Theoretical Computer Science 757 (2019) 11–26

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Hardness of RNA folding problem with four symbols [☆]

Yi-Jun Chang

University of Michigan, Ann Arbor, MI, USA

A R T I C L E   I N F O

Article history:
Received 17 March 2018
Received in revised form 5 July 2018
Accepted 17 July 2018
Available online 31 July 2018
Communicated by T. Calamoneri

Keywords:
Conditional lower bound
Dyck edit distance
Longest common subsequence
RNA folding

A B S T R A C T

An RNA sequence is a string composed of four types of nucleotides, A, C, G, and U. The goal of the RNA folding problem is to find a maximum cardinality set of crossing-free pairs of the form $\{A, U\}$ or $\{C, G\}$ in a given RNA sequence. The problem is central in bioinformatics and has received much attention over the years. Abboud, Backurs, and Williams (FOCS 2015) demonstrated a conditional lower bound for a generalized version of the RNA folding problem based on a conjectured hardness of the k-clique problem. Their lower bound requires the RNA sequence to have at least 36 types of symbols, making the result not applicable to the RNA folding problem in real life (i.e., alphabet size 4). In this paper, we present an improved lower bound that works for the alphabet size 4 case.

We also investigate the Dyck edit distance problem, which is a string problem closely related to RNA folding. We demonstrate a reduction from RNA folding to Dyck edit distance with alphabet size 10. This leads to a much simpler proof of the conditional lower bound for Dyck edit distance problem given by Abboud, Backurs, and Williams (FOCS 2015), and lowers the alphabet size requirement.

1. Introduction

An *RNA sequence* is a string composed of four types of nucleotides, namely A, C, G, and U. Given an RNA sequence, the goal of the *RNA folding* problem is to find a maximum cardinality set of crossing-free pairs of nucleotides, where all the pairs are either $\{A, U\}$ or $\{C, G\}$. The problem is central in bioinformatics, and it has found application in predicting the secondary structure of RNA molecules, which is of importance in molecular biology; see e.g., [2] for more details.

It is well-known that the RNA folding problem can be solved in $O(n^3)$ time via dynamic programming [3]. Due to the importance of the problem in practice, there has been a long line of research aiming at improving the runtime, practically or theoretically [4,2,5–7]. Based on log-shaving techniques such as the four-Russian method, the time complexity of $O\left(\frac{n^3}{\log^2 n}\right)$ can be obtained [6].

Whether the RNA folding problem can be solved in truly sub-cubic time (i.e., $O(n^{3-\epsilon})$ time for some constant $\epsilon > 0$) had been a major open problem until recently. In 2016, Bringmann, Grandoni, Saha, and Williams [8] showed that the problem can be solved in randomized $O(n^{2.8244})$ time and deterministic $O(n^{2.8603})$ time via fast bounded-difference min-plus matrix multiplication. The RNA folding problem can be reduced to min-plus matrix multiplication [8]. The current state-of-the-art algorithm for min-plus matrix multiplication [9] takes $O(n^3/c^{\sqrt{\log n}})$ time, for some constant $c > 1$. Using this algorithm, one

---

[☆] A preliminary version of this paper was presented at the 27th Annual Symposium on Combinatorial Pattern Matching (CPM 2016) Tel Aviv, Israel, June 27–29, 2016 [1]. This work is supported by NSF grants CCF-1217338, CNS-1318294, and CCF-1514383.

E-mail address: cyijun@umich.edu.

https://doi.org/10.1016/j.tcs.2018.07.010
0304-3975/© 2018 Elsevier B.V. All rights reserved.

immediately obtains an RNA folding algorithm with the same time complexity. Bringmann, Grandoni, Saha, and Williams observed that the min-plus matrix multiplication instance resulting from the reduction has the "bounded differences" property, and they showed that bounded-difference min-plus matrix multiplication can be solved in truly sub-cubic time [8].

The algorithm of [8] uses fast matrix multiplication, which does not perform very efficiently in practice. It is still an open question whether there is a combinatorial, non-algebraic, truly sub-cubic time algorithm for RNA folding.

1.1. Conditional lower bounds

A popular way to show hardness of a problem is to demonstrate a lower bound conditioned on some widely accepted hypothesis.

Conjecture 1 *(Strong exponential time hypothesis). There exist no $\epsilon > 0$ and $k_0 > 0$ such that k-SAT with n variables can be solved in time $O(2^{(1-\epsilon)n})$ for all $k > k_0$.*

Conjecture 2 *(k-clique conjecture). There exist no $\epsilon > 0$ and $k_0 > 0$ such that k-clique on graphs with n nodes can be solved in time $O\left(n^{(\omega-\epsilon)k/3}\right)$ for all $k > k_0$, where $\omega < 2.373$ is the matrix multiplication exponent.*

For instance, assuming Strong Exponential Time Hypothesis (SETH), the following bounds are unattainable for any $\epsilon > 0$: (i) an $O(n^{k-\epsilon})$ algorithm for k-dominating set problem [10]; (ii) an $O(n^{2-\epsilon})$ algorithm for dynamic time warping, longest common subsequence, and edit distance [11–13]; (iii) an $O(m^{2-\epsilon})$ algorithm for $(3/2-\epsilon)$-approximating the diameter of a graph with m edges [14].

As remarked in [15], it is straightforward to reduce the longest common subsequence (LCS) problem on binary strings to the RNA folding problem as follows. Given two binary strings X, Y, let $\hat{X} \in \{A, C\}^{|X|}$ be a string defined as $\hat{X}[i] = A$ if $X[i] = 0$, and $\hat{X}[i] = C$ if $X[i] = 1$; similarly, let $\hat{Y} \in \{G, U\}^{|Y|}$ be a string defined as $\hat{Y}[i] = U$ if $Y[i] = 0$, and $\hat{Y}[i] = G$ if $Y[i] = 1$. Then we have an 1-1 correspondence between RNA foldings of $\hat{X} \circ \hat{Y}^R$ (i.e., concatenation of $\hat{X}$ and the reversal of $\hat{Y}$) and common subsequences of X and Y. It has been shown in [13] that there is no $O(n^{2-\epsilon})$-time algorithm for LCS on binary strings, assuming SETH. Thus, we immediately obtain the same conditional lower bound for RNA folding.

Abboud, Backurs, and Williams demonstrated a higher conditional lower bound for a generalized version of the RNA folding problem (which coincides with the RNA folding problem when the alphabet size is 4) from the k-clique Conjecture [15].

Theorem 1 *([15]). If the generalized RNA folding problem on sequences of length n with alphabet size 36 can be solved in $T(n)$ time, then $3k$-clique on graphs with $|V| = n$ can be solved in $T\left(O(n^{k+2}\log n)\right)$ time.*

Therefore, an $O(n^{\omega-\epsilon})$-time algorithm for the generalized RNA folding with alphabet size at least 36 will disprove the k-clique Conjecture, yielding a breakthrough to the parameterized complexity of the k-clique problem. The current state-of-the-art algorithm for k-clique takes $O\left(n^{\omega k/3}\right)$ time (when k is a multiple of 3), and it requires the use of fast matrix multiplication [16]. For combinatorial, non-algebraic algorithms for k-clique, the current state-of-the-art upper bound is $O\left(\frac{n^k}{\log^k n}\right)$ [17]. Therefore, an $O(n^{3-\epsilon})$-time combinatorial algorithm for RNA folding would imply a breakthrough for combinatorial algorithms for k-clique.

1.2. Our contribution

Due to its alphabet size requirement, Theorem 1 is not applicable to the RNA folding problem in real life (i.e., alphabet size 4). It is unknown whether the generalized RNA folding for alphabet size 4 admits a faster algorithm than the case for alphabet size > 4. There are plenty of string algorithms whose runtime depends on the alphabet size (e.g., string matching with mismatches [18] and jumbled indexing [19,20]). Note that when the alphabet size is 2, the generalized RNA folding problem can be trivially solved in linear time. In this paper, we improve upon Theorem 1 by showing the same conditional lower bound still for the case of alphabet size 4. Note that we also get an $O(n)$ factor improvement inside $T(\cdot)$, though it does not affect the conditional lower bound.

Theorem 2. *If the RNA folding problem on sequences in $\{A, C, G, U\}^n$ can be solved in $T(n)$ time, then $3k$-clique on graphs with $|V| = n$ can be solved in $T\left(O(n^{k+1}\log n)\right)$ time.*

In what follows, we briefly overview the proof of Theorem 2. At a high level, our reduction (from $3k$-clique to RNA folding) follows the approach in [15]. Given a graph G, enumerate all k-cliques, and each of them is encoded as some gadgets. All the gadgets are then put together to form an RNA sequence. The goal is to ensure that an optimal RNA folding corresponds to choosing three k-cliques that form a $3k$-clique, given that the underlying graph admits a $3k$-clique.

