Ruleset Optimization on Isomorphic Oritatami Systems

Yo-Sub Han and Hwee Kim

Department of Computer Science, Yonsei University
50 Yonsei-Ro, Seodaemun-Gu, Seoul 03722, Republic of Korea
{emmous, kimhwee}@yonsei.ac.kr

Abstract. RNA cotranscriptional folding refers to the phenomenon in which an RNA transcript folds upon itself while being synthesized out of a gene. The oritatami system (OS) is a computation model of this phenomenon, which 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 maintaining the terminal conformations geometrically same. We first prove the hardness of finding the smallest ruleset, and suggest two approaches that reduce the ruleset size efficiently.

1 Introduction

In nature, a one-dimensional RNA sequence folds itself autonomously and gives rise to a highly-dimensional tertiary structure. It has been a challenging question to predict the tertiary structure from a primary structure. Recently, biochemists showed that the kinetics plays an essential role in the geometric shape of the RNA foldings [1], since the folding caused by intermolecular reactions is faster than the RNA transcription rate [4]. By controlling cotranscriptional foldings, researchers succeeded in assembling a rectangular tile out of RNA, which is called RNA Origami [3]. From this kinetic point of view, Geary et al. [2] proposed a new folding model called the oritatami system (OS). An OS consists of a sequence of beads (which is the transcript) and a set of rules for 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 place the current bead at the location. Then it reads the next bead and repeat the same procedure until there is no more bead to place. The lookahead represents the reaction rate of the cotranscriptional folding and the number of interactions represents the energy level. In OS, we call the secondary structure the conformation, and the resulting secondary structure the terminal conformation.

Since an OS folds its transcript according to its own ruleset, with more rules, it becomes more difficult to realize the system in experiments. 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. 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) We propose the bead type merging method—merge two bead types that have the same interaction with other bead types. 2) We propose representative fuzzy ruleset construction—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 by a modified bead type merging, and construct a reduced ruleset from the fuzzy ruleset.

2 Preliminaries

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 $T = (V, E)$ be the triangular grid graph. A directed simple path $P = p_1p_2 \cdots$ in $T$ 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 $H \subseteq \Sigma \times \Sigma$ is a symmetric relation over the set of pairs of bead types such that, for all bead types $a, b \in \Sigma$, $(a, b) \in H$ implies $(b, a) \in H$.

A conformation instance, or configuration, is a triple $(P, w, H)$ of a directed path $P$ in $T$, $w \in \Sigma^* \cup \Sigma^\omega$, and a set $H \subseteq \{(i, j) \mid 1 \leq i, i + 2 \leq j, \{P[i], P[j]\} \in E\}$ of interactions. This is to be 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 denoted by $C = [(P, w, H)]$. We call $w$ a primary structure of $C$. Let $\mathcal{H}$ be a ruleset. A rule $(a, b) \in \mathcal{H}$ is useful in the conformation $C = [(P, w, H)]$ if there exists $(i, j) \in H$ such that $w[i] = a$ and $w[j] = b$ or vice versa. Otherwise, the rule is useless in the conformation. 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 $\alpha \geq 1$, $C$ is of arity $\alpha$ if the maximum number of interactions per bead is $\alpha$, that is, if for any $k \geq 1$, $|\{i \mid (i, k) \in H\}| + |\{j \mid (k, j) \in H\}| \leq \alpha$ and this inequality holds as an equation for some $k$. By $C_{\leq \alpha}$, we 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 $H' \subseteq \{(i, |P|+1) \mid 1 \leq i \leq |P|-1, \{P[i], p\} \in E, (w[i], b) \in \mathcal{H}\}$. This operation is recursively extended to the elongation by a finite sequence of beads as follows: For any configuration $C, C' \xrightarrow{\mathcal{H}}_w C$; 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 $C_1 \xrightarrow{\mathcal{H}}^+_w b C_2$, if there is a conformation $C'$ that satisfies $C_1 \xrightarrow{\mathcal{H}}^+_w C'$ and $C' \xrightarrow{\mathcal{H}}_b C_2$.

