Theoretical Computer Science 785 (2019) 128–139

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

# Ruleset optimization on isomorphic oritatami systems ☆,☆☆

Yo-Sub Han [a], Hwee Kim [b,*]

[a] Department of Computer Science, Yonsei University, 50 Yonsei-Ro, Seodaemun-Gu, Seoul 03722, Republic of Korea
[b] Department of Mathematics and Statistics, University of South Florida, 4202 E. Fowler Ave., Tampa, FL 33620, USA

ARTICLE INFO

*Article history:*
Received 6 December 2018
Received in revised form 8 March 2019
Accepted 13 March 2019
Available online 27 March 2019
Communicated by T. Yokomori

*Keywords:*
Oritatami system
Self-assembly
Optimization
RNA cotranscriptional folding

ABSTRACT

We study an optimization problem of a computational folding model, proving its hardness and proposing heuristic algorithms. RNA cotranscriptional folding refers to the phenomenon in which an RNA transcript folds upon itself while being synthesized out of a gene. An oritatami model (OM) is a computational model of this phenomenon that lets its sequence of beads (abstract molecules) fold cotranscriptionally by the interactions between beads, according to its ruleset. We study the problem of reducing the ruleset size, while keeping the terminal conformations geometrically the same. We first prove the hardness of finding the smallest ruleset, and then suggest two approaches that reduce the ruleset size efficiently.



## 1. Introduction

In nature, a one-dimensional RNA sequence folds itself autonomously, giving rise to a high-dimensional tertiary structure. However, predicting the tertiary structure from a primary structure is a challenge. Based on experimental observations, researchers have established various RNA structure prediction models, including RNAfold [1], Pknots [2], mFold [3], and KineFold [4].

Recently, biochemists showed that the kinetics—the step-by-step dynamics of the reaction—plays an essential role in the geometric shape of the RNA foldings [6], because the folding caused by intermolecular reactions is faster than the RNA transcription rate [7]. By controlling cotranscriptional foldings, researchers have succeeded in assembling a rectangular tile out of RNA, which is called RNA Origami [5]. This cotranscriptional folding is observed even at a single-nucleotide resolution [8]. From this kinetic point of view, Geary et al. [9] proposed a new folding model called an oritatami model (OM) as in Fig. 1. An Oritatami System (OS) consists of a sequence of beads (the transcript) and a set of rules for the possible intermolecular reactions between beads. An OS folds its bead sequence as follows. For each bead, the OS determines the best location that maximizes the number of possible interactions, using a lookahead of a few upcoming beads, and then places the current bead at that location. Then, it reads the next bead, and repeats the procedure until there are no further beads. The lookahead represents the reaction rate of the cotranscriptional folding, and the number of interactions represents the energy level. In an OS, we call the secondary structure *the conformation*, and the resulting secondary structure *the terminal*

☆ Han was supported by the International Cooperation Program (NRF-2017K2A9A2A08000270) and the Basic Science Research Program (NRF-2018R1D1A1A09084107). Kim was supported in part by the NIH grant R01 GM109459 and NSF grant DMS-1800443.
☆☆ The preliminary version of this work had been published in the proceedings of the 23rd International Conference on DNA Computing and Molecular Programming.
* Corresponding author.
*E-mail addresses:* emmous@yonsei.ac.kr (Y.-S. Han), hweekim@mail.usf.edu (H. Kim).

https://doi.org/10.1016/j.tcs.2019.03.020
0304-3975/

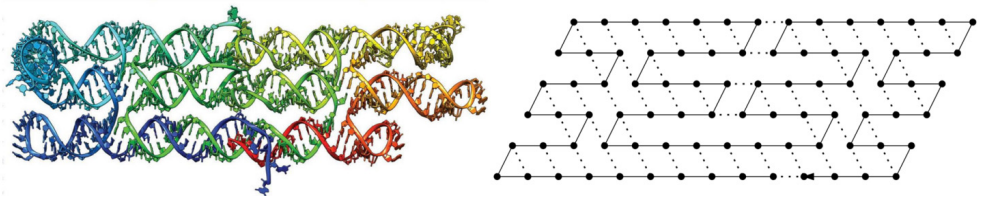

**Fig. 1.** (Left) An example of an RNA tile generated by RNA Origami [5]. (Right) A conformation representing the RNA tile in OM. The directed solid line represents a path, dots represent beads, and dotted lines represent interactions.

| RNA Origami | Oritatami Model |
|---|---|
| (A set of) Nucleotides | Beads |
| Transcript | Sequence of beads connected by a line |
| h-bonds between nucleotides | Interactions |
| Cotranscriptional folding rate | Delay |
| Resulting secondary structure | Conformation |

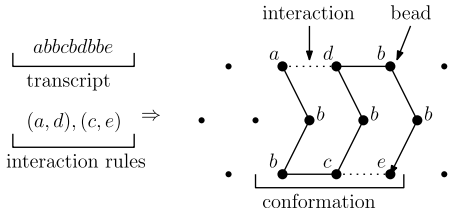


**Fig. 2.** A Comparison of RNA origami and oritatami model.

*conformation*. Fig. 2 compares RNA origami and oritatami model. Geary et al. implemented an OS counting in binary [10] and an OS simulating a cyclic tag system [9]. Han et al. [11] implemented an OS to solve the DNF tautology problem, and proved the hardness of the OS equivalence problem. Han et al. [12] proposed the problem of removing self-attracting rules, and proved upper and lower bounds for the number of copies of bead types required to remove self-attracting rules.

RNA consists of only four nucleotides (A, G, C, U) and three interaction rules (A-U, C-G, G-U). However, we can design a system with more bead types and rules in experiments by combining multiple nucleotides to represent a bead type [5,10]. It is straightforward that with more rules, it becomes more difficult to realize the system in experiments. Using experiments, biochemists have studied methods that synthesize the desired structure using a smaller number of basic components [13]. This motivates us to consider the problem of reducing the size of the alphabet and the ruleset from a theoretical point of view. Since an OS folds its transcript on the triangular lattice, it is important to preserve its geometric properties, including the transcript path and interactions between beads, while reducing the ruleset. Geary et al. [10] proved that, given a set of paths and a transcript, it is NP-complete to find a ruleset that folds the transcript to the set of paths. Ota and Seki [14] proved that, given a path, a transcript, and a set of interactions, it is NP-complete to find a ruleset that folds the transcript to the path according to the given interactions. However, there is no research on reducing and optimizing the ruleset of a given OS while preserving all its geometric properties.

We say that two OSs are isomorphic if both have the same geometric properties. We first prove that, *given an OS, it is NP-hard to find the smallest ruleset of an isomorphic OS*, in general. Then, we propose two practical approaches to the problem: 1) the bead-type merging method, which merges two bead types that have the same interactions with other bead types; and 2) a representative fuzzy ruleset construction, which is a set of rulesets that results in the same set of terminal conformations. *We design efficient algorithms to find a representative fuzzy ruleset from a given OS*, reduce the size of the fuzzy ruleset using a modified bead-type merge, and construct a reduced ruleset from the fuzzy ruleset.

## 2. Preliminaries

Let $\Sigma$ be a finite set of types of abstract molecules, or *bead types*. By $\Sigma^*$ (respectively, $\Sigma^\omega$), we denote the set of finite (one-way infinite) sequences of bead types in $\Sigma$. A sequence $w$ in $\Sigma^*$ can be represented as $w = b_1 b_2 \cdots b_n$, for some $n \geq 0$ and bead types $b_1, b_2, \ldots, b_n \in \Sigma$, where $n$ is the *length* of $w$ and is denoted by $|w|$; in other words, a sequence $w$ is a string over $\Sigma$. The sequence of length 0 is denoted by $\lambda$. For $1 \leq i \leq j \leq n$, the subsequence of $w$ ranging from the $i$-th bead to $j$-th bead is denoted by $w[i:j]$; that is, $w[i:j] = b_i b_{i+1} \cdots b_j$. This notation is simplified as $w[i]$ when $j = i$, referring to the $i$-th bead of $w$. For $k \geq 1$, $w[1:k]$ is a *prefix* of $w$. We use $w = w_1 \cdot w_2$, or simply $w_1 w_2$ to denote the catenation of two strings $w_1$ and $w_2$.

An undirected graph $G = (V, E)$ consists of a finite nonempty set $V$ of nodes, and a set $E$ of unordered pairs of nodes of $V$. Each pair $e = \{u, v\}$ of nodes in $E$ is an edge of $G$, and $e$ is said to join $u$ and $v$. A weighted graph $G = (V, E)$ is a graph where each edge $e = (u, v)$ has an assigned weight $w(e)$. We denote an edge between $u$ and $v$ with a weight $w$ in a weighted graph by $e = (\{u, v\}, w)$. The reader may refer to Gibbons [15] for more details in graph theory.