Intuitively, in order to force the gadgets to be matched in a desired manner in an optimal RNA folding, we have to build various "walls" that prevent undesired pairings. The main challenge is that we have to achieve this goal using merely 4 types of symbols. Our main tool is to use the technique "alignment gadget" developed in [13], whose original purpose is to prove that longest common subsequence and other edit distance problems are SETH-hard even on binary strings. We apply this tool as a black box during the construction of the RNA sequence.

Dyck edit distance. The RNA folding problem can be alternatively defined as follows. Given a string S, delete the minimum number of letters in S to transform it into another string S' in the language defined by the grammar $\mathbf{S} \to \mathbf{SS}, A\mathbf{S}U, U\mathbf{S}A, C\mathbf{S}G, G\mathbf{S}C, \epsilon$ (empty string). The *Dyck edit distance problem* [21,22], which asks for the minimum number of edits to transform a given string to well-balanced parentheses of s different types, admits a similar formulation. Due to the similarity, the Dyck edit distance problem was shown to admit the same conditional lower bound as Theorem 1 [15]. Their conditional lower bound requires the alphabet size to be at least 48. In this paper, we present a simple reduction from RNA folding to Dyck edit distance.

Theorem 3. *If Dyck edit distance problem on sequences of length n with alphabet size 10 can be solved in $T(n)$ time, then the RNA folding problem on sequences in $\{A, C, G, U\}^n$ can be solved in $O(T(n))$ time.*

Combining Theorem 2 and Theorem 3, we obtain the following corollary.

Corollary 1. *If the Dyck edit distance problem on sequences of length n with alphabet size 10 can be solved in $T(n)$ time, then $3k$-clique on graphs with $|V| = n$ can be solved in $T\left(O(n^{k+1}\log n)\right)$ time.*

This improves upon the conditional lower bound in [15] (reducing the alphabet size from 48 to 10), and it also simplifies the proof (the original proof in [15] takes about 9 pages).

2. Preliminaries

Given a set of letters Σ, the set Σ' is defined as $\{x' \mid x \in \Sigma\}$. It is required that $\Sigma \cap \Sigma' = \emptyset$, and $\forall x, y \in \Sigma, (x \neq y) \to (x' \neq y')$. Therefore, $|\Sigma'| = |\Sigma|$ and $|\Sigma \cup \Sigma'| = 2|\Sigma|$.

For any string $X = (x_1, \ldots, x_k) \in \Sigma^k$, we write $p(X)$ to denote $(x'_1, \ldots, x'_k)$. The letter p stands for the prime symbol. We denote the reversal of the sequence X as X^R. The concatenation of two sequences X, Y is denoted as $X \circ Y$, or simply XY. We write *substring* to denote a contiguous subsequence. We say that two pairs of indices (i_1, j_1), (i_2, j_2), with $i_1 < j_1$ and $i_2 < j_2$, form a *crossing pair* if

$$(\{i_1, j_1\} \cap \{i_2, j_2\} \neq \emptyset) \vee (i_1 < i_2 < j_1 < j_2) \vee (i_2 < i_1 < j_2 < j_1).$$

Generalized RNA folding. Given a string $S \in (\Sigma \cup \Sigma')^n$, an *RNA folding* of S is a set $A \subseteq \{(i, j) \mid 1 \le i < j \le n\}$ meeting the following two conditions.

- A does not contain any crossing pair.
- For each $(i, j) \in A$, either $S[i] \in \Sigma$ and $S[j] = S[i]'$, or $S[j] \in \Sigma$ and $S[i] = S[j]'$ is true.

The goal of the generalized RNA folding problem is to find a maximum cardinality RNA folding A^*. We write $\mathsf{RNA}(S) = |A^*|$, where A^* is any maximum cardinality RNA folding of S. Any RNA folding A satisfying $|A| = \mathsf{RNA}(S)$ is said to be *optimal*.

In the paper we only focus on the generalized RNA folding problem with four types of letters, i.e. $\Sigma = \{0, 1\}$, $\Sigma' = \{0', 1'\}$, which coincides with the RNA folding problem for alphabet $\{A, C, G, U\}$. With a slight abuse of notation, sometimes we write $(S[i], S[j])$ to denote a pair $(i, j) \in A$. The notation $\{\cdot, \cdot\}$ is used to indicate an unordered pair.

Longest common subsequence. Given $X \in \Sigma^n$ and $Y \in \Sigma^m$, we define $\delta_{\mathrm{LCS}}(X, Y)$ as the minimum number of letters from X and Y needed to be deleted to make them identical. That is, $\delta_{\mathrm{LCS}}(X, Y) = n + m - 2k$, where k is the length of the longest common subsequence (LCS) of X and Y. Observe that $\mathsf{RNA}(X \circ p(Y^R)) = (n + m - \delta_{\mathrm{LCS}}(X, Y))/2$ equals the length of LCS of X and Y. Hence the LCS problem can be viewed as the RNA folding problem with some structural constraint on the RNA sequence.

Alignment gadgets. In [13], a SETH-based conditional lower bound for LCS with $|\Sigma| = 2$ was shown. A key technique in their approach is a function that transforms an instance of an alignment problem between two sets of sequences to an instance of the LCS problem, which is briefly described as follows.

Let $\mathbf{X} = (X_1, \ldots, X_n)$ and $\mathbf{Y} = (Y_1, \ldots, Y_m)$ be two linearly ordered sets of sequences of alphabet Σ. We assume that $n \ge m$. An *alignment* is a set $A = \{(i_1, j_1), (i_2, j_2), \ldots, (i_{|A|}, j_{|A|})\}$ with $1 \le i_1 < i_2 < \cdots < i_{|A|} \le n$ and $1 \le j_1 < j_2 < \cdots < j_{|A|} \le m$. An alignment A is called *structural* iff $|A| = m$ and $i_m = i_1 + m - 1$. That is, all sequences in $\mathbf{Y}$ are matched, and the matched positions in $\mathbf{X}$ are contiguous. The set of all alignments is denoted by $\mathcal{A}_{n,m}$, and the set of all structural alignments is denoted by $\mathcal{S}_{n,m}$. The *cost* of an alignment A (with respect to $\mathbf{X}$ and $\mathbf{Y}$) is defined as

$$\delta(A) = \sum_{(i,j)\in A} \delta_{\mathrm{LCS}}(X_i, Y_j) + (m - |A|) \max_{i,j} \delta_{\mathrm{LCS}}(X_i, Y_j).$$

That is, unaligned parts of $\mathbf{Y}$ are penalized by $\max_{i,j}\delta_{\mathrm{LCS}}(X_i, Y_j)$. Given a sequence X, the *type* of X is defined as $(|X|, \sum_i X[i])$, where each letter is assumed to be a number. Note that if the alphabet is $\Sigma = \{0, 1\}$ (which is the case in this paper), then $\sum_i X[i]$ is the number of occurrences of 1 in X. The following lemma was proved in [13].[1]

Lemma 1 *([13]). Let* $\mathbf{X} = (X_1, \ldots, X_n)$ *and* $\mathbf{Y} = (Y_1, \ldots, Y_m)$ *be two linearly ordered sets of binary strings such that* $n \geq m$. *All* X_i *are of type* $\mathcal{T}_X = (\ell_X, s_X)$, *and all* Y_i *are of type* $\mathcal{T}_Y = (\ell_Y, s_Y)$. *There are two binary strings* $S_X = \mathrm{GA}_X^{m,\mathcal{T}_Y}(X_1, \ldots, X_n)$, $S_Y = \mathrm{GA}_Y^{n,\mathcal{T}_X}(Y_1, \ldots, Y_m)$ *and an integer* C *meeting the following requirements.*

- $\min_{A \in \mathcal{A}_{n,m}} \delta(A) \leq \delta_{\mathrm{LCS}}(S_X, S_Y) - C \leq \min_{A \in \mathcal{S}_{n,m}} \delta(A)$.
- *the integer* C *and the types of* S_X *and* S_Y *only depend on* $n, m, \mathcal{T}_X, \mathcal{T}_Y$.
- S_X, S_Y, *and* C *can be calculated in time* $O((n+m)(\ell_X + \ell_Y))$; *hence* $|S_X|$ *and* $|S_Y|$ *are both* $O((n+m)(\ell_X + \ell_Y))$.