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, $\delta \geq 1$ 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 minimize energy collaboratively with the succeeding $\delta - 1$ nascent beads. The energy of a conformation $C = [(P, w, H)]$ is $U(C) = -|H|$; namely, the more interactions a conformation has, the more stable it becomes. The set $\mathcal{F}(\Xi)$ of conformations foldable by this system is recursively defined as follows: The seed $C_\sigma$ is in $\mathcal{F}(\Xi)$; and provided that an elongation $C_i$ of $C_\sigma$ by the prefix $w[1 : i]$ be foldable (i.e., $C_0 = C_\sigma$), its further elongation $C_{i+1}$ by the next bead $w[i+1]$ is foldable if

$$C_{i+1} \in \arg\min_{C \in C_{\leq \alpha}} \min_{C_i \xrightarrow{\mathcal{H}}_w^+ w[i+1] C} \{U(C') \mid C \xrightarrow{\mathcal{H}}^+_w w[i+2 : i+k] C', k \leq \delta, C' \in C_{\leq \alpha}\}. \quad (1)$$

Once we have $C_{i+1}$, we say that the bead $w[i+1]$ and its interactions are stabilized according to $C_{i+1}$. 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 (1). Namely, a deterministic OS folds into a unique terminal conformation.

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 $C_1$ and $C_2$ of conformations, we say that two sets are isomorphic if there exists an one-to-one mapping $C_1 \in C_1 \rightarrow C_2 \in C_2$ 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 if the rule is useful in one of the terminal conformations of the system.
3 Hardness of ruleset optimization on isomorphic oritatami systems

We first define the ruleset optimization problem on isomorphic OSs.

**Problem 1 (RSOPT-Isomorphic).** Given an OS \( \Xi = (\Sigma, w, H, \delta, \alpha, C_\sigma = [(P_\sigma, w_\sigma, H_\sigma)]) \), find an isomorphic OS \( \Xi' = (\Sigma', w', H', \delta, \alpha, C'_\sigma = [(P'_\sigma, w'_\sigma, H'_\sigma)]) \) where \( |H'| \) is minimum.

Before we tackle the problem, we revisit the following problem.

**Problem 2 (RSD-UniqConformation [5]).** Given a finite conformation \( C = [(P, w, H)] \), an alphabet \( \Sigma \), an arity \( \alpha \), a delay \( \delta \), a seed \( C_\sigma = [(P_\sigma, w_\sigma, H_\sigma)] \)\(^1\), and a finite transcript \( w \in \Sigma^* \), find a ruleset \( H \) such that \( C'' = [(P_\sigma P, w_\sigma w, H_\sigma \cup H)] \) is the unique terminal conformation of the OS \( \Xi = (\Sigma, w, H, \delta, \alpha, C_\sigma) \).

The problem is NP-hard when \( \alpha, \delta \geq 2 \) or \( \delta \geq 3 \). Now, we propose another problem (Problem 3) and prove its hardness based on the proof for the RSD-UniqConformation problem [5]. Then, we prove the hardness of the RSOPT-Isomorphic problem using the hardness result of Problem 3.

**Problem 3 (RSD-Isomorphic).** Given a path \( P \), a set \( H \) of interactions, an alphabet \( \Sigma \), an arity \( \alpha \), a delay \( \delta \), a seed \( C_\sigma = [(P_\sigma, w_\sigma, H_\sigma)] \), find a ruleset \( H \) and a finite transcript \( w \) such that \( C = [(P_\sigma P, w_\sigma w, H_\sigma \cup H)] \) is the unique terminal conformation of the OS \( \Xi = (\Sigma, w, H, \delta, \alpha, C_\sigma) \).