Oritatami systems fold their transcript, a sequence of beads, over the triangular lattice cotranscriptionally by letting nascent beads form as many hydrogen-bond-based interactions (*h-interactions*, or simply *interactions*) as possible, according to a given set of rules. Let $\mathbb{T} = (V, E)$ be the triangular grid graph. A directed simple path $P = p_1 p_2 \cdots$ in $\mathbb{T}$ is a possibly infinite sequence of pairwise-distinct points (vertices). Let $P[i]$ be the $i$-th point $p_i$ and $|P|$ be the number of points in $P$. A *ruleset* $\mathcal{H} \subset \Sigma \times \Sigma$ is a symmetric relation over the set of pairs of bead types, such that, for all bead types $a, b \in \Sigma$, $(a, b) \in \mathcal{H}$ implies $(b, a) \in \mathcal{H}$.

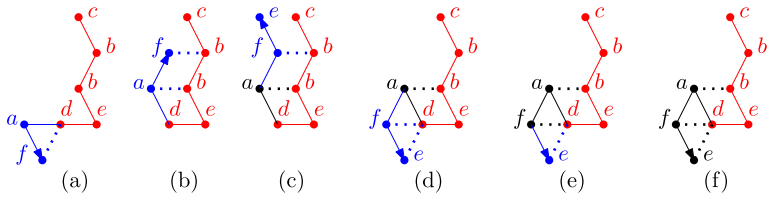


**Fig. 3.** An example OS with delay 2 and arity 2. The seed is colored in red, and the stabilized beads and interactions are colored in black. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

A conformation instance, or *configuration*, is a triple $(P, w, H)$ of a directed path $P$ in $\mathbb{T}$, $w \in \Sigma^* \cup \Sigma^\omega$, and a set $H \subseteq \left\{(i,j) \mid 1 \le i, i+2 \le j, \{P[i], P[j]\} \in E\right\}$ of interactions. This is interpreted as the sequence $w$ being folded, while its $i$-th bead $w[i]$ is placed on the $i$-th point $P[i]$ along the path, and there is an interaction between the $i$-th and $j$-th beads if and only if $(i,j) \in H$. Configurations $(P_1, w_1, H_1)$ and $(P_2, w_2, H_2)$ are *congruent* provided $w_1 = w_2$, $H_1 = H_2$, and $P_1$ can be transformed into $P_2$ by a combination of a translation, a reflection, and rotations by 60 degrees. The set of all configurations congruent to a configuration $(P, w, H)$ is called the *conformation* of the configuration, and is denoted by $C = [(P, w, H)]$. We call $w$ a *primary structure* of $C$. Given a conformation $C$, we say that a point $p$ is *annotated* in $C$ if there exists a bead placed on $p$, and *unannotated* otherwise.

Let $\mathcal{H}$ be a ruleset. An interaction $(i,j) \in H$ is *valid with respect to* $\mathcal{H}$, or simply $\mathcal{H}$*-valid*, if $(w[i], w[j]) \in \mathcal{H}$. We say that a conformation $C$ is $\mathcal{H}$*-valid* if all of its interactions are $\mathcal{H}$-valid. For an integer $\alpha \ge 1$, $C$ is *of arity* $\alpha$ if the maximum number of interactions per bead is $\alpha$; that is, if for any $k \ge 1$, $\left|\{i \mid (i,k) \in H\}\right| + \left|\{j \mid (k,j) \in H\}\right| \le \alpha$, and this inequality holds as an equation for some $k$. Then, we use $\mathcal{C}_{\le\alpha}$ to denote the set of all conformations of arity at most $\alpha$.

Oritatami systems grow conformations by elongating them under their own ruleset. For a finite conformation $C_1$, we say that a finite conformation $C_2$ is an *elongation* of $C_1$ by a bead $b \in \Sigma$ under a ruleset $\mathcal{H}$, written as $C_1 \xrightarrow{\mathcal{H}}_b C_2$, if there exists a configuration $(P, w, H)$ of $C_1$, such that $C_2$ includes a configuration $(P \cdot p, w \cdot b, H \cup H')$, where $p \in V$ is a point not in $P$ and $H' \subseteq \left\{(i, |P|+1) \mid 1 \le i \le |P|-1, \{P[i], p\} \in E, (w[i], b) \in \mathcal{H}\right\}$. This operation is recursively extended to the elongation by a finite sequence of beads, as follows: For any conformation $C$, $C \xrightarrow{\mathcal{H}^*}_\lambda C$, and for a finite sequence of beads $w$ and a bead $b$, a conformation $C_1$ is elongated to a conformation $C_2$ by $w \cdot b$, written as $C_1 \xrightarrow{\mathcal{H}^*}_{w \cdot b} C_2$, if there is a conformation $C'$ that satisfies $C_1 \xrightarrow{\mathcal{H}^*}_w C'$ and $C' \xrightarrow{\mathcal{H}}_b C_2$. We denote the set of all conformations that are elongated to $C_2$ by $w \cdot b$ under $\alpha$ as $\mathcal{E}_\alpha(C_2, w \cdot b)$.

An *oritatami system (OS)* is a 6-tuple $\Xi = (\Sigma, w, \mathcal{H}, \delta, \alpha, C_\sigma = [(P_\sigma, w_\sigma, H_\sigma)])$, where $\mathcal{H}$ is a *ruleset*, $\delta \ge 1$ is a *delay*, and $C_\sigma$ is an $\mathcal{H}$-valid initial *seed* conformation of arity at most $\alpha$, upon which its *transcript* $w \in \Sigma^* \cup \Sigma^\omega$ is to be folded by stabilizing beads of $w$ one at a time and minimizing energy collaboratively with the succeeding $\delta - 1$ nascent beads. The set $\mathcal{F}(\Xi)$ of conformations *foldable* by this system is defined recursively, as follows: the seed $C_\sigma$ is in $\mathcal{F}(\Xi)$; then provided that an elongation $C_i$ of $C_\sigma$ by the prefix $w[1:i]$ is foldable (i.e., $C_0 = C_\sigma$), its further elongation $C_{i+1}$ by the next bead $w[i+1]$ is foldable if

$$C_{i+1} \in \operatorname*{argmin}_{C \in \mathcal{E}_\alpha(C_1, x[1])} \min\left\{\Delta G(C') \mid C' \in \mathcal{E}_\alpha(C, x[2,d]))\right\}, \tag{1}$$

where $\Delta G(C')$ is an energy function that assigns to $C'$ with the negation of the number of h-interactions within $C'$ as energy. Informally speaking, $C_2$ is a conformation obtained by elongating $C_1$ by the bead $x[1]$ such that the beads $x[1], x[2], \ldots, x[d]$ create as many h-interactions as possible. Then, we write $C_1 \xhookrightarrow{\Xi}_x C_2$, and the superscript $\Xi$ is omitted whenever $\Xi$ is clear from the context. Through the folding, the first bead of $x$ is *stabilized*.

A conformation foldable by $\Xi$ is *terminal* if none of its elongations is foldable by $\Xi$. We use $C = [(P_\sigma P, w_\sigma w, H_\sigma \cup H)]$ to denote a terminal conformation, where $w$ is folded along the path $P$ with interactions in $H$. An OS is *deterministic* if, for all $i$, there exists at most one $C_{i+1}$ that satisfies Equation (1). In other words, a deterministic OS folds into a unique terminal conformation.

Fig. 3 illustrates an example of an OS with delay 2, arity 2, and the ruleset $\{(a,b), (b,f), (d,f), (d,e)\}$; in (a), the system tries to stabilize the first bead $a$ of the transcript, and the elongation in (a) gives one interaction. However, it is not the most stable because the elongation in (b) gives two interactions in total. Thus, the first bead $a$ is stabilized according to the location in (b). In (c), the system tries to stabilize the second bead $f$, and the elongation in (c) gives one interaction for the primary structure $fe$. However, the elongation in (d) gives two interactions in total. Thus, the second bead $f$ is stabilized according to the location in (d). Note that $f$ is not stabilized according to the location in (b), although the elongation in (b) is used to stabilize the first bead $a$.

Conformations $C_1$ and $C_2$ are *isomorphic* if there exist an instance $(P_1, w_1, H_1)$ of $C_1$ and an instance $(P_2, w_2, H_2)$ of $C_2$, such that $P_1 = P_2$ and $H_1 = H_2$. For two sets $\mathcal{C}_1$ and $\mathcal{C}_2$ of conformations, we say that two sets are isomorphic if there exists a one-to-one mapping $C_1 \in \mathcal{C}_1 \to C_2 \in \mathcal{C}_2$, such that $C_1$ and $C_2$ are isomorphic. We say that two oritatami systems are isomorphic if they fold the isomorphic set of foldable terminal conformations. A rule $(a,b)$ is *useful* in an OS $\Xi = (\Sigma, w, \mathcal{H}, \delta, \alpha, C_\sigma)$ if $\Xi' = (\Sigma, w, \mathcal{H} \setminus \{(a,b)\}, \delta, \alpha, C_\sigma)$ does not fold the same set of terminal conformations as $\Xi$.