In Lemma 1, $\mathrm{GA}_X^{m,\mathcal{T}_Y}(X_1, \ldots, X_n)$ is a function of $X_1, \ldots, X_n$ parameterized by m and $\mathcal{T}_Y$; and $\mathrm{GA}_Y^{n,\mathcal{T}_X}(Y_1, \ldots, Y_m)$ is a function of $Y_1, \ldots, Y_m$ parameterized by n and $\mathcal{T}_X$.

Intuitively, computing an optimal alignment (or an optimal structural alignment) of two sets of sequences is at least as hard as computing a longest common subsequence. Lemma 1 gives a reduction from the computation of a number s with $\min_{A \in \mathcal{A}_{n,m}} \delta(A) \leq s \leq \min_{A \in \mathcal{S}_{n,m}} \delta(A)$ (which can be regarded as an approximately optimal alignment) to a single LCS instance.

We use Lemma 1 as a black box to devise two encodings, the clique node gadget $\mathrm{CNG}(t)$ and the clique list gadget $\mathrm{CLG}(t)$, for any k-clique t in a graph, in such a way that whether two k-cliques t_1 and t_2 form a $2k$-clique can be inferred from the value of $\delta_{\mathrm{LCS}}(\mathrm{CNG}(t_1), \mathrm{CLG}(t_2))$.

3. From cliques to RNA folding

The goal of this section is to prove Theorem 2. Let $G = (V, E)$ be a graph, and let $n = |V|$. We write $\mathcal{C}_k$ to denote the set of k-cliques in G. We fix $\Sigma = \{0, 1\}$. As in [15], we construct a sequence $S_G \in (\Sigma \cup \Sigma')^*$ such that we can decide whether G has a $3k$-clique according to the value of $\mathrm{RNA}(S_G)$. The building blocks in the construction of S_G carry the same names as their analogues in [15], though they have different lower-level implementations.

The organization of this section is as follows. In Section 3.1 we describe the two gadgets $\mathrm{CNG}(t)$ and $\mathrm{CLG}(t)$ for a k-clique t based on the black box tool of Lemma 1. In Section 3.2, adapting the gadgets developed in Section 3.1, we present the definition of the binary sequence S_G. In Section 3.3, we show that there exists an optimal RNA foldings of S_G satisfying some good properties, and then we calculate the value of $\mathrm{RNA}(S_G)$ in Section 3.4.

3.1. Testing 2k-cliques via LCS

We associate with each vertex $v \in V$ a unique ID in $\{0, 1, \ldots\ n-1\}$. Let s_v be the binary encoding of the ID of v. Note that $|s_v| = \lceil \log(n) \rceil$ for each vertex v. We define $\bar{v}$ as the binary string resulting from replacing each 0 in s_v by 01 and replacing each 1 in s_v by 10. Observe that (i) $\bar{v}$ is of type $\mathcal{T}_0 = (2\lceil \log(n) \rceil, \lceil \log(n) \rceil)$ for each $v \in V$, and (ii) $\delta_{\mathrm{LCS}}(\bar{u}, \bar{v}) = 0$ if and only if $u = v$.

Let $v \in V$ be any vertex, and let $N(v) = \{u_1, u_2, \ldots, u_{|N(v)|}\}$ be the set of neighbors of v. The *list gadget* $\mathrm{LG}(v)$ and the *node gadget* $\mathrm{NG}(v)$ for the vertex v are defined as follows.

$$\mathrm{LG}(v) = \mathrm{GA}_X^{1,\mathcal{T}_0}\left(\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_{|N(v)|}, 1^{\lceil \log(n) \rceil} 0^{\lceil \log(n) \rceil}, \ldots, 1^{\lceil \log(n) \rceil} 0^{\lceil \log(n) \rceil}\right),$$

where the number of occurrences of $1^{\lceil \log(n) \rceil} 0^{\lceil \log(n) \rceil}$ is $n - |N(v)|$.

$$\mathrm{NG}(v) = \mathrm{GA}_Y^{n,\mathcal{T}_0}(\bar{v}).$$

Lemma 2. *There is a number* c_0, *depending only on* n, *such that for any two vertices* $v_1, v_2 \in V$, *we have* $\delta_{\mathrm{LCS}}(LG(v_1), NG(v_2)) \geq c_0$, *with equality if and only if* $\{v_1, v_2\} \in E$.

Proof. Let $v_1, v_2 \in V$. Let $N(v_1) = \{u_1, u_2, \ldots, u_{|N(v_1)|}\}$. Define the two sequences of binary strings $\mathbf{X}$ and $\mathbf{Y}$ as follows.

$$\mathbf{X} = \left(\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_{|N(v_1)|}, 1^{\lceil \log(n) \rceil} 0^{\lceil \log(n) \rceil}, \ldots, 1^{\lceil \log(n) \rceil} 0^{\lceil \log(n) \rceil}\right),$$

where the number of occurrences of $1^{\lceil \log(n) \rceil} 0^{\lceil \log(n) \rceil}$ is $n - |N(v_1)|$.

$$\mathbf{Y} = (\bar{v}_2).$$

Note that $|\mathbf{X}| = n$ and $|\mathbf{Y}| = 1$; we have $\mathrm{LG}(v) = \mathrm{GA}_X^{1,\mathcal{T}_0}(\mathbf{X})$ and $\mathrm{NG}(v) = \mathrm{GA}_Y^{n,\mathcal{T}_0}(\mathbf{Y})$.

---

[1] See Lemma 4.3 in the arXiv version (1504.01431v2) of [13].

In view of Lemma 1, we have $\min_{A\in\mathcal{A}_{n,1}} \delta(A) \le \delta_{\mathrm{LCS}}(\mathsf{LG}(v_1),\ \mathsf{NG}(v_2)) - C \le \min_{A\in\mathcal{S}_{n,1}} \delta(A)$, for some number C whose value depends on $|\mathbf{X}|, |\mathbf{Y}|$, and $\mathcal{T}_0$. As these parameters depend solely on n, the number C also depends only on n (i.e., the choice of the two vertices v_1 and v_2 does not affect C). We claim that setting $c_0 = C$ suffices to prove the lemma.

Since $|\mathbf{Y}| = 1$, any non-empty alignment between $\mathbf{X}$ and $\mathbf{Y}$ is structural. This implies that $\delta_{\mathrm{LCS}}(\mathsf{LG}(v_1),\ \mathsf{NG}(v_2)) - C = \min_{A\in\mathcal{A}_{n,1}} \delta(A) = \min_{A\in\mathcal{S}_{n,1}} \delta(A)$.

For the case $\{v_1, v_2\} \in E$, since $\bar{v}_2$ is contained in $\mathbf{X}$, clearly $\min_{A\in\mathcal{S}_{n,m}} \delta(A) = 0$. For the case $\{v_1, v_2\} \notin E$, $\bar{v}_2$ does not appear in $\mathbf{X}$, so $\min_{A\in\mathcal{S}_{n,m}} \delta(A) > 0$. Note that $1^{\lceil\log(n)\rceil}0^{\lceil\log(n)\rceil} \neq \bar{v}$, for each $v \in V$. As a result, for any two vertices $v_1, v_2 \in V$, we have $\delta_{\mathrm{LCS}}(LG(v_1), NG(v_2)) \ge c_0$, with equality if and only if $\{v_1, v_2\} \in E$. □

In view of Lemma 1, the type of list gadgets and the type of node gadgets depends only on n; that is, they are independent of the underlying vertex v. Let $\mathcal{T}_X$ be the type of the list gadgets, and let $\mathcal{T}_Y$ be the type of the node gadgets. For each k-clique $t = \{u_1, u_2, \ldots, u_k\}$, define the *clique node gadget* $\mathsf{CNG}(t)$ and the *clique list gadget* $\mathsf{CLG}(t)$ as follows.

$$\mathsf{CLG}(t) = \mathsf{GA}_X^{k^2,\mathcal{T}_Y}\left(\mathsf{LG}(u_1), \ldots, \mathsf{LG}(u_1), \mathsf{LG}(u_2), \ldots, \mathsf{LG}(u_2), \ldots, \mathsf{LG}(u_k), \ldots, \mathsf{LG}(u_k)\right),$$

where the number of occurrences of each $\mathsf{LG}(u_i)$ is k.

$$\begin{aligned}\mathsf{CNG}(t) = \mathsf{GA}_Y^{k^2,\mathcal{T}_X}\big(&\mathsf{NG}(u_1), \mathsf{NG}(u_2), \ldots, \mathsf{NG}(u_k),\\ &\mathsf{NG}(u_1), \mathsf{NG}(u_2), \ldots, \mathsf{NG}(u_k),\\ &\ldots,\\ &\mathsf{NG}(u_1), \mathsf{NG}(u_2), \ldots, \mathsf{NG}(u_k)\big),\end{aligned}$$

where the number of occurrences of each $\mathsf{NG}(u_1), \mathsf{NG}(u_2), \ldots, \mathsf{NG}(u_k)$ is k.

