILP.

$$\begin{array}{ll} \textbf{minimize} & \sum_{(i,j)\in V_{\text{all}}} x_{i,j} \\ \textbf{subject to} & (7) \text{ with respect to } V = V_{\text{all}} \\ & \forall (i,j) \in V_{\text{all}},\quad 0 \le x_{i,j} \le 8 \text{ and } x_{i,j} \in \mathbb{Z} \end{array}$$

This ILP model has 30 integer variables, $x_{1,2}, x_{1,3}, \ldots, x_{6,5}$, and 7.7×10^3 inequality constraints, and does not contain the disjunctive constraint (III). IBM ILOG CPLEX solves it within a couple of seconds, providing $\mathcal{U}$ as output.

5 Concluding Remarks

We solved three problems on a generalization of "Eight Blocks to Madness" puzzle, i.e., Existence, MaxInfeasible and MinUniversal, with the help of a computer search. The answers are summarized as Theorems 1, 3 and 4, respectively. In particular, MaxInfeasible answers an open problem in [6].

Smarter proofs are expected, especially for Existence and MaxInfeasible, since the proofs are computer-assisted; we admit that "computer-assisted proof" is a controversial proof technique [22].

We leave two open problems.

Open Problem 1 Characterize subsets V of V_{all} for which a V-generatable instance exists.

We have seen that a V-generatable instance exists if $V = \emptyset$ or V_{all}, whereas it does not exist if V is a set of five varieties in a single row (or column) of Conway's table (Theorem 1). However, we are not sure what discriminates variety subsets that have generatable instances and ones that do not have generatable instances.

Open Problem 2 Characterize subsets V of V_{all} for which every V-generatable instance is finite.

This problem is restricted to V for which a V-generatable instance exists. Here we do not restrict $\mathcal{I}_{i,j}$ to be no more than eight but permit it to be infinite. When $V = \emptyset$, all V-generatable instances are finite because the sizes are no more than 23 (Theorem 3). On the other hand, when $V = V_{\text{all}}$ or $|V| = 1$, there exists a V-generatable instance of infinite size. It is interesting to explore what determines the finiteness of V-generatable instance.

Although being quite simple and classical, colored cube puzzles still provide us with many mathematical problems. We hope this paper forms a basis of additional research in recreational mathematics.

Acknowledgments

We gratefully acknowledge very careful and detailed comments given by anonymous reviewers. A large part of this work was accomplished when the author worked for Ishinomaki Senshu University in Japan. He is deeply grateful to Professor Akira Maruoka for his proofreading, enthusiastic guidance on technical writing and persistent encouragement. The author would also like to thank undergraduate students Masaki Ishigaki, Kouta Kakizaki and Kousuke Sakai for working on preliminary studies. This work is partially supported by JSPS KAKENHI Grant Number 25870661.

References

[1] Greg Aloupis, E. D. Demaine, Alan Guo, and Giovanni Viglietta. Classic nintendo games are (computationally) hard. In *Proc. Fun with Algorithms (FUN2014)*, volume 8496 of *LNCS*, pages 40–51. 2014.

[2] Daniel Andersson. Hashiwokakero is NP-complete. *Information Processing Letters*, 109(19):1145–1146, 2009.

[3] Kenneth Appel and Wolfgang Haken. Every planar map is four colorable part i. discharging. *Illinois J. Mathematics*, 21:429–490, 1977.

[4] Kenneth Appel and Wolfgang Haken. Solution of the four color map problem. *Scientific American*, 237(4):108–121, 1977.

[5] Kenneth Appel, Wolfgang Haken, and John Koch. Every planar map is four colorable part ii. reducibility. *Illinois J. Mathematics*, 21:491–567, 1977.

[6] E. Berkove, J. V. Sickle, B. Hummon, and J. Kogut. An analysis of the (colored cubes)^3 puzzle. *Discrete Mathematics*, 308:1033–1045, 2008.

[7] Rina Dechter, editor. *Constraint Processing*. Morgan Kaufmann Publishers, San Francisco, CA, USA, 2003.