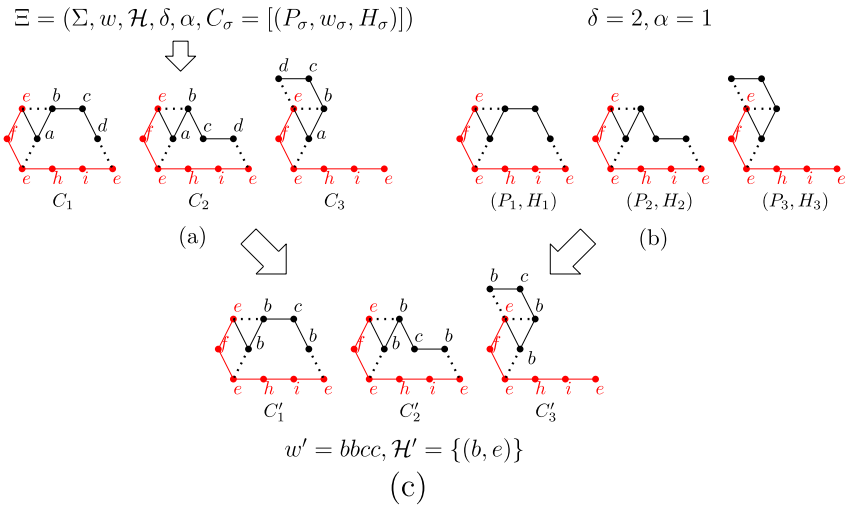


**Fig. 4.** An Illustration of two representations of Problem 1. The seed is colored in red.

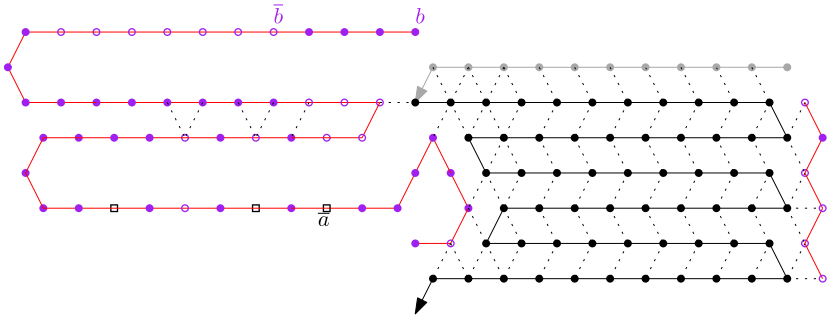


**Fig. 5.** An Illustration of a block for $C_i = (v_1 \vee v_2 \vee v_3)$ when $m = 4$. The path from the previous block is colored in gray.

## 3. Hardness of ruleset optimization on isomorphic oritatami systems

We first define the Ruleset Optimization problem on isomorphic OSs.

**Problem 1** *(Ruleset Optimization). Given an OS* $\Xi = (\Sigma, w, \mathcal{H}, \delta, \alpha, C_\sigma = [(P_\sigma, w_\sigma, H_\sigma)])$*, find an isomorphic OS* $\Xi' = (\Sigma', w', \mathcal{H}', \delta, \alpha, C'_\sigma = [(P_\sigma, w_\sigma, H_\sigma)])$*, where* $|\mathcal{H}'|$ *is minimum. (See Fig. 4 (a) and (c).)*

We can think of the problem as follows: Suppose we are given a delay $\delta$, an arity $\alpha$, a path $P_\sigma$, a set $H_\sigma$ of interactions, and a set $\{(P_i, H_i)\}$, where $P_i$ is a path and $H_i$ is a set of interactions on $P_i$. Then, the problem is to find a transcript $w'$ and a smallest ruleset $\mathcal{H}'$, where the OS $\Xi' = (\Sigma', w', \mathcal{H}', \delta, \alpha, C'_\sigma = [(P_\sigma, w_\sigma, H_\sigma)])$ successfully folds the set $\{[(P_\sigma P_i, w_\sigma w', H_\sigma \cup H_i)]\}$ of terminal conformations. (See Fig. 4 (b) and (c).)

**Theorem 2.** *The Ruleset Optimization problem is NP-hard.*

**Proof.** We reduce the problem from 1-IN-3-SAT problem. The problem is similar to 3SAT problem, except that every clause has only positive literals and exactly one literal on each clause should be satisfied. Suppose a formula $\phi = \bigwedge_{1 \le i \le n}(u_{(i,1)} \vee u_{(i,2)} \vee u_{(i,3)})$ is given, where $u_{(i,k)} \in \{v_j \mid 1 \le j \le m\}$. We construct a Ruleset Optimization problem instance from the formula. We set $\Sigma = \{a, \overline{a}, b, \overline{b}\}$, $\delta = 2m + 3$ and $\alpha = 4$. The seed $C_\sigma$, the path $P$ and the set $H$ of interactions consist of repetition of $n$ blocks, where each block represents one clause in $\phi$.

Fig. 5 shows one block for $C_i = (v_1 \vee v_2 \vee v_3)$ when $m = 4$. The path for a partial transcript consists of three zigzags, colored in black. Two parts of the seed surround the path from the left and the right. On the left of the zigzags, there exist two tunnel-like structures which allow only the straight path to fold. Fig. 6 (a) shows the block for $C_1 = (v_1 \vee v_2 \vee v_3)$ when $m = 4$. Since there is no previous block, the seed guides the zig-transcription. Fig. 6 (b) shows the block for $C_n = (v_1 \vee v_2 \vee v_3)$ when $m = 4$.

Now, we claim that the Ruleset Optimization problem instance has a solution of $|\mathcal{H}'| = 2$ if and only if the formula satisfies the 1-IN-3-SAT problem. First, suppose that the formula satisfies the problem. We use $\mathcal{H}' = \{(a, \overline{a}), (b, \overline{b})\}$. For the transcript $w'$, we claim that there exists a transcript that folds into the target conformation. Note that the ruleset is

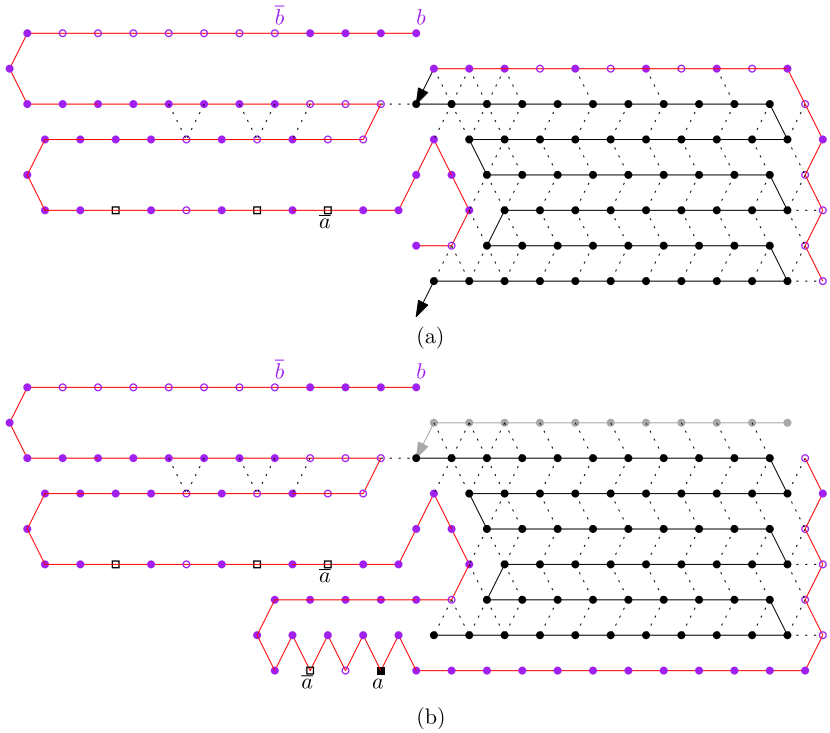


**Fig. 6.** (a) An Illustration of a block for $C_1 = (v_1 \vee v_2 \vee v_3)$ when $m = 4$. (b) An Illustration of a block for $C_n = (v_1 \vee v_2 \vee v_3)$ when $m = 4$.