Lemma 3. *There is a number c_1, depending only on n and k, such that for any two k-cliques $t_1, t_2 \in \mathcal{C}_k$, we have $\delta_{\mathrm{LCS}}(\mathsf{CLG}(t_1), \mathsf{CNG}(t_2)) \ge c_1$, with equality if and only if the set of vertices $t_1 \cup t_2$ form a $2k$-clique.*

Proof. Let $t_1 = \{u_1, u_2, \ldots, u_k\}$, and let $t_2 = \{v_1, v_2, \ldots, v_k\}$ be two k-cliques. Define the two sequences of binary strings $\mathbf{X}$ and $\mathbf{Y}$ as follows.

$$\mathbf{X} = \left(\mathsf{LG}(u_1), \ldots, \mathsf{LG}(u_1), \mathsf{LG}(u_2), \ldots, \mathsf{LG}(u_2), \ldots, \mathsf{LG}(u_k), \ldots, \mathsf{LG}(u_k)\right),$$

where the number of occurrences of each $\mathsf{LG}(u_i)$ is k.

$$\begin{aligned}\mathbf{Y} = \big(&\mathsf{NG}(v_1), \mathsf{NG}(v_2), \ldots, \mathsf{NG}(v_k),\\ &\mathsf{NG}(v_1), \mathsf{NG}(v_2), \ldots, \mathsf{NG}(v_k),\\ &\ldots,\\ &\mathsf{NG}(v_1), \mathsf{NG}(v_2), \ldots, \mathsf{NG}(v_k)\big),\end{aligned}$$

where the number of occurrences of each $\mathsf{NG}(v_1), \mathsf{NG}(v_2), \ldots, \mathsf{NG}(v_k)$ is k.

Note that $|\mathbf{X}| = |\mathbf{Y}| = k^2$; we have $\mathsf{CLG}(t) = \mathsf{GA}_X^{k^2,\mathcal{T}_Y}(\mathbf{X})$ and $\mathsf{CNG}(t) = \mathsf{GA}_Y^{k^2,\mathcal{T}_X}(\mathbf{Y})$. In view of Lemma 2, $\min_{w_1,w_2\in V} \delta_{\mathrm{LCS}}(\mathsf{LG}(w_1), \mathsf{NG}(w_2)) = c_0$, and so $\min_{A\in\mathcal{A}_{k^2,k^2}} \delta(A) \ge k^2 c_0$.

In view of Lemma 1, $\min_{A\in\mathcal{A}_{k^2,k^2}} \delta(A) \le \delta_{\mathrm{LCS}}(\mathsf{CLG}(t_1),\ \mathsf{CNG}(t_2)) - C \le \min_{A\in\mathcal{S}_{k^2,k^2}} \delta(A)$, for some number C whose value depends on $|\mathbf{X}|$, $|\mathbf{Y}|$, $\mathcal{T}_X$, and $\mathcal{T}_Y$. As these parameters depend solely on n, k, the number C only depends on n, k (i.e., the choice of the two k-cliques t_1 and t_2 does not affect C). We claim that setting $c_1 = C + k^2 c_0$ suffices to prove the lemma.

Consider the case $t_1 \cup t_2$ form a $2k$-clique. That is, each $u_i \in t_1$ is adjacent to each $v_j \in t_2$. Thus, by Lemma 2, we have $\delta_{\mathrm{LCS}}(X_i, Y_j) = c_0$, for all i, j. Recall that X_i denotes the ith string in $\mathbf{X}$, and Y_j denotes the jth string in $\mathbf{Y}$. The structural alignment $A^* = \{(i,i) \mid i \in \{1, 2, \ldots, k^2\}\} \in \mathcal{S}_{k^2,k^2}$ achieves the minimum possible cost $k^2 c_0$. Thus, for the case $t_1 \cup t_2$ form a $2k$-clique, we have

$$\delta_{\mathrm{LCS}}(\mathsf{CLG}(t_1), \mathsf{CNG}(t_2)) - C = \min_{A\in\mathcal{A}_{k^2,k^2}} \delta(A) = \min_{A\in\mathcal{S}_{k^2,k^2}} \delta(A) = k^2 c_0,$$

and so $\delta_{\mathrm{LCS}}(\mathsf{CLG}(t_1), \mathsf{CNG}(t_2)) = C + k^2 c_0 = c_1$, as desired.

Next, consider the case $t_1 \cup t_2$ does not form a $2k$-clique. That is, there exist two vertices $u_{i'} \in t_1$ and $v_{j'} \in t_2$ that are not adjacent. By Lemma 2, we have $\delta_{\mathrm{LCS}}(\mathsf{LG}(u_{i'}), \mathsf{NG}(u_{j'})) > c_0$. We claim that $\min_{A\in\mathcal{A}_{k^2,k^2}} \delta(A) > k^2 c_0$. Suppose that there exists an alignment $A' \in \mathcal{A}_{k^2,k^2}$ such that $\delta(A') = k^2 c_0$. Then all k^2 strings in $\mathbf{Y}$ must be aligned, as each unaligned string in $\mathbf{Y}$ contributes a cost that is higher than c_0. Thus, we must have $A' = \{(i,i) \mid i \in \{1, 2, \ldots, k^2\}\}$. In order to have $\delta(A') = k^2 c_0$, we must have $\delta_{\mathrm{LCS}}(X_i, Y_i) = c_0$, for all $i \in \{1, 2, \ldots, k^2\}$. However, $X_{j'+k(i'-1)} = \mathsf{LG}(u_{i'})$ and $Y_{j'+k(i'-1)} = \mathsf{NG}(v_{j'})$, and so $\delta_{\mathrm{LCS}}(X_{j'+k(i'-1)}, Y_{j'+k(i'-1)}) > c_0$, a contradiction.

Since $\min_{A \in \mathcal{A}_{k^2,k^2}} \delta(A) > k^2 c_0$ for the case $t_1 \cup t_2$ is not a $2k$-clique, we have

$$\delta_{\mathrm{LCS}}(\mathsf{CLG}(t_1), \mathsf{CNG}(t_2)) - C \geq \min_{A \in \mathcal{A}_{k^2,k^2}} \delta(A) > k^2 c_0,$$

and so $\delta_{\mathrm{LCS}}(\mathsf{CLG}(t_1), \mathsf{CNG}(t_2)) > C + k^2 c_0 = c_1$, as desired. □

The following lemma is a simple consequence of Lemma 1.

Lemma 4. *There exist four integers* $\ell_{\mathsf{CNG},0}$, $\ell_{\mathsf{CNG},1}$, $\ell_{\mathsf{CLG},0}$, *and* $\ell_{\mathsf{CLG},1} \in O(k^2 n \log n)$ *such that for any* $t \in \mathcal{C}_k$, *and for any* $b \in \{0, 1\}$, *we have (i)* $\ell_{\mathsf{CNG},b}$ *is the number of occurrences of* b *in* $\mathsf{CNG}(t)$, *and (ii)* $\ell_{\mathsf{CLG},b}$ *is the number of occurrences of* b *in* $\mathsf{CLG}(t)$.