**Lemma 1.** The RSD-Isomorphic problem is NP-hard when \( \alpha, \delta \geq 2 \) or \( \delta \geq 3 \).

**Theorem 1.** The RSOPT-Isomorphic problem is NP-hard when \( \alpha, \delta \geq 2 \) or \( \delta \geq 3 \).

4 Ruleset reduction by bead type merging

Since the RSOPT-Isomorphic problem is NP-hard in general, we consider a poly-time heuristic for optimizing a ruleset efficiently. Because not all rules in a ruleset are useful, we start with removing useless rules. For a deterministic OS, it is sufficient to simulate the OS and find useless rules. The simulation takes \( O(n \cdot 5^d) \) time, where \( n \) is the length of the transcript.

**Corollary 1.** For a deterministic OS \( \Xi = (\Sigma, w, H, \delta, \alpha, C_\sigma = [(P_\sigma, w_\sigma, H_\sigma)]) \), we can remove useless rules in \( O(n \cdot 5^d) \) time, where \( n = |w| \).

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

**Theorem 2.** For a nondeterministic OS \( \Xi = (\Sigma, w, H, \delta, \alpha, C_\sigma = [(P_\sigma, w_\sigma, H_\sigma)]) \) and a rule \( r \in H \), it is coNP-hard to determine whether or not \( r \) is useful.

---

\(^1\) In the original paper, the seed was defined by a single term \( \sigma \).
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 $(a; c) \in H$ if and only if $(b; c) \in H$ 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 4 (RSR-BTM-Isomorphic).** Given a ruleset $H \subseteq \Sigma \times \Sigma$ of an OS, find a minimum alphabet $\Sigma'$ and a ruleset $H' \subseteq \Sigma' \times \Sigma'$, where there exists a homomorphism $h : \Sigma \to \Sigma'$ such that for each pair of bead types $(x_1, x_2) \in \Sigma \times \Sigma$, $(x_1, x_2) \in H \Leftrightarrow (h(x_1), h(x_2)) \in H'$.

For the RSR-BTM-Isomorphic problem, we construct a binary string $x_i$ for each bead type $\sigma_i$, where $x_i[j] = 1$ if $(\sigma_i, \sigma_j) \in H$ and 0 otherwise. It is straightforward that if $x_i = x_j$, then $\sigma_i$ and $\sigma_j$ can be successfully merged. We run 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 successfully merged. Since the length of the strings is $t$, the radix sort requires $O(t^2)$ time using $O(t)$ space.

**Theorem 3.** We can solve the RSR-BTM-Isomorphic problem in $O(t^2)$ time using $O(t)$ space, where $t = |\Sigma|$.

## 5 Rule set reduction by fuzzy ruleset construction

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

1. If $(a, b) \in H_P$, then $(a, b) \in H'$.
2. If $(a, b) \notin H_N$, then $(a, b) \notin H'$.

Namely, if a fuzzy ruleset $(H_P, H_N)$ is representative, then rules in $H_P$ should be included in the ruleset, and rules in $H_N$ should be excluded from the ruleset, which ensures that the system is isomorphic to the original system.

We reduce the ruleset size in two phases: First, given an OS $\Xi$, we extract a representative fuzzy ruleset from $\Xi$. Second, we modify the ruleset graph reduction in Section 4 and reduce the size of the fuzzy ruleset. Then, using the fuzzy ruleset, we further reduce the ruleset size.

**Problem 5 (FRS-Extraction).** Given an OS $\Xi = (\Sigma, w, H, \delta, \alpha, C_\sigma = [P_\sigma, w_\sigma, H_\sigma])$, find a representative fuzzy ruleset $(H_P, H_N)$ minimizing $|H_P| + |H_N|$.