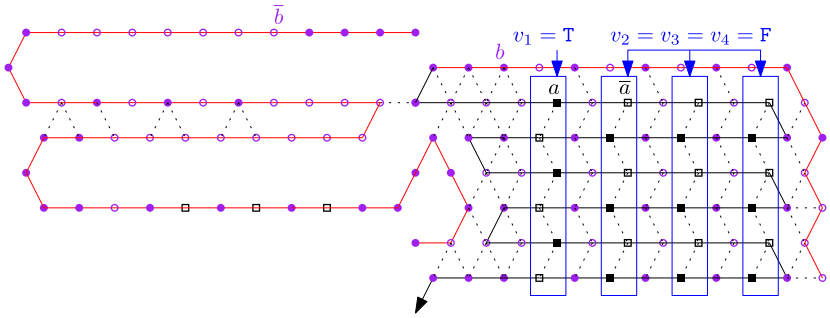


**Fig. 7.** Partially determined bead types for the first block.

complementary—for each bead type, there exists an unique bead type that can interact with. From Fig. 5 and Fig. 6, there are beads that are connected by a sequence of interactions with a bead on the seed. These beads are immediately determined as in Fig. 7 and Fig. 8. In the first zig of Fig. 7, we have $m$ beads that are both adjacent to $b$ and $\overline{b}$ and do not have interactions with either. This implies that one of $a$ or $\overline{a}$ should be assigned for these beads, which refers to a possible assignment of truth values for $v_i$'s. We regard bead type $a$ as $\mathtt{T}$ and $\overline{a}$ as $\mathtt{F}$. Then, an assignment in Fig. 7 represents that $v_1 = \mathtt{T}$ and $v_2 = v_3 = v_4 = \mathtt{F}$. The assignment at the first zig of the first block is propagated downward by the complementary rule $(a, \overline{a})$.

Now, when we try to stabilize the first bead of the first zig, there are three major elongations that we can use.

1. Proceed to the upper left tunnel.
2. Proceed to the lower left tunnel.
3. Follow the path of the terminal conformation.

Given the bead type assignment so far, the number of interactions for the first and the third elongation is $2m + 5$. Since the system has only one terminal conformation and should be deterministic, the second elongation, which stabilizes the first bead following the terminal conformation, should have at least $2m + 6$ interactions, as in Fig. 9.

Once the first bead is stabilized following the terminal conformation, to stabilize the second bead, we may extend the second and the third elongations. Since the location of the second bead is different in two elongations, the extension of the third elongation should be the most powerful one. Since the number of interactions is $2m + 6$, the extension of the second elongation should have at most $2m + 5$ interactions. For the second elongation, $\overline{a}$ is adjacent to the sequence of beads on the

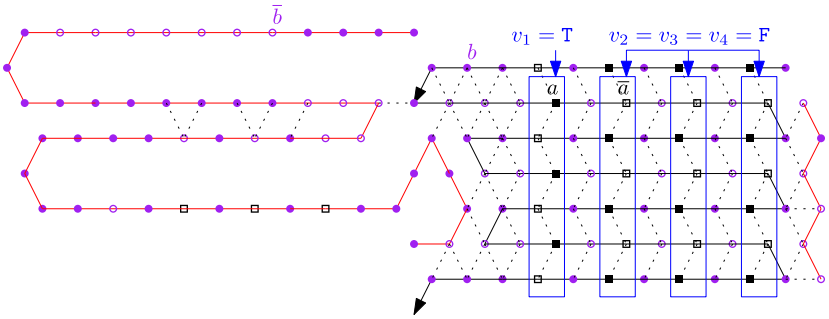


**Fig. 8.** Partially determined bead types for a block.

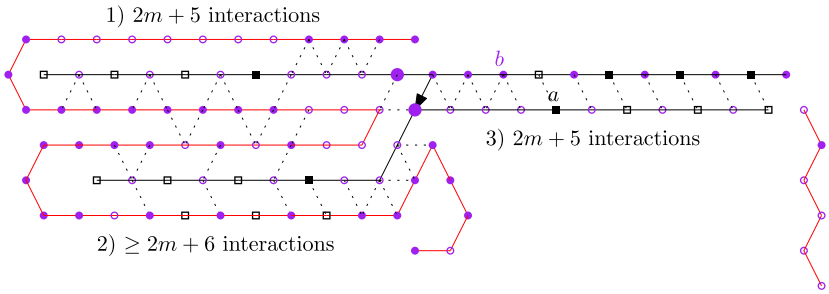


**Fig. 9.** Stabilizing the first bead of the block.

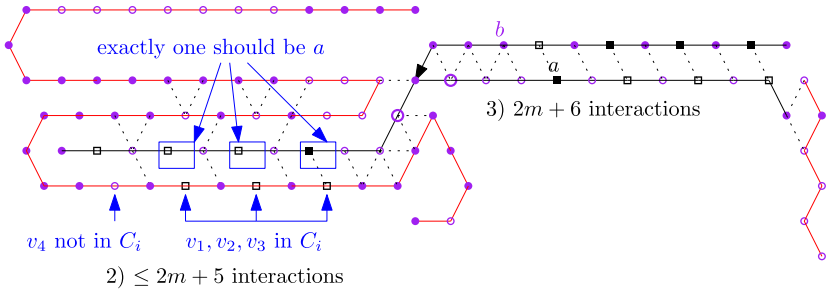


**Fig. 10.** Stabilizing the second bead of the block.

seed that represent truth value assignments for variable in the clause $C_i$. It turns out that the previous two conditions about the number of interactions can be satisfied if and only if exactly one bead on the elongation interacts with a bead $\overline{a}$ on the seed, which implies that only one variable in the clause is true from the current assignment. (See Fig. 10.) Thus, if there exists a truth value assignment that satisfies all clauses, then there exists a transcript that can fold the target conformation uniquely.

Second, suppose there exists a ruleset $\mathcal{H}'$ of size 2 and a transcript $w'$ such that the OS $\Xi' = (\Sigma, w', \mathcal{H}', \delta, \alpha, C'_\sigma = [(P_\sigma, w_\sigma, H_\sigma)])$ successfully folds the target conformation. From the adjacent bead pairs in the seed, we have the following observations about $\mathcal{H}'$:

- $(b, \overline{b}) \in \mathcal{H}'$.
- $(b, b), (\overline{b}, \overline{b}), (a, b), (a, \overline{b}), (\overline{a}, b), (\overline{a}, \overline{b}) \notin \mathcal{H}'$.

Since $(b, \overline{b}) \in \mathcal{H}'$, bead type assignments in Fig. 7 and Fig. 8 hold for $w'$. Same as before, we may assign $a$ or $\overline{a}$ for $m$ beads in the first zig of Fig. 7. Since $|\mathcal{H}'| = 2$, we may have only one of three rules $\{(a, a), (a, \overline{a}), (\overline{a}, \overline{a})\}$ in $\mathcal{H}'$.

1. If $(a, a) \in \mathcal{H}'$, then all beads in the first zig of Fig. 7 are $a$, since these beads have interactions with following beads. Then, while stabilizing the first bead of the block, all three elongations in Fig. 9 yields the same $2m + 5$ interactions, which leads to nondeterministic conformations.
2. If $(\overline{a}, \overline{a}) \in \mathcal{H}'$, then all beads in the first zig of Fig. 7 are $\overline{a}$. Then, while stabilizing the second bead of the block, the elongation to the lower left tunnel has $2m+7$ interactions and becomes the most stable one, which is not a target path.
3. If $(a, \overline{a}) \in \mathcal{H}'$, then we may assign $a$ or $\overline{a}$ beads in the first zig of Fig. 7. As previously described, if the transcript folds into the target conformation, then only one variable in the clause is true from the current assignment. Thus, if there

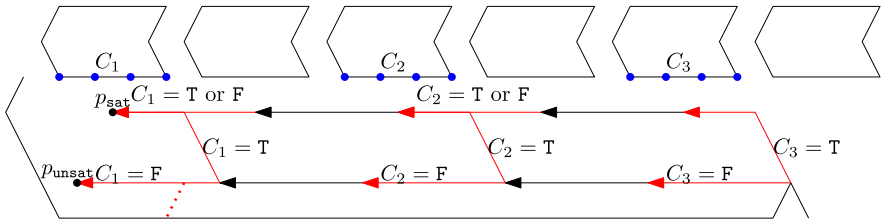


**Fig. 11.** A part of a DNF tautology checking OS by Han et al. [11]. In the figure, the formula has three clauses $C_1, C_2$ and $C_3$, and their values are evaluated. The last part of the transcript has periodic probes (colored in red) that scan the sequence of beads (colored in blue) and changes the path if it encodes TRUE. Interactions in the terminal conformation are omitted. The dotted line represents an interaction from a rule $r_1$ that attracts the last probe downward when $C_1$ is evaluated to FALSE.

exists a transcript that can fold the target conformation uniquely, then there exists an truth value assignment that satisfies all clauses. □

## 4. Ruleset reduction by bead type merging

Since the Ruleset Optimization problem is NP-hard in general, we consider poly-time heuristics to reduce the size of a ruleset efficiently. Because not all rules in a ruleset are useful, we start with removing useless rules. Note that some rules can be useful but not visible in the terminal conformation, even when the system is deterministic. For a deterministic OS, it is sufficient to simulate the OS and find the useless rules. The simulation takes $O(n \cdot 5^{\delta})$ time, where $n$ is the length of the transcript.