Proof. As a consequence of Lemma 1, all $\mathsf{CNG}(t)$ have the same type, and all $\mathsf{CLG}(t)$ have the same type. Therefore, the existence of these four integers is guaranteed. We show that these numbers are $O(k^2 n \log n)$. In view of Lemma 1, for each $v \in V$, both $\mathsf{LG}(v)$ and $\mathsf{NG}(v)$ have length at most $(n+1) \cdot (2\lceil \log n \rceil + 2\lceil \log n \rceil) = O(n \log n)$. Applying Lemma 1 again, the length of both $\mathsf{CNG}(t)$ and $\mathsf{CLG}(t)$ for each $t \in \mathcal{C}_k$ is $(k^2 + k^2)(O(n \log n) + O(n \log n)) = O(k^2 n \log n)$. □

3.2. The RNA sequence S_G

Based on the parameters in Lemma 4, we define $\ell_0 = \ell_{\mathsf{CNG},0} + \ell_{\mathsf{CNG},1} + \ell_{\mathsf{CLG},0} + \ell_{\mathsf{CLG},1} = O(k^2 n \log n)$; for each $i \in \{1, 2, 3\}$, we set $\ell_i = 100\ell_{i-1}$; and $\ell_4 = 100|\mathcal{C}_k|\ell_3 = \binom{n}{k} O(k^2 n \log n) = O(n^{k+1} \log n/(k-2)!)$. The RNA sequence S_G is then defined as follows.

$$S_G = 0^{\ell_4}\left[0'^{\ell_3} \bigcirc_{t \in \mathcal{C}_k} \left(\mathsf{CG}_\alpha(t) 0'^{\ell_3}\right)\right] 0^{\ell_4}\left[0'^{\ell_3} \bigcirc_{t \in \mathcal{C}_k} \left(\mathsf{CG}_\beta(t) 0'^{\ell_3}\right)\right] 0^{\ell_4}\left[0'^{\ell_3} \bigcirc_{t \in \mathcal{C}_k} \left(\mathsf{CG}_\gamma(t) 0'^{\ell_3}\right)\right],$$

where

$$\mathsf{CG}_\alpha(t) = 1'^{2\ell_2} p(\mathsf{CLG}(t)^R) 0'^{\ell_1} 1^{\ell_2} 0^{\ell_1} \mathsf{CNG}(t) 1^{\ell_2},$$
$$\mathsf{CG}_\beta(t) = 1'^{\ell_2} p(\mathsf{CLG}(t)^R) 0'^{\ell_1} 1'^{2\ell_2} 0'^{\ell_1} p(\mathsf{CNG}(t)) 1'^{\ell_2},$$
$$\mathsf{CG}_\gamma(t) = 1^{\ell_2} \mathsf{CLG}(t)^R 0^{\ell_1} 1^{\ell_2} 0^{\ell_1} \mathsf{CNG}(t) 1^{2\ell_2}.$$

For each $t \in \mathcal{C}_k$, and for each $x \in \{\alpha, \beta, \gamma\}$, the string $\mathsf{CG}_x(t)$ is called a *clique gadget*. Note that $\mathsf{CG}_\alpha(t) \in (\Sigma \cup \Sigma')^*$, $\mathsf{CG}_\beta(t) \in \Sigma'^*$, and $\mathsf{CG}_\gamma(t) \in \Sigma^*$. The length of this RNA sequence is $|S_G| = O(|\mathcal{C}_k|\ell_0) = O(n^{k+1} \log n/(k-2)!)$. Before proceeding further, we present a simple lower bound on $\mathsf{RNA}(S_G)$ by constructing an RNA folding of S_G as follows.

Step 1: Matching the Letters in $0'^{\ell_3}$. Given some pairings between the letters in $0'^{\ell_3}$ and the letters in 0^{ℓ_4}, a clique gadget C is said to be *blocked* if all letters within C can only be paired up with the letters within the same clique gadget or the letters in 0^{ℓ_4}. In particular, a clique gadget that is blocked is unable to participate in the RNA folding with other clique gadgets.

We link all $0'$ in all $0'^{\ell_3}$ to some 0 in some 0^{ℓ_4} in such a way that for each $x \in \{\alpha, \beta, \gamma\}$, there is exactly one clique gadget $\mathsf{CG}_\alpha(t_x)$ that is not blocked, among all clique gadgets $\{\mathsf{CG}_x(t) \mid t \in \mathcal{C}_k\}$. The three clique gadgets $\mathsf{CG}_\alpha(t_\alpha)$, $\mathsf{CG}_\beta(t_\beta)$, and $\mathsf{CG}_\gamma(t_\gamma)$ that are not blocked are called the *selected* clique gadgets. See Fig. 1. This step makes $3(|\mathcal{C}_k| + 1)\ell_3$ pairs.

Step 2: Matching the Letters in a Blocked Clique Gadget. Pair up the letters in each blocked clique gadget as follows. For each blocked $\mathsf{CG}_\alpha(t)$, we pair up as many $\{1', 1\}$ pairs as possible within the clique gadget; this gives us $2\ell_2 + \min(\ell_{\mathsf{CLG},1}, \ell_{\mathsf{CNG},1})$ pairs. For each blocked $\mathsf{CG}_\beta(t)$, we pair up all $0'$ to some 0 in 0^{ℓ_4}; this gives us $2\ell_1 + \ell_{\mathsf{CLG},0} + \ell_{\mathsf{CNG},0}$ pairs. For each blocked $\mathsf{CG}_\gamma(t)$, no pairing can be made. See Fig. 2. In this step, $(|\mathcal{C}_k| - 1)(2\ell_1 + 2\ell_2 + \min(\ell_{\mathsf{CLG},1}, \ell_{\mathsf{CNG},1}) + \ell_{\mathsf{CLG},0} + \ell_{\mathsf{CNG},0})$ pairs are produced.

Step 3: Matching the Letters in the Three Selected Clique Gadget. For the three clique gadgets $\mathsf{CG}_\alpha(t_\alpha)$, $\mathsf{CG}_\beta(t_\beta)$, and $\mathsf{CG}_\gamma(t_\gamma)$ that are not blocked, We pair up the letters in these clique gadgets in such a way that

- all letters in $1'^{2\ell_2}$, $1^{2\ell_2}$, $1'^{\ell_2}$, 1^{ℓ_2}, $0'^{\ell_1}$, and 0^{ℓ_1} are matched, and
- for each $(x, y) \in \{(\alpha, \beta), (\alpha, \gamma), (\beta, \gamma)\}$, $\frac{1}{2}(\ell_0 - \delta_{\mathrm{LCS}}(\mathsf{CLG}(t_x), \mathsf{CNG}(t_y)))$ number of pairs are made between the two gadgets $\mathsf{CLG}(t_x)$ and $\mathsf{CNG}(t_y)$.

See Fig. 3. Recall that $\frac{1}{2}(\ell_0 - \delta_{\mathrm{LCS}}(\mathsf{CLG}(t_x), \mathsf{CNG}(t_y)))$ is the length of the LCS between $\mathsf{CLG}(t_x)$ and $\mathsf{CNG}(t_y)$. The total number of pairs made in this step is

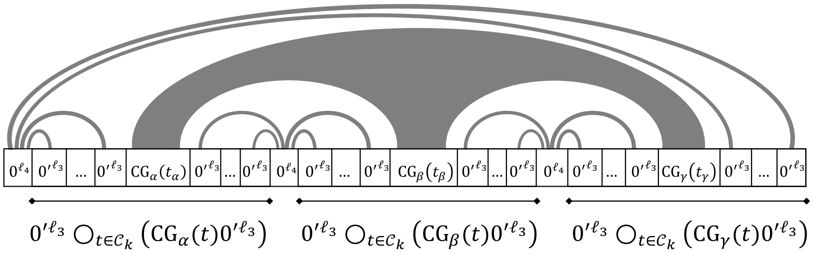

Fig. 1. The three selected clique gadgets and the matchings between $0'^{\ell_3}$ and 0^{ℓ_4}.

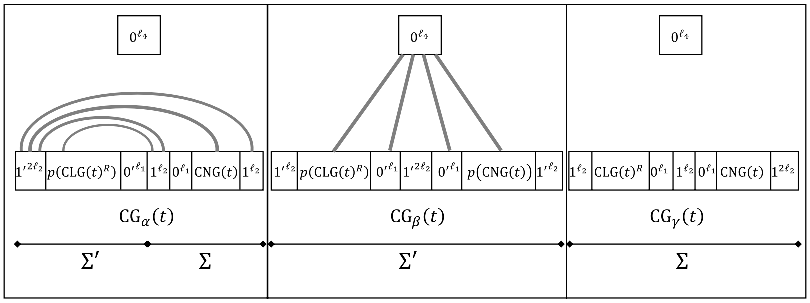

Fig. 2. The matchings between a blocked clique gadget and 0^{ℓ_4}.