[8] E. D. Demaine, M. L. Demaine, Sarah Eisenstat, T. D. Morgan, and Ryuhei Uehara. Variations on instant insanity. In *Space-Efficient Data Structures, Streams, and Algorithms*, volume 8066 of *LNCS*, pages 33–47, 2013.

[9] Erik D. Demaine, Susan Hohenberger, and David Liben-Nowell. Tetris is hard, even to approximate. In *Proc. 9th Annual International Conference on Computing and Combinatorics (COCOON'03)*, pages 351–363, 2003.

[10] Erik D. Demaine, Yoshio Okamoto, Ryuhei Uehara, and Yushi Uno. Computational complexity and an integer programming model of Shakashaka. *IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences*, E97-A(6):1219–1219, 2014.

[11] Niklas Eén and Niklas Sörensson. The MiniSat Page. `http://minisat.se/Main.html`, accessed on February 1, 2015.

[12] M. Gardner. *Fractal Music, Hypercards and More...* W. H. Freeman and Company, New York, NY, USA, 1991.

[13] Global Constraint Catalog. `http://sofdem.github.io/gccat/gccat/index.html`, accessed on February 1, 2015.

[14] Kazuya Haraguchi. The number of inequality signs in the design of Futoshiki puzzle. *J. Information Processing*, 21(1):26–32, 2013.

[15] Kazuya Haraguchi, Yasutaka Abe, and Akira Maruoka. How to produce Blocksum instances with various levels of difficulty. *J. Information Processing*, 20(3):727–737, 2012.

[16] Kazuya Haraguchi and Hirotaka Ono. Blocksum is NP-complete. *IEICE Trans. Information and Systems*, E96-D(3):481–488, 2013.

[17] Robert A Hearn and Erik D Demaine. *Games, Puzzles, and Computation*. AK Peters Limited, Natick, MA, USA, 2009.

[18] IBM ILOG CPLEX Optimizer. `http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/`, accessed on February 1, 2015. ver. 12.6.

[19] S. J. Kahan. 'Eight Blocks to Madness' - a logical solution. *Mathematics Magazine*, 45:57–65, 1972.

[20] Jonas Kölker. Kurodoko is NP-complete. *J. Information Processing*, 20(3):694–706, 2012.

[21] L. Lovász and M. D. Plummer. *Matching Theory*. American Mathematical Society, Providence, RI, USA, 2009.

[22] Donald MacKenzie. *Mechanizing Proof: Computing, Risk, and Trust.* MIT Press, Cambridge, MA, USA, 2001.

[23] P. A. MacMahon. *New Mathematical Pastimes.* Cambridge University Press, Cambridge, UK, 1921.

[24] MacMahon's Coloured Cubes. `http://www.mathematische-basteleien.de/macmahon.htm`, accessed on February 1, 2015.

[25] Gary McGuire, Bastian Tugemann, and Gilles Civario. There is no 16-clue Sudoku: Solving the Sudoku minimum number of clues problem. *CoRR*, abs/1201.0749, 2012.

[26] Rob's Puzzle Page -Pattern Puzzles-. `http://home.comcast.net/~stegmann/pattern.htm`, accessed on February 1, 2015.

[27] A. Sobczyk. More progress to madness via 'Eight Blocks'. *Mathematics Magazine*, 47:115–124, 1974.

[28] Naoyuki Tamura. Sugar: a SAT-based Constraint Solver. `http://bach.istc.kobe-u.ac.jp/sugar/`, accessed on February 1, 2015.

[29] Toby Walsh. Candy crush is NP-hard. *CoRR*, abs/1403.1911, 2014.

[30] Laurence A. Wolsey, editor. *Integer Programming.* Wiley-Interscience, New York, NY, USA, 1998.

[31] Takayuki Yato and Takahiro Seta. Complexity and completeness of finding another solution and its application to puzzles. *IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences*, E86-A(5):1052–1060, 2003.