**Theorem 4.** The FRS-Extraction problem is NP-hard when $\alpha, \delta \geq 2$ or $\delta \geq 3$. 
Next, we design a heuristic algorithm for the FRS-Extraction problem. Assume that an OS $\Xi$ folds the set $\{C_1=([P_\sigma P_\xi, w_\sigma w, H_\sigma \cup H_\xi])\}$ of $t$ terminal conformations. We assume that $|w|=n$, and $|w_\sigma|$ and $|H_\sigma|$’s 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. We take the following approach for the problem: First, for each point in $P_\sigma$ or $P_\xi$, we assign a distinct bead type to retrieve $w_\sigma$ and $w$. Second, we find conditions of the rules, which are necessary and sufficient for an isomorphic OS. Third, we construct a representative fuzzy ruleset from these conditions. Let $P_1 = p_1 p_2 \cdots p_n$ and $P_\sigma = p_{n+1} p_{n+2} \cdots p_{n+m}$.

At first, let $\Sigma = \{\kappa_i \mid 1 \leq i \leq n+m\}$ and assume that $\kappa_i$ is placed at $p_i$. 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 $H$. The last output is the conditional ruleset $H_k = \{(K \in \Sigma \times \Sigma, s)\}$, which implies that the number of rules in $K \cap H$ should not exceed $s$. The conditional ruleset has information of rules that are not explicitly shown in the most stable elongation but prevent the path from not following $P_1$.

**Lemma 2.** Algorithm 1 runs in $O(5^5 \delta n)$ time using $O(5^5 \delta n)$ space.

Since conditions in $H_k$ 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 $H_k$. Thus, we construct a representative fuzzy ruleset $(P, H_N)$, where $H_N = N \cup N_{add}$ and $N_{add}$ satisfies conditions in $H_k$. We prove that minimizing $|H_N|$ is NP-complete.

**Lemma 3.** Given a set $H_k \subseteq 2^{\Sigma \times \Sigma} \times N$, let $N_{add} \subseteq \Sigma \times \Sigma$ be a set such that, for all $(K_i, s_i) \in H_k$, $|K_i| - |K_i \cap N_{add}| < s_i$ holds. Then, it is NP-complete to find $N_{add}$ with the minimum size.

Since finding the minimum $N_{add}$ is NP-complete, we run Algorithm 2.

**Lemma 4.** Algorithm 2 runs in $O(5^5 \delta \log n)$ time using $O(5^5 \delta n)$ space.

Once we have a representative fuzzy ruleset $(H_P, H_N)$, the next step is to construct a reduced ruleset that satisfies conditions of the fuzzy ruleset. We construct a fuzzy ruleset graph from $(H_P, H_N)$ by adding positive edges for rules in $H_P$ and negative edges for rules in $H_N$.

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

**Problem 6 (FRSR-BTM-Isomorphic).** Given a representative fuzzy ruleset $(H_P, H_N)$ of an OS over an alphabet $\Sigma$, find a minimum alphabet $\Sigma'$ and a ruleset $H' \subseteq \Sigma' \times \Sigma'$, where there exists a homomorphism $h : \Sigma \rightarrow \Sigma'$ such that for every $(x_1, x_2) \in \Sigma \times \Sigma$, $((x_1, x_2) \in P \land (x_1, x_2) \notin N) \Leftrightarrow (h(x_1), h(x_2)) \in H'$. 
Lemma 5. The FRSR-BTM-Isomorphic problem is NP-complete.

Note that we reduce the vertex coloring problem to the FRSR-BTM-Isomorphic problem. We formally establish the function $f$ from a fuzzy ruleset graph $G_r = (V_r, E_r)$ to a graph $G_c = (V_r, E_c)$ by the following rules: For all $(v_i, v_j) \in V_r$, $(v_i, v_j) \in E_c$ 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 $n = |V_r|$; We establish the following lemma.

Lemma 6. For a fuzzy ruleset graph $G_r = (V_r, E_r)$, let $G_c = f(G_r)$. Let $v_1$ and $v_2$ be two mergeable nodes in $V$. Let $G_c'$ ($G_r'$) be the graph resulting from $G_c$ ($G_r$) after merging $v_1$ and $v_2$. Then, $G_c' = f(G_r')$.