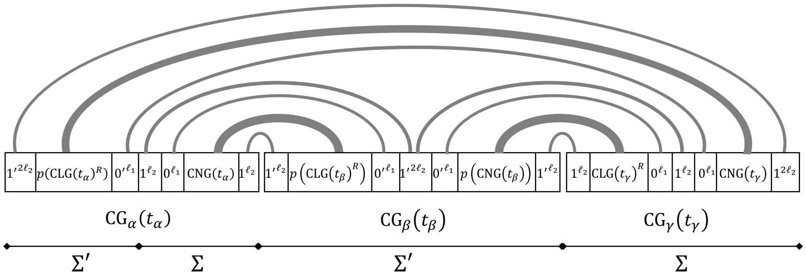

Fig. 3. The matchings within the three selected clique gadgets.

$$6\ell_2+3\ell_1+\frac{1}{2}\left(\ell_0-\delta_{\mathrm{LCS}}(\mathsf{CLG}(t_\alpha),\mathsf{CNG}(t_\beta))\right)$$
$$+\frac{1}{2}\left(\ell_0-\delta_{\mathrm{LCS}}(\mathsf{CLG}(t_\alpha),\mathsf{CNG}(t_\gamma))\right)+\frac{1}{2}\left(\ell_0-\delta_{\mathrm{LCS}}(\mathsf{CLG}(t_\beta),\mathsf{CNG}(t_\gamma))\right).$$

In view of the above discussion, we define the following two numbers.

$$m_1=3(|\mathcal{C}_k|+1)\ell_3+(|\mathcal{C}_k|-1)(2\ell_1+2\ell_2+\min(\ell_{\mathsf{CLG},1},\ell_{\mathsf{CNG},1})+\ell_{\mathsf{CLG},0}+\ell_{\mathsf{CNG},0}),$$
$$m_2=6\ell_2+3\ell_1+\frac{3}{2}\ell_0-\min_{t_\alpha,t_\beta,t_\gamma\in\mathcal{C}_k}Q\left(t_\alpha,t_\beta,t_\gamma\right),\ \text{where } Q\left(t_\alpha,t_\beta,t_\gamma\right) \text{ is defined as}$$
$$\frac{1}{2}\left(\delta_{\mathrm{LCS}}(\mathsf{CLG}(t_\alpha),\mathsf{CNG}(t_\beta))+\delta_{\mathrm{LCS}}(\mathsf{CLG}(t_\alpha),\mathsf{CNG}(t_\gamma))+\delta_{\mathrm{LCS}}(\mathsf{CLG}(t_\beta),\mathsf{CNG}(t_\gamma))\right).$$

The RNA folding given in the above construction has cardinality $m_1+6\ell_2+3\ell_1+\frac{3}{2}\ell_0-Q\left(t_\alpha,t_\beta,t_\gamma\right)$, and so m_1+m_2 is a lower bound of $\mathsf{RNA}(S_G)$.

Lemma 5. $\mathsf{RNA}(S_G)\geq m_1+m_2$.

We will ultimately show that $\mathsf{RNA}(S_G)=m_1+m_2$. Due to Lemma 3, if $t_\alpha\cup t_\beta\cup t_\gamma$ form a $3k$-clique, then $Q\left(t_\alpha,t_\beta,t_\gamma\right)=3c_1/2$; otherwise $Q\left(t_\alpha,t_\beta,t_\gamma\right)>3c_1/2$. Therefore, the number $\mathsf{RNA}(S_G)=m_1+m_2$ offers sufficient information to decide whether G has a $3k$-clique. The following auxiliary lemma will be useful in subsequent discussion.

Lemma 6. *The following statements are true for any two cliques $t, t' \in \mathcal{C}_k$.*

1. $\mathsf{RNA}(0^{\ell_4}\mathsf{CG}_\alpha(t)) = 2\ell_2 + \min(\ell_{\mathsf{CLG},1}, \ell_{\mathsf{CNG},1})$
2. $\mathsf{RNA}(0^{\ell_4}\mathsf{CG}_\beta(t)) = 2\ell_1 + \ell_{\mathsf{CLG},0} + \ell_{\mathsf{CNG},0}$
3. $\mathsf{RNA}(0^{\ell_4}\mathsf{CG}_\gamma(t)) = 0$
4. $\mathsf{RNA}(0^{\ell_4}\mathsf{CG}_\alpha(t)0^{\ell_4}\mathsf{CG}_\beta(t')) \le 3.1\ell_1 + 2\ell_2$
5. $\mathsf{RNA}(0^{\ell_4}\mathsf{CG}_\alpha(t)0^{\ell_4}\mathsf{CG}_\gamma(t')) \le 1.1\ell_1 + 2\ell_2$
6. $\mathsf{RNA}(0^{\ell_4}\mathsf{CG}_\beta(t)0^{\ell_4}\mathsf{CG}_\gamma(t')) \le 1.1\ell_1 + 4\ell_2$

Proof. The value of $\mathsf{RNA}(\cdot)$ for each of the six strings are calculated as follows.

(1) $\mathsf{RNA}(0^{\ell_4}\mathsf{CG}_\alpha(t)) = 2\ell_2 + \min(\ell_{\mathsf{CLG},1}, \ell_{\mathsf{CNG},1})$**:** Pairing up as many 1 to $1'$ yields a matching of size $m = 2\ell_2 + \min(\ell_{\mathsf{CLG},1}, \ell_{\mathsf{CNG},1})$. To see that it is optimal, it suffices to show that both $(0', 0)$ and $(0, 0')$ cannot appear in an optimal RNA folding.
- If the RNA folding contains $(0, 0')$, then none of $1'$ can participate in the RNA folding. As the total number of $0'$ is $\ell_1 + \ell_{\mathsf{CLG},0}$, the size of RNA folding is at most $\ell_1 + \ell_{\mathsf{CLG},0} < m$.
- If the RNA folding contains $(0', 0)$, then at most $\ell_{\mathsf{CLG},1}$ number of letters within the middle 1^{ℓ_2} (the one between $0'^{\ell_1}$ and 0^{ℓ_1}) can participate in the RNA folding. It implies that the number of $(1', 1)$ pairs in the RNA folding is at most $\ell_{\mathsf{CLG},1} + \ell_2$. Hence the size of the RNA folding can be upper bounded by $(\ell_1 + \ell_{\mathsf{CLG},0}) + (\ell_{\mathsf{CLG},1} + \ell_2) < m$.

(2) $\mathsf{RNA}(0^{\ell_4}\mathsf{CG}_\beta(t)) = 2\ell_1 + \ell_{\mathsf{CLG},0} + \ell_{\mathsf{CNG},0}$**:** Since there is no 1, the equation follows from the fact that there are $2\ell_1 + \ell_{\mathsf{CLG},0} + \ell_{\mathsf{CNG},0}$ occurrences of $0'$, all of which can be matched to some 0 without crossing.

(3) $\mathsf{RNA}(0^{\ell_4}\mathsf{CG}_\gamma(t)) = 0$**:** It is impossible to produce any pair since there are no $0'$ and $1'$ in the string.

(4) $\mathsf{RNA}(0^{\ell_4}\mathsf{CG}_\alpha(t)0^{\ell_4}\mathsf{CG}_\beta(t')) \le 3.1\ell_1 + 2\ell_2$**:** The value of $\mathsf{RNA}(\cdot)$ can be upper bounded by the number of 1 and $0'$, which is $(2\ell_2 + \ell_{\mathsf{CNG},1}) + (3\ell_1 + 2\ell_{\mathsf{CLG},0} + \ell_{\mathsf{CNG},0}) \le 3.1\ell_1 + 2\ell_2$.

(5) $\mathsf{RNA}(0^{\ell_4}\mathsf{CG}_\alpha(t)0^{\ell_4}\mathsf{CG}_\gamma(t')) \le 1.1\ell_1 + 2\ell_2$**:** The value of $\mathsf{RNA}(\cdot)$ can be upper bounded by the number of $1'$ and $0'$, which is $(2\ell_2 + \ell_{\mathsf{CLG},1}) + (\ell_1 + \ell_{\mathsf{CLG},0}) \le 1.1\ell_1 + 2\ell_2$.

(6) $\mathsf{RNA}(0^{\ell_4}\mathsf{CG}_\beta(t)0^{\ell_4}\mathsf{CG}_\gamma(t')) \le 1.1\ell_1 + 4\ell_2$**:** Define the string $S = 0^{\ell_4} \circ \left(1'^{\ell_2}0'^{\ell_1}1'^{2\ell_2}0'^{\ell_1}1'^{\ell_2}\right) \circ 0^{\ell_4} \circ \left(1^{\ell_2}0^{\ell_1}1^{\ell_2}0^{\ell_1}1^{2\ell_2}\right)$ as the result of removing the clique node gadgets and the clique list gadgets in $0^{\ell_4}\mathsf{CG}_\beta(t)0^{\ell_4}\mathsf{CG}_\gamma(t')$. It is clear that $\mathsf{RNA}(0^{\ell_4}\mathsf{CG}_\beta(t)0^{\ell_4}\mathsf{CG}_\gamma(t')) \le 0.1\ell_1 + \mathsf{RNA}(S)$, as the total length of the removed substrings can be upper bounded by $0.1\ell_1$. Therefore, it suffices to show that $\mathsf{RNA}(S) \le \ell_1 + 4\ell_2$. Let A be any RNA folding of S.