**Corollary 3.** *For a deterministic OS* $\Xi = (\Sigma, w, \mathcal{H}, \delta, \alpha, C_\sigma = [(P_\sigma, w_\sigma, H_\sigma)])$*, we can remove useless rules in* $O(n \cdot 5^{\delta})$ *time, where* $n = |w|$.

For a nondeterministic OS, we show the hardness of the problem.

**Theorem 4.** *For a nondeterministic OS* $\Xi = (\Sigma, w, \mathcal{H}, \delta, \alpha, C_\sigma = [(P_\sigma, w_\sigma, H_\sigma)])$ *and a rule* $r \in \mathcal{H}$*, it is coNP-hard to determine if* $r$ *is useless.*

**Proof.** We reduce the problem from the DNF tautology problem. A DNF formula $\phi$ is written as $\bigvee_{1 \le i \le n} C_i$ for clauses $C_1, \ldots, C_n$, each of which is a logical AND ($\wedge$) of Boolean variables $v_1, \ldots, v_m$ and their negations. The *DNF tautology problem* asks whether or not a given DNF formula is TRUE on all possible assignments. This problem is coNP-complete [16]. We use the tautology checking OS by Han et al. [11], which nondeterministically assigns values to variables and evaluates the formula by the location of the last bead. At the last part of the terminal conformation, the folding starts from the lower row. For each clause, the transcript probes a sequence of beads that represents the value of the clause. If the clause is evaluated TRUE, the transcript changes its path to the upper row, as depicted in Fig. 11. If there exists at least one TRUE evaluation, the last bead stabilizes at the point $p_{\text{sat}}$ at the upper row, and $p_{\text{unsat}}$ at the lower row otherwise. The probe part is periodic and repeats $n$ times, according to the number of clauses.

Now, assume that we use different, distinct bead types for the last probe, which interacts in the same way as the original bead types. There exists a rule $r_1$ that attracts the bead in the last probe downward when $C_1$ is evaluated to FALSE, as shown in Fig. 11. If the tautology problem is true, then the formula is always evaluated to TRUE, and there is no chance that the last bead stabilizes at $p_{\text{unsat}}$. Thus, the last probe cannot reach $p_{\text{unsat}}$ and $r_1$ becomes useless. On the other hand, if the tautology problem is false, then there exists a terminal conformation where the last bead is stabilized at $p_{\text{unsat}}$, which means that $r_1$ is useful. Therefore, the problem becomes coNP-hard. □

It is coNP-hard to identify and remove useless rules in general. Thus, we propose a method to reduce the ruleset size regardless of usefulness of rules. For two bead types $a$ and $b$, suppose $(a, c) \in \mathcal{H}$ if and only if $(b, c) \in \mathcal{H}$ for all possible bead types $c$. If we merge beads $a$ and $b$ and replace all $b$'s in the transcript and the seed to $a$'s, it is straightforward to verify that the resulting OS is isomorphic to the original OS. We formally define the problem of finding a smallest ruleset based on the bead type merging.

**Problem 5** *(Ruleset Optimization by Bead Type Merging). Given a ruleset* $\mathcal{H} \subseteq \Sigma \times \Sigma$ *of an OS, find a minimum alphabet* $\Sigma'$ *and a ruleset* $\mathcal{H}' \subseteq \Sigma' \times \Sigma'$*, where there exists a homomorphism* $h : \Sigma \to \Sigma'$ *such that*

- *for each* $(x_i, x_j) \in \mathcal{H}$, $(h(x_i), h(x_j)) \in \mathcal{H}'$, *and*
- *for each* $(x_i, x_j) \notin \mathcal{H}$, $(h(x_i), h(x_j)) \notin \mathcal{H}'$.

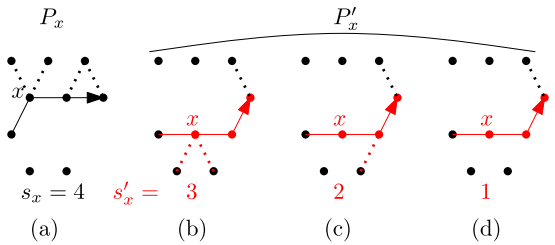


**Fig. 12.** When we stabilize the bead $x$, suppose the path $P_x$ in (a) is the most stable one with the sum of interactions $s_x = 4$. Since the path $P'_x$ in (b), (c) and (d) stabilizes $x$ at a location different from $P_x$, we may assign arbitrary interactions for these paths as long as the sum $s'_x$ of interactions does not exceed 4.

**Theorem 6.** *We can solve the Ruleset Optimization by Bead Type Merging problem in $O(t^2)$ time, using $O(t)$ space, where $t = |\Sigma|$.*

**Proof.** We may construct a binary string $x_i$ for each bead type $\sigma_i$, where $x_i[k]$ is 1 if $(\sigma_i, \sigma_k) \in \mathcal{H}$, and 0 otherwise. It is straightforward that if $x_i = x_j$, then $\sigma_i$ and $\sigma_j$ can be merged successfully and result in the same homomorphic bead type. We run a radix sort for strings $x_1, x_2, \ldots, x_t$, where $t = |\sigma|$. After the sorting, any set of bead types corresponding to the same (consecutive) string can be merged successfully. Since the length of the strings is $t$, the radix sort requires $O(t^2)$ time, using $O(t)$ space.

Suppose we have the new alphabet $\Sigma' \subseteq \Sigma$ after the merging by the radix sort. For each bead type pair $(x_i, x_j) \in {\Sigma'}^2$, there exists a bead type $x_k \in \Sigma'$ such that $(x_i, x_k) \in {\Sigma'}^2$ and $(x_j, x_k) \notin {\Sigma'}^2$. Thus, if there exists a minimum alphabet $\Sigma''$ smaller than $\Sigma'$ and the corresponding homomorphism $h'' : \Sigma \to \Sigma''$, there should exist a bead type triple $(x_i, x_j, x_k) \in \Sigma^3$ such that $h''(x_i) = h''(x_j)$, $(x_i, x_k) \in \mathcal{H}$, $(x_j, x_k) \notin \mathcal{H}$, and $(h''(x_i), h''(x_k)) \in \mathcal{H}$ if and only if $(h''(x_j), h''(x_k)) \in \mathcal{H}$, which is contradiction. Thus, we know that $\Sigma'$ is minimal. □

## 5. Ruleset reduction by fuzzy ruleset construction

The bead-type merging only uses information from the ruleset, not the whole OS. Note that we can remove rules from or add rules to the ruleset while maintaining an OS as isomorphic. Thus, we propose another, more efficient heuristic that finds a reduced ruleset from a set of rulesets for an isomorphic OS.

Given an alphabet $\Sigma$, we define a *fuzzy ruleset* to be a pair of a required ruleset $\mathcal{H}_P \subseteq \Sigma \times \Sigma$, and a forbidden ruleset $\mathcal{H}_N \subseteq \Sigma \times \Sigma$ such that $\mathcal{H}_P \cap \mathcal{H}_N = \emptyset$. Given an OS $\Xi = (\Sigma, w, \mathcal{H}, \delta, \alpha, C_\sigma)$, we say that a fuzzy ruleset $(\mathcal{H}_P, \mathcal{H}_N)$ is a *representative fuzzy ruleset* of the OS if $\Xi' = (\Sigma, w, \mathcal{H}', \delta, \alpha, C_\sigma)$ is isomorphic to $\Xi$ for all $\mathcal{H}'$ satisfying the following conditions:

1. If $(a, b) \in \mathcal{H}_P$, then $(a, b) \in \mathcal{H}'$.
2. If $(a, b) \notin \mathcal{H}_N$, then $(a, b) \notin \mathcal{H}'$.