From Lemma 6, we know that any solution to the vertex coloring problem has its pair solution to the FRSR-BTM-Isomorphic problem. Therefore, we can use approximation algorithms for the vertex coloring problem to find approximate solutions for the FRSR-BTM-Isomorphic problem. One algorithm is Welsh-Powell algorithm [6]. 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, min{ $d_i + 1, t$} colors.

Algorithm 1: ExtractConditionSets

Input: An arity $\alpha$, a delay $\delta$, a path $P_o$ for a seed, a path $P_1$ for a transcript and a set $H_i$ of interactions.

Output: A required condition set $P$, a forbidden condition set $N$ and a conditional ruleset $H_k$.

1. $\Sigma \leftarrow \{\kappa_i | 1 \leq i \leq n + m\}$.
2. place $\kappa_{n+1}, \kappa_{n+2}, \ldots, \kappa_{n+m}$ to $p_{n+1}, p_{n+2}, \ldots, p_{n+m}$ to form $C_o$.
3. place $\kappa_1$ to $p_1$.
4. for $i \leftarrow 2$ to $n$ do
   5. place $\kappa_i$ to $p_i$.
   6. calculate the sum $s_i$ of the interactions that led $\kappa_i$ to the position $p_i$.
   7. for each annotated neighbors $p_j$ of $p_i$ do
      8. if $\{p_i, p_j\} \in H_i$ then add $(\kappa_i, \kappa_j)$ to $P$.
      9. else add $(\kappa_i, \kappa_j)$ to $N$.
   10. for each unannotated path $P' = p'_1p'_2\ldots p'_j$ where $p'_i \neq p_i$ is an unannotated neighbor of $p_{i-1}$ do
      11. $o_j \leftarrow 0, K \leftarrow \emptyset$
      12. for $j \leftarrow 1$ to $\delta$ do
      13. for each annotated neighbors $p_k$ of $p'_j$ where $p_k$ has interactions less than $\alpha$ do
         14. if $(\kappa_{i+j-1}, \kappa_k) \in P$ then $o_j \leftarrow o_j + 1$
         15. else if $s_i = 1$ then add $(\kappa_{i+j-1}, \kappa_k)$ to $N$.
         16. else add $(\kappa_{i+j-1}, \kappa_k)$ to $K$.
      17. if $s_i \neq 1$ then add $(K, s_i - o_j)$ to $H_k$.
5. return $P, N, H_k$. 


Algorithm 2: ExtractFuzzyRuleset

Input: a conditional ruleset $\mathcal{H}_k$

Output: A set $N_{\text{add}}$

1. while $\mathcal{H}_k \neq \emptyset$
   2. for each $(K_i, s_i)$ with the largest $s_i$ do
      3. count the number $\text{occ}(j,k)$ of appearances of $(\kappa_j, \kappa_k)$ in all $K_i$'s.
   4. for each $(K_i, s_i)$ with the largest $s_i$ do
      5. while the condition does not hold do
          6. find a pair $(\kappa_j, \kappa_k)$ of bead types with the biggest $\text{occ}(j,k)$.
          7. add $(\kappa_j, \kappa_k)$ to $N_{\text{add}}$.
      8. delete $(K_i, s_i)$ from $\mathcal{H}_k$.
   9. return $N_{\text{add}}$

In summary, we first extract necessary and sufficient conditions of rules from the set of ruleset sizes by Algorithm 1. We accumulate $P$, $N$ and $\mathcal{H}_c$ by running Algorithm 1 for $1 \leq i \leq t$, and then run Algorithm 2 to construct a representative fuzzy ruleset. 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 FRSR-BTM-Isomorphic problem. We establish the following theorem.

**Theorem 5.** Using Algorithm 1, Algorithm 2 and an approximation algorithm for the vertex coloring problem, we can approximately solve the RSOPT-Isomorphic problem in $O(5^4 \delta n (\delta + \log n + t) + n^3)$ time using $O(5^4 \delta n)$ space.

**References**