Case 1: There is a pair $(0, 0') \in A$ where the letter $0'$ comes from the first $0'^{\ell_1}$ in S. Clearly, the first substring $1'^{\ell_2}$ cannot participate in any pairing. Therefore, $|A| \le |0'^{\ell_1}1'^{2\ell_2}0'^{\ell_1}1'^{\ell_2}| = 2\ell_1 + 3\ell_2 < \ell_1 + 4\ell_2$.

Case 2: There is a pair $(0', 0) \in A$ where the letter $0'$ comes from the first $0'^{\ell_1}$ in S. Consider any pair $(1', 1)$ such that the letter 1 is in the substring $1^{2\ell_2}$ in S. In order to have this pair not crossing any pair $(0', 0) \in A$, the letter $1'$ must be in the first substring $1'^{\ell_2}$ in S. Therefore, at most half of the letters in the substring $1^{2\ell_2}$ can participate in the RNA folding A, and so $|A|$ is at most the total number of $0'$ and 1 in S minus ℓ_2, i.e., $|A| \le 2\ell_1 + 3\ell_2 < \ell_1 + 4\ell_2$.

Case 3: The first $0'^{\ell_1}$ in S does not participate in the RNA folding. In this case, we have $|A| \le |1'^{\ell_2}1'^{2\ell_2}0'^{\ell_1}1'^{\ell_2}| = \ell_1 + 4\ell_2$. □

Note that Lemma 6(1, 2, 3) implies that the RNA folding for blocked clique gadgets described in Fig. 2 is optimal, and the optimal number of pairings is irrelevant to the underlying k-clique.

3.3. Optimal RNA foldings of S_G

In this section, we show that there exists an optimal RNA folding of S_G satisfying some good properties. Let A be an RNA folding of a string S, and let S_1 and S_2 be two disjoint substrings of S. Recall that a substring is a subsequence of consecutive elements. We write $S_1 \overset{A}{\longleftrightarrow} S_2$ if there exists a pair $\{x_1, x_2\} \in A$ with $x_1 \in S_1, x_2 \in S_2$. Given an RNA folding A of the string S_G, the two properties (P1) and (P2) are defined as follows.

(P1) All $0'$ in all $0'^{\ell_3}$ are paired up with some 0 in some 0^{ℓ_4} in A.

(P2) There exist $t_\alpha, t_\beta, t_\gamma \in \mathcal{C}_k$ such that the following holds. If $\mathsf{CG}_{u_1}(t_1) \overset{A}{\longleftrightarrow} \mathsf{CG}_{u_2}(t_2)$, then $\{(u_1, t_1), (u_2, t_2)\} \subseteq \{(\alpha, t_\alpha), (\beta, t_\beta), (\gamma, t_\gamma)\}$.

Intuitively, (P2) says that all clique gadgets are blocked by the pairings between $0'^{\ell_3}$ and 0^{ℓ_4}, except the three selected clique gadgets $\mathsf{CG}_\alpha(t_\alpha)$, $\mathsf{CG}_\beta(t_\beta)$, and $\mathsf{CG}_\gamma(t_\gamma)$, for some choices of three k-cliques t_α, t_β, and t_γ.

Lemma 7. *Let A be any RNA folding of S_G. Let S_1 be a substring $0'^{\ell_3}$ of S_G, and let S_2 be a substring 0^{ℓ_4} of S_G. If there is a pair in A linking a letter $0'$ in S_1 to a letter 0 in S_2, then there is another RNA folding A' of S_G with $|A'| \geq |A|$ where all letters in S_1 are paired up with letters in S_2.*

Proof. The lemma immediately follows from the fact that ℓ_4 is greater than the total number of $0'$ in S_G, which makes it possible to rematch all the letters in S_1 to letters in S_2. □

Lemma 8 shows that there is an optimal RNA folding A of S_G satisfying (P1).

Lemma 8. *There is an optimal RNA folding of S_G satisfying (P1).*

Proof. Choose any RNA folding A of S_G with $|A| = \mathsf{RNA}(S_G)$. In view of Lemma 7, we assume that for each substring $0'^{\ell_3}$ in S_G, either (i) all its letters are matched to letters in the same 0^{ℓ_4}, or (ii) none of its letters is matched to any letters in any 0^{ℓ_4}. Let z be the number of $0'^{\ell_3}$ such that none of its letters is matched to any letters in any 0^{ℓ_4}.

Let $t \in \mathcal{C}_k$, and let $x \in \{\alpha, \beta, \gamma\}$. We say that $\mathsf{CG}_x(t)$ is *trapped* in A if each letter in $\mathsf{CG}_x(t)$ is either (i) unmatched, (ii) matched to letters within $\mathsf{CG}_x(t)$, or (iii) matched to a letter in some 0^{ℓ_4}. Note that a sufficient condition for a clique gadget $\mathsf{CG}_x(t)$ to be trapped is that all letters in its two neighboring $0'^{\ell_3}$ are matched to letters in the same substring 0^{ℓ_4}.

Suppose that the clique gadget $\mathsf{CG}_x(t)$ is not trapped in A, then $\mathsf{CG}_x(t)$ falls into one of the following two cases.

Case 1: The letters in the two neighboring substrings $0'^{\ell_3}$ of $\mathsf{CG}_x(t)$ are matched to letters in two distinct substrings 0^{ℓ_4}.
Case 2: A neighboring $0'^{\ell_3}$ of $\mathsf{CG}_x(t)$ is not matched to any 0^{ℓ_4}.

Observe that at most 3 clique gadgets belong to the first case, and at most $2z$ clique gadgets belong to the second case. Thus, the number of clique gadgets that are not trapped in A is at most $3 + 2z$. We derive an upper bound of $|A|$ as follows.

$$\begin{aligned}
|A| \leq{} & (3(|\mathcal{C}_k| + 1) - z)\ell_3 && \text{(matched } 0'^{\ell_3}) \\
& + |\mathcal{C}_k| \Big(\max_{t \in \mathcal{C}_k} \mathsf{RNA}(0^{\ell_4}\mathsf{CG}_\alpha(t)) + \max_{t \in \mathcal{C}_k} \mathsf{RNA}(0^{\ell_4}\mathsf{CG}_\beta(t)) && \text{(trapped clique gadgets)} \\
& \qquad + \max_{t \in \mathcal{C}_k} \mathsf{RNA}(0^{\ell_4}\mathsf{CG}_\gamma(t)) \Big) \\
& + (3 + 2z) \max_{t \in \mathcal{C}_k, x \in \{\alpha,\beta,\gamma\}} |\mathsf{CG}_x(t)|. && \text{(remaining clique gadgets)}
\end{aligned}$$

In view of the calculation in Lemma 6, $|A|$ is at most

$$m_1 - z\ell_3 + \big(2\ell_2 + \min(\ell_{\mathsf{CLG},1}, \ell_{\mathsf{CNG},1}) + 2\ell_1 + \ell_{\mathsf{CLG},0} + \ell_{\mathsf{CNG},0}\big) + (3 + 2z) \max_{t,x} |\mathsf{CG}_x(t)|.$$

Due to the two facts (i) $2\ell_2 + \min(\ell_{\mathsf{CLG},1}, \ell_{\mathsf{CNG},1}) + 2\ell_1 + \ell_{\mathsf{CLG},0} + \ell_{\mathsf{CNG},0} < 0.1\ell_3$, and (ii) the length of a clique gadget $< 0.1\ell_3$, we have:

$$|A| < m_1 - 0.8z\ell_3 + 0.4\ell_3.$$

Thus, if $z > 0$, then $|A| < m_1 < \mathsf{RNA}(S_G)$, contradicting the assumption that A is optimal. Hence we must have $z = 0$, i.e., A satisfies (P1). □

Next, we deal with property (P2). We need some terminologies for ease of notation. For each $x \in \{\alpha, \beta, \gamma\}$, we call $\mathsf{CG}_x(t)$ a *type-x* clique gadget. We say that the two clique gadgets C_1 and C_2 are *linked* in A if $C_1 \overset{A}{\longleftrightarrow} C_2$. We write $\mathcal{M}_{\alpha,\beta,\gamma}$ to denote the set of all RNA foldings A of S_G satisfying (P1) and (P2). We write $\mathcal{M}_\alpha$ to denote the set of all RNA foldings A of S_G satisfying (P1) and the following condition (P2$'_\alpha$).

(P2$'_\alpha$) There exist $t_{\alpha,1}, t_{\alpha,2}, t_\beta, t_\gamma \in \mathcal{C}_k$ satisfying $t_{\alpha,1} \neq t_{\alpha,2}$ such that the following holds. If $\mathsf{CG}_{u_1}(t_1) \overset{A}{\longleftrightarrow} \mathsf{CG}_{u_2}(t_2)$, then $\{(u_1, t_1), (u_2, t_2)\} \in \{\{(\alpha, t_{\alpha,1}), (\beta, t_\beta)\}, \{(\alpha, t_{\alpha,2}), (\gamma, t_\gamma)\}\}$.