We say that such $\mathcal{H}'$ is in the representative fuzzy ruleset $(\mathcal{H}_P, \mathcal{H}_N)$. In other words, if a fuzzy ruleset $(\mathcal{H}_P, \mathcal{H}_N)$ is representative, then rules in $\mathcal{H}_P$ should be included in the ruleset, and rules in $\mathcal{H}_N$ should be excluded from the ruleset, which ensures that the system is isomorphic to the original system. These conditions are obtained by the property of the cotranscriptional folding. When we want to design an isomorphic system, we should keep the same location of the stabilized beads and the same interactions. While stabilizing a bead $x$, the bead and its $\delta - 1$ nascent beads choose a path $P_x$ that maximizes the sum $s_x$ of interactions. Then, for any alternative path $P'_x$, where the bead is not stabilized at the target location, we can arbitrarily assign interactions for $P'_x$, as long as the sum $s'_x$ of interactions does not exceed $s_x$, as illustrated in Fig. 12.

We reduce the ruleset size in two phases: 1) given an OS $\Xi$, we extract a representative fuzzy ruleset from $\Xi$; 2) we propose a graph representation of the representative fuzzy ruleset and reduce the ruleset size based on the graph theory approach.

**Problem 7** *(Fuzzy Ruleset Optimization). Given an OS $\Xi = (\Sigma, w, \mathcal{H}, \delta, \alpha, C_\sigma = [(P_\sigma, w_\sigma, H_\sigma)])$, find a representative fuzzy ruleset $(\mathcal{H}_P, \mathcal{H}_N)$ minimizing $|\mathcal{H}_P| + |\mathcal{H}_N|$.*

**Theorem 8.** *The Fuzzy Ruleset Optimization problem is NP-hard.*

**Proof.** We use the same reduction used in the proof of Theorem 2, except the last block shown in Fig. 13. We claim that the 1-IN-3-SAT problem instance has a solution if and only if there exists a representative fuzzy ruleset where $|\mathcal{H}_P| + |\mathcal{H}_N| = 10$. If the 1-IN-3-SAT problem instance has a solution, we may use $\mathcal{H}_P = \{(a, \overline{a}), (b, \overline{b})\}$ and $\mathcal{H}_N = \{(a, a), (b, b), (\overline{a}, \overline{a}), (\overline{a}, \overline{b}), (a, b), (a, \overline{b}), (\overline{a}, b), (\overline{a}, \overline{b})\}$. This fuzzy ruleset represents one ruleset that was used in the proof of Theorem 2, which folds to the target conformation if the 1-IN-3-SAT problem instance has a solution. Now, suppose there

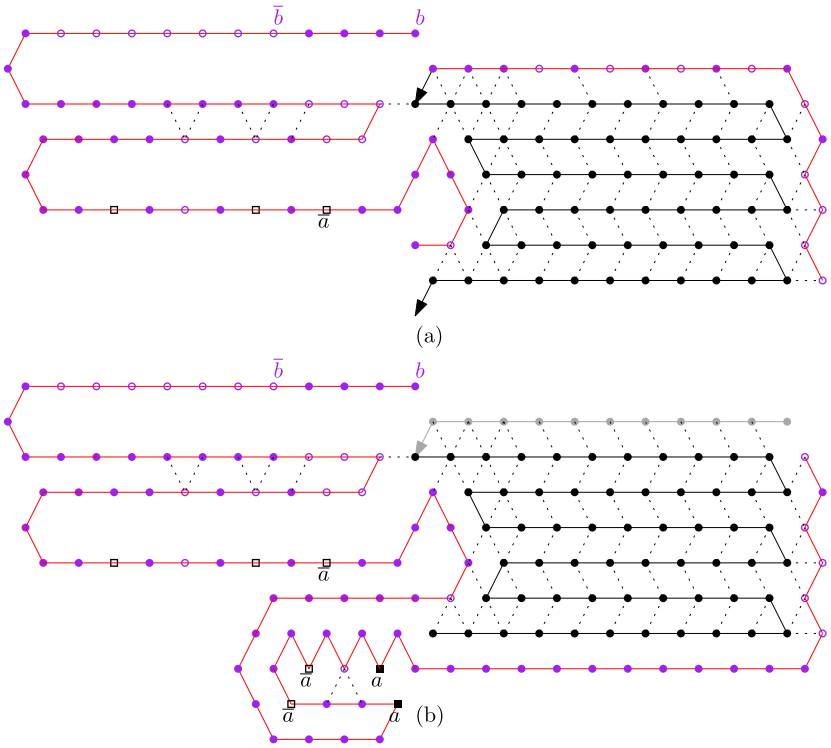


**Fig. 13.** The modified block for $C_n = (v_1 \vee v_2 \vee v_3)$ when $m = 4$.

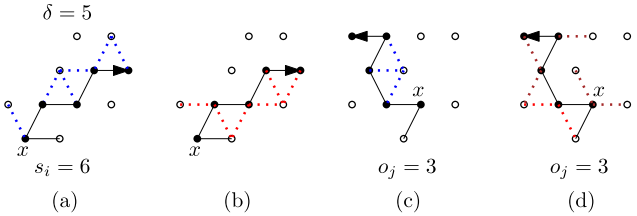


**Fig. 14.** An illustration of Algorithm 1. Beads in the elongation are represented by disks, and beads already stabilized are represented by circles. Suppose the delay of the system is 5 and the path in (a) is chosen to stabilize the first bead $x$, with the sum of interactions $s_i = 6$. Rules for blue interactions in (a) are added to $P$. In (b), interactions that should not exist are colored in red, and added to $N$. In (c), one path that is not chosen is illustrated. This path has the basic strength of $o_j = 3$ from rules in $P$. In (d), two interactions that should not exist (those in $N$) are colored in red. The set of all 5 possible interactions that are not from $P$ are colored in brown. Rules for brown interactions are added to $K$, and $(K, 3)$ is added to $\mathcal{H}_c$, indicating that we can add no more than three rules for brown interactions to the ruleset.

exists a representative fuzzy ruleset where $|\mathcal{H}_P| + |\mathcal{H}_N| = 10$. The seed assigns nine rules explicitly to $\mathcal{H}_P$ and $\mathcal{H}_N$, except one rule $(a, \overline{a})$. The rule $(a, \overline{a})$ is necessary for interactions between the seed and the given path of the primary structure, so $(a, \overline{a}) \notin \mathcal{H}_N$. If $(a, \overline{a}) \in \mathcal{H}_P$, then the 1-IN-3-SAT problem instance has a solution, following the proof of Theorem 2. □

Next, we design a heuristic algorithm for the Fuzzy Ruleset Optimization problem. Assume that an OS $\Xi$ folds the set $\{C_i = [(P_\sigma P_i, w_\sigma w, H_\sigma \cup H_i)]\}$ of $t$ terminal conformations. We assume that $|w| = n$, and $|w_\sigma|$ and $|H_i|$ are bounded to $O(n)$. We first propose an algorithm for one terminal conformation $C_1$, and then apply the algorithm for all terminal conformations. Given that $\Xi$ folds $C_1$, we find conditions of the rules, which are necessary and sufficient for an isomorphic OS. Then, we construct a representative fuzzy ruleset from these conditions.

Let $P_1 = p_1 p_2 \cdots p_n$ and $P_\sigma = p_{n+1} p_{n+2} \cdots p_{n+m}$. We run Algorithm 1, which returns three conditions that are necessary and sufficient for an isomorphic OS. The required condition set $P$ (the forbidden condition set $N$) includes the set of rules that should be included in (excluded from) the desired ruleset $\mathcal{H}$. Later, the construction of a representative fuzzy ruleset $(\mathcal{H}_P, \mathcal{H}_N)$ starts from $(P, N)$. The last output is the conditional ruleset $\mathcal{H}_c = \{(K \subset \Sigma \times \Sigma, s)\}$. The element $(K, s) \in \mathcal{H}_c$ implies that the number of rules in $K \cap \mathcal{H}'$ should not exceed $s$, where $\mathcal{H}'$ is in $(\mathcal{H}_P, \mathcal{H}_N)$. The conditional ruleset has information of rules that are not explicitly shown in the most stable elongation, but that prevent the path from not following $P_1$. Fig. 14 illustrates Algorithm 1.

**Lemma 9.** *Algorithm 1 runs in $O(5^\delta \delta n)$ time using $O(5^\delta \delta n)$ space.*

**Algorithm 1:** ExtractConditionSets.

**Input**: An OS $\Xi = (\Sigma, w, \mathcal{H}, \delta, \alpha, C_\sigma = [(P_\sigma, w_\sigma, H_\sigma)])$, the terminal conformation $C_1 = [(P_1, w, H_1)]$
**Output**: A required condition set $P$, a forbidden condition set $N$, and a conditional ruleset $\mathcal{H}_c$

```
1  for i ← 1 to n do
2      κ_i ← w_σ[i]
3  for i ← n+1 to n+m do
4      κ_i ← w[i−n]
5      calculate the sum s_i of the interactions that led kappa_i to the position p_i.
6      for each annotated neighbors p_j of p_i do
7          if {p_i, p_j} ∈ H_1 then add (κ_i, κ_j) to P.
8          else add (κ_i, κ_j) to N.
9      for each unannotated path P' = p'_1 p'_2 ··· p'_δ where p'_1 ≠ p_i is an unannotated neighbor of p_{i−1} do
10         o_j ← 0, K ← ∅
11         for j ← 1 to δ do
12             for each annotated neighbors p_k of p'_j where p_k has interactions less than α do
13                 if (κ_{i+j−1}, κ_k) ∈ P then o_j ← o_j + 1
14                 else
15                     if s_i = 1 then add (κ_{i+j−1}, κ_k) to N.
16                     else add (κ_{i+j−1}, κ_k) to K.
17         if s_i ≠ 1 then add (K, s_i − o_j − 1) to H_c.
18 return P, N, H_c
```

**Proof.** It takes $O(\delta)$ time to check $s_i$ in Line 5 and $O(1)$ time to check annotated neighbors of $p_i$ in Lines 6 to 8. In Line 9, there are $O(5^\delta)$ unannotated paths. For each path, calculating $o_j$ in Line 13 takes $O(\delta)$, adding pairs to $N$ in Line 15 takes $O(\delta)$, and adding elements to $C$ in Line 16 takes $O(\delta)$. Therefore, the total runtime is $O(n \cdot 5^\delta \delta)$. For the space requirement, the size of $P$ is $O(n)$, the size of $N$ is $O(n+\delta)$, and the space requirement for $\mathcal{H}$ is $O(5^\delta \delta n)$. Therefore, the space complexity is $O(5^\delta \delta n)$. □

Since the conditions in $\mathcal{H}_c$ are about the rules that are not explicitly shown in the most stable elongation, there is no necessary rule that should be added to $P$, because of $\mathcal{H}_c$. Thus, we construct a representative fuzzy ruleset $(P, \mathcal{H}_N)$, where $\mathcal{H}_N = N \cup N_{add}$ and $N_{add}$ satisfies the conditions in $\mathcal{H}_c$. We prove that minimizing $|\mathcal{H}_N|$ is NP-complete.

**Lemma 10.** *Given a set $\mathcal{H}_c \subseteq 2^{\Sigma\times\Sigma} \times \mathbb{N}$, let $N_{add} \subseteq \Sigma \times \Sigma$ be a set such that, for all $(K_i, s_i) \in \mathcal{H}_c$, $|K_i| - |K_i \cap N_{add}| < s_i$ holds. Then, it is NP-complete to decide whether $|N_{add}| < k$ for given $k$.*

**Proof.** Once $N_{add}$ with $|N_{add}| < k$ is given, we can check whether or not $N_{add}$ satisfies the condition in polynomial time. Therefore, the problem is NP.

Next, we prove that the problem is NP-hard. We reduce the set cover problem to the proposed problem. Suppose that a universe $\mathcal{U} = \{1, 2, \ldots, n\}$ and a set $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ of subsets of $\mathcal{U}$ are given. Then, we construct a set $\mathcal{H}_c$ as follows: For each $i \in \mathcal{U}$, $(K_i, s_i) \in \mathcal{H}_c$, where $K_i = \{(\kappa_{j1}, \kappa_{j2})\}$ for all $S_j \in \mathcal{S}$ that contains $i$, and $s_i = |K_i|$. For a cover $\mathcal{C} \subseteq \mathcal{S}$, we construct $N_{add} = \{(\kappa_{j1}, \kappa_{j2})\}$ for all $S_j \in \mathcal{C}$. Now, suppose that a cover $\mathcal{C}$ satisfies the set cover problem. Then, for all $i \in \mathcal{U}$, there exists $j$ such that $S_j \in \mathcal{C}$, which implies that $|K_i| - |K_i \cap N_{add}| < |K_i| = s_i$. Therefore, $N_{add}$ satisfies the given condition in the problem. For the other direction, suppose that $N_{add}$ satisfies the condition in the problem. Then, for all $i \in \mathcal{U}$, there exists $j$ such that $(\kappa_{j1}, \kappa_{j2}) \in K_i$, which implies that $i \in S_j \in \mathcal{C}$. Therefore, the problem is NP-hard. □

Since finding the minimum $N_{add}$ is NP-complete, we suggest three heuristics to create $N_{add}$. We assume that a condition $(K_i, s_i)$ is in $\mathcal{H}_c$.

1. While adding pairs to $N$ to satisfy conditions in $\mathcal{H}_c$, we add as few pairs as possible, since more pairs in a negative condition set makes reduction harder.
2. We prefer $(K_i, s_i)$ with the largest $s_i$, since we need to add more pairs to satisfy that condition.
3. For a pair $(\kappa_j, \kappa_k) \in K_i$, we prefer a pair with the most frequent appearances in all $K_i$'s, since adding the pair $(\kappa_j, \kappa_k)$ helps satisfy all these conditions.

Based on these heuristics, we run Algorithm 2.

**Lemma 11.** *Algorithm 2 runs in $O(5^\delta \delta n(\delta + \log n))$ time using $O(5^\delta \delta n)$ space.*

**Proof.** The size of $\mathcal{H}_c$ is $O(5^\delta n)$, where each $K_i$ has $O(\delta)$ pairs. It takes $O(5^\delta \delta n \log n)$ time to sort $\mathcal{H}_c$ according to $s_i$. Note that for $(\kappa_j, \kappa_k) \in K_i$, either $\kappa_j$ or $\kappa_k$ is from a point in a path $P_1$. Thus, if we maintain links between all $K_i$'s from

**Algorithm 2:** ExtractFuzzyRuleset.

```
Input: A conditional ruleset H_c
Output: A set N_add
while H_c ≠ ∅ do
    for each (K_i, s_i) with the largest s_i do
        count the number occ_(j,k) of appearances of (κ_j, κ_k) in all K_i's.
    for each (K_i, s_i) with the largest s_i do
        while the condition does not hold do
            find a pair (κ_j, κ_k) of bead types with the biggest occ_(j,k).
            add (κ_j, κ_k) to N_add.
        delete (K_i, s_i) from H_c.
return N_add
```

same point, it requires $O(5^\delta \delta^2 n)$ time to calculate the number of occurrences $occ_{(j,k)}$. Finding a pair of bead types with the biggest $occ_{(j,k)}$ takes $O(5^\delta \delta n \log n)$ time in total. Therefore, the total runtime is $O(5^\delta \delta n(\delta + \log n))$. Since the size of $N_{add}$ is $O(5^\delta \delta n)$, the space requirement is $O(5^\delta \delta n)$. □

Once we have a representative fuzzy ruleset $(\mathcal{H}_P, \mathcal{H}_N)$, the next step is to construct a reduced ruleset that satisfies the conditions of the fuzzy ruleset. We construct a *fuzzy ruleset graph* from $(\mathcal{H}_P, \mathcal{H}_N)$ by adding positive edges for rules in $\mathcal{H}_P$ and negative edges for rules in $\mathcal{H}_N$.

- $V = \Sigma$
- For each pair of molecules $(x_1, x_2) \in \Sigma \times \Sigma$,
  - add $(\{x_1, x_2\}, 1)$ to $E$ if $(x_1, x_2) \in \mathcal{H}_P$,
  - add $(\{x_1, x_2\}, -1)$ to $E$ if $(x_1, x_2) \in \mathcal{H}_N$.

**Problem 12** *(Fuzzy Ruleset Optimization by Bead Type Merging). Given a representative fuzzy ruleset $(\mathcal{H}_P, \mathcal{H}_N)$ of an OS over an alphabet $\Sigma$, find a minimum alphabet $\Sigma'$ and a ruleset $\mathcal{H}' \subseteq \Sigma' \times \Sigma'$, where there exists a homomorphism $h : \Sigma \to \Sigma'$ such that for every $(x_1, x_2) \in \Sigma \times \Sigma$, $((x_1, x_2) \in \mathcal{H}_P \wedge (x_1, x_2) \notin \mathcal{H}_N) \Leftrightarrow (h(x_1), h(x_2)) \in \mathcal{H}'$.*

**Lemma 13.** *The Fuzzy Ruleset Optimization by Bead Type Merging problem is NP-complete.*

**Proof.** Note that there exists a one-to-one correspondence between a fuzzy ruleset and a fuzzy ruleset graph. We consider the problem of finding a ruleset that is reduced from the ruleset yielded by $G$, with an alphabet size less than $k$. First, once a reduced fuzzy graph $G' = (V', E')$ with $|V'| < k$ is given, we can check whether or not $G'$ can be reduced from $G$ in polynomial time. Therefore, the problem is NP.

Next, we prove that the problem is NP-hard. We reduce the vertex coloring problem to the given problem. For a graph $G_c = (V_c, E_c)$, we construct a fuzzy ruleset graph $G_r = (V_c, E_r)$ using the following rules:

- For all $v \in V_c$, $(\{v, v\}, -1) \in E_r$.
- For all $(v_i, v_j) \in E_c$, $(\{v_i, v_j\}, 1) \in E_r$.

Then, it is straightforward to verify that we can merge two nodes $v_i, v_j$ in $G_r$ if and only if they can be colored by the same color in $G_c$. Therefore, once we know the existence of a reduced graph $G' = (V', E')$ with $|V'| < k$, we can determine whether or not $G_c$ can be colored using fewer than $k$ colors. Therefore, the problem is NP-hard. □

Note that we reduce the vertex coloring problem to the Fuzzy Ruleset Optimization by Bead Type Merging problem. We formally establish the function $f$ from a fuzzy ruleset graph $G_r = (V, E_r)$ to a graph $G_c = (V, E_c)$ using the following rules: For all $(v_i, v_j) \in V^2$, $(v_i, v_j) \in E_c$ if and only if we cannot merge $v_i$ and $v_j$. It requires $O(n^3)$ to construct $f(G_r)$ from $G_r$, when $n = |V|$. We establish the following lemma.

**Lemma 14.** *For a fuzzy ruleset graph $G_r = (V, E_r)$, let $G_c = f(G_r)$. Let $v_1$ and $v_2$ be two mergeable nodes in $V$. Let $G'_r$ ($G'_c$) be the graph resulting from $G_r$ ($G_c$) after merging $v_1$ and $v_2$. Then, $G'_c = f(G'_r)$.*

**Proof.** Let $\{v_i, v_j\}$ be a pair of nodes in $V$, except $\{v_1, v_2\}$. We consider two cases separately.

1. There is no edge between $\{v_i, v_j\}$ and $\{v_1, v_2\}$ in $G_r$: It is immediate that merging $v_1$ and $v_2$ does not change whether or not $v_i$ and $v_j$ can be merged. Therefore, $(v_i, v_j)$ is an edge of $G'_c$ if and only if $v_i$ and $v_j$ can be merged in $G'_r$.

2. There is an edge between $\{v_i, v_j\}$ and $\{v_1, v_2\}$ in $G_r$: Without loss of generality, suppose there is an edge $(\{v_i, v_1\}, 1)$ in $G_r$. Since $v_1$ and $v_2$ are mergeable, $(\{v_i, v_2\}, -1) \notin E_r$. Therefore, $(\{v_i, v_1\}, 1) \in E'_r$. Then, $v_i$ and $v_j$ can be merged in $G'_r$ if and only if they can be merged in $G_r$, and $(v_i, v_j)$ is an edge of $G'_c$ if and only if $v_i$ and $v_j$ can be merged in $G'_r$. The same analysis holds for the case when $v_1$ has an edge with both $v_i$ and $v_j$, or both $v_1$ and $v_2$ have an edge with $v_i$ or $v_j$. □

From Lemma 14, we know that any solution to the vertex coloring problem has its pair solution to the Fuzzy Ruleset Optimization by Bead Type Merging problem. Therefore, we can use approximation algorithms for the vertex coloring problem to find approximate solutions for the Fuzzy Ruleset Optimization by Bead Type Merging problem. One such algorithm is the Welsh-Powell algorithm [17]. Once all vertices $v_i$ are ordered according to their degrees $d_i$, the algorithm runs in $O(n^2)$ time and gives at most $\max_i \min\{d_i + 1, i\}$ colors.

In summary, we have proposed the Fuzzy Ruleset Optimization problem, which is a variation of the Ruleset Optimization problem and NP-hard. We propose a heuristic to construct a representative fuzzy ruleset. We first extract the necessary and sufficient conditions of rules from the set of ruleset sizes using Algorithm 1. We accumulate $P$, $N$, and $\mathcal{H}_c$ by running Algorithm 1 for $1 \le i \le t$, and then run Algorithm 2 to construct a representative fuzzy ruleset. The Ruleset Optimization by Bead Type Merging problem, calculating the minimum ruleset from the given representative fuzzy ruleset by bead type merging, is also NP-complete. Thus, we construct a fuzzy ruleset graph from the representative fuzzy ruleset, and use an approximation algorithm for the vertex coloring problem to find an approximate solution for the Fuzzy Ruleset Optimization by Bead Type Merging problem. We propose a heuristic that can solve the Ruleset Optimization problem in $O(5^\delta \delta n(\delta + \log n + t) + n^3)$ time, using $O(5^\delta \delta n)$ space. Note that the bead-type modification for a given ruleset in Section 4 is a special case of the Fuzzy Ruleset Optimization by Bead Type Merging problem, where $\mathcal{H}_P \cup \mathcal{H}_N = \Sigma \times \Sigma$. Therefore, the method proposed in Section 5 is at least as efficient as bead-type merging in the size of the reduced ruleset.

## 6. Conclusions

The oritatami system (OS) is a computational model inspired by RNA cotranscriptional folding, where an RNA transcript folds upon itself while being synthesized out of a gene. One element of the OS is the ruleset, which defines interactions between beads in the system. It is crucial to reduce the ruleset size to implement a simpler OS in experiments. We first defined the concept of isomorphism of OSs. Then, we proved that it is NP-hard to find the smallest ruleset of an isomorphic OS, in general. We have proposed a bead-type merging method and a representative fuzzy ruleset construction to reduce the ruleset size.

There remain open questions. For example, it is necessary to find theoretical bounds of approximation ratios—the size of the resulting ruleset over the size of the optimal ruleset—of the proposed heuristic algorithms, and to design an efficient algorithm that removes useless rules. We can also consider a ruleset optimization for a given path without considering the set of interactions, and a transcript optimization for a given ruleset.

## References


[1] M. Zuker, P. Stiegler, Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information, Nucleic Acids Res. 9 (1) (1981) 133–148.
[2] E. Rivas, S.R. Eddy, A dynamic programming algorithm for RNA structure prediction including pseudoknots, J. Mol. Biol. 285 (5) (1999) 2053–2068.
[3] M. Zuker, Mfold web server for nucleic acid folding and hybridization prediction, Nucleic Acids Res. 31 (13) (2003) 3406–3415.
[4] A. Xayaphoummine, T. Bucher, H. Isambert, Kinefold web server for RNA/DNA folding path and structure prediction including pseudoknots and knots, Nucleic Acids Res. 33 (2005) W605–W610.
[5] C. Geary, P.W.K. Rothemund, E.S. Andersen, A single-stranded architecture for cotranscriptional folding of RNA nanostructures, Science 345 (2014) 799–804.
[6] K.L. Frieda, S.M. Block, Direct observations of cotranscriptional folding in an adenine riboswitch, Science 338 (6105) (2012) 397–400.
[7] D. Lai, J.R. Proctor, I.M. Meyer, On the importance of cotranscriptional RNA structure formation, RNA 19 (2013) 1461–1473.
[8] K.E. Watters, E.J. Strobel, A.M. Yu, J.T. Lis, J.B. Lucks, Cotranscriptional folding of a riboswitch at nucleotide resolution, Nat. Struct. Mol. Biol. 23 (12) (2016) 1124–1131.
[9] C. Geary, P. Meunier, N. Schabanel, S. Seki, Programming biomolecules that fold greedily during transcription, in: Proceedings of the 41st International Symposium on Mathematical Foundations of Computer Science, 2016, pp. 43:1–43:14.
[10] C. Geary, P. Meunier, N. Schabanel, S. Seki, Efficient universal computation by greedy molecular folding, CoRR abs/1508.00510, http://arxiv.org/abs/1508.00510.
[11] Y. Han, H. Kim, M. Ota, S. Seki, Nondeterministic seedless oritatami systems and hardness of testing their equivalence, Nat. Comput. 17 (1) (2018) 67–79.
[12] Y. Han, H. Kim, T.A. Rogers, S. Seki, Self-attraction removal from oritatami systems, in: Proceedings of the 19th International Conference on Descriptional Complexity of Formal Systems, 2017, pp. 164–176.
[13] J. Rogers, G.F. Joyce, A ribozyme that lacks cytidine, Nature 402 (6759) (1999) 323–325.
[14] M. Ota, S. Seki, Rule set design problems for oritatami system, Theor. Comput. Sci. 671 (2017) 16–35.
[15] A. Gibbons, Algorithmic Graph Theory, Cambridge University Press, 1985.
[16] J. Kleinberg, É. Tardos, Algorithm Design, Addison-Wesley, 2011.
[17] K.H. Rosen, Discrete Mathematics and Its Applications, McGraw-Hill Education, 2006.