The two properties (P2$'_\beta$) and (P2$'_\gamma$), and the two sets $\mathcal{M}_\beta$ and $\mathcal{M}_\gamma$ are defined analogously.

Lemma 9. *Let A be an optimal RNA folding of S_G satisfying (P1). For each $x \in \{\alpha, \beta, \gamma\}$, there do not exist two distinct type-x clique gadgets C_1 and C_2 with $C_1 \overset{A}{\longleftrightarrow} C_2$.*

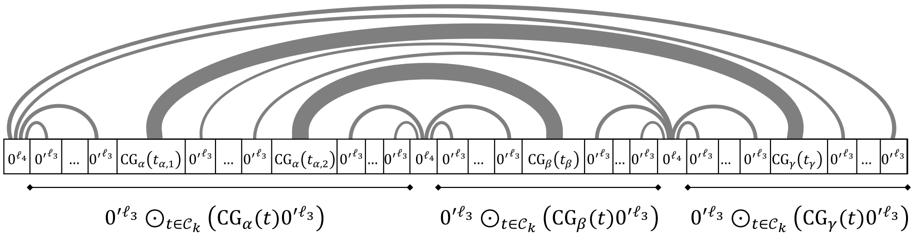

Fig. 4. Alignment in $\mathcal{M}_\alpha$.

Proof. There is a substring $S^\diamond = 0'^{\ell_3}$ located between C_1 and C_2. The existence of a pair in A linking a letter in C_1 and a letter in C_2 makes it impossible for any letter in $S^\diamond$ be matched to letters in any 0^{ℓ_4}, which is a contradiction to the assumption that A has property (P1). □

Lemma 10. *Let A be an optimal RNA folding of S_G satisfying (P1). For each $\{x, y\} \in \{\{\alpha, \beta\}, \{\alpha, \gamma\}, \{\beta, \gamma\}\}$, there do not exist two distinct type-x clique gadgets C_1 and C_2 and two not necessarily distinct type-y clique gadgets C_3 and C_4 such that $C_1 \overset{A}{\longleftrightarrow} C_3$ and $C_2 \overset{A}{\longleftrightarrow} C_4$.*

Proof. There is a substring $S^\diamond = 0'^{\ell_3}$ located between C_1 and C_2. Since $C_1 \overset{A}{\longleftrightarrow} C_3$ and $C_2 \overset{A}{\longleftrightarrow} C_4$, letters in $S^\diamond$ can only be matched to (i) letters in C_1, C_2, C_3, and C_4, (i) letters located between C_1 and C_2, and (iii) letters located between C_3 and C_4. This contradicts the assumption that A has property (P1). □

Lemma 11. *Let A be an optimal RNA folding of S_G satisfying (P1). Let $x \in \{\alpha, \beta, \gamma\}$. Suppose that there are two distinct type-x clique gadgets C_1 and C_2 such that $C_1 \overset{A}{\longleftrightarrow} C_3$ and $C_2 \overset{A}{\longleftrightarrow} C_4$, where C_3 and C_4 are two not necessarily distinct clique gadgets. Then $A \in \mathcal{M}_x$.*

Proof. Suppose that C_3 is a type-y clique gadget, and C_4 is a type-z clique gadget. By Lemma 9 and Lemma 10, the three symbols x, y, and z must be distinct, and so $C_3 \neq C_4$.

To prove that $A \in \mathcal{M}_x$, it suffices to show that A satisfies (P2$'_x$). Suppose that (P2$'_x$) is not met, then there are two clique gadgets C_5 and C_6 that are linked in A, and $\{C_5, C_6\} \notin \{\{C_1, C_3\}, \{C_2, C_4\}\}$. We show that this leads to a contradiction.

Observe that none of C_5 and C_6 can be a type-x clique gadget. Suppose that C_5 is of type-x. Then C_6 is either type-y or type-z by Lemma 9. In any case, Lemma 10 is violated. Therefore, without loss of generality, we assume C_5 is of type-y. Then, by Lemma 9, C_6 must be of type-z.

Since C_1 and C_2 are distinct, there must be a substring $S^\diamond = 0'^{\ell_3}$ located between C_1 and C_2. Since C_1 is linked to a type-y clique gadget, and since C_2 is linked to a type-z clique gadget, letters in $S^\diamond$ can only be paired up with letters in the substring $S' = 0^{\ell_4}$ bordering both $0'^{\ell_3} \bigodot_{t\in\mathcal{C}_k} \left(\mathrm{CG}_y(t)0'^{\ell_3}\right)$ and $0'^{\ell_3} \bigodot_{t\in\mathcal{C}_k} \left(\mathrm{CG}_z(t)0'^{\ell_3}\right)$, viewing S_G as a circular string. However, the existence of a pair linking a letter in C_5 (which is of type-y) and a letter in C_6 (which is of type-z) implies that no letter in S' can be matched with a letter in $0'^{\ell_3}$ without a crossing. This contradicts the assumption that A has property (P1). □

Lemma 12 follows from Lemma 8 and Lemma 11. An illustration of alignment in $\mathcal{M}_\alpha$ can be found in Fig. 4 ($\mathcal{M}_\beta$ and $\mathcal{M}_\gamma$ are analogous). Note that an illustration of alignment in $\mathcal{M}_{\alpha,\beta,\gamma}$ is in Fig. 1.

Lemma 12. *There is an optimal RNA folding of S_G that belongs to $\mathcal{M}_\alpha \cup \mathcal{M}_\beta \cup \mathcal{M}_\gamma \cup \mathcal{M}_{\alpha,\beta,\gamma}$.*

Proof. By Lemma 8, we restrict our consideration to optimal RNA foldings having (P1), and let A be any such optimal RNA folding. There are two cases.

Case 1: For each $x \in \{\alpha, \beta, \gamma\}$, there is at most one type-x clique gadget that is linked to other clique gadgets. Then A satisfies (P2), and so $A \in \mathcal{M}_{\alpha,\beta,\gamma}$.

Case 2: For some $x \in \{\alpha, \beta, \gamma\}$, there are two distinct type-x clique gadgets that are linked to other clique gadgets. By Lemma 11, $A \in \mathcal{M}_x$. □

We are now in a position to prove the main lemma in this section, which shows that there is an optimal RNA folding satisfying (P1) and (P2).

Lemma 13. *There is an optimal RNA folding of S_G that belongs to $\mathcal{M}_{\alpha,\beta,\gamma}$.*

Proof. In view of Lemma 12, it suffices to show that for any $A \in \mathcal{M}_\alpha \cup \mathcal{M}_\beta \cup \mathcal{M}_\gamma$, we have $|A| < \mathsf{RNA}(S_G)$. Let $A \in \mathcal{M}_x$. Let $t_{x,1}, t_{x,2}, t_y, t_z \in \mathcal{C}_k$ and $\{y, z\} = \{\alpha, \beta, \gamma\} \setminus \{x\}$ be the parameters specified in the definition of $\mathcal{M}_x$. That is, if $\mathsf{CG}_{u_1}(t_1) \xleftrightarrow{A} \mathsf{CG}_{u_2}(t_2)$, then $\{(u_1, t_1), (u_2, t_2)\} \in \{\{(\alpha, t_{\alpha,1}), (\beta, t_\beta)\}, \{(\alpha, t_{\alpha,2}), (\gamma, t_\gamma)\}\}$. Each pair in A falls into one of the following cases.

Case 1: The pair links a letter $0'$ in some substring $0'^{\ell_3}$ to a letter 0 in some substring 0^{ℓ_4}. Note that there are exactly $3(|\mathcal{C}_k| + 1)\ell_3$ number of such pairs in A.

Case 2: The pair contains a letter in some $\mathsf{CG}_u(t)$, where $(u, t) \notin \{(x, t_{x,1}), (x, t_{x,2}), (y, t_y), (z, t_z)\}$. Observe that any letter in such $\mathsf{CG}_u(t)$ can only be matched to letters within the same clique gadget $\mathsf{CG}_u(t)$ or some substring 0^{ℓ_4}, and so the number of pairs in A that fall into this case is upper bounded by $(|\mathcal{C}_k| - 2) \max_{t \in \mathcal{C}_k} \mathsf{RNA}\left(0^{\ell_4}\mathsf{CG}_x(t)\right) + (|\mathcal{C}_k| - 1) \max_{t \in \mathcal{C}_k} \mathsf{RNA}\left(0^{\ell_4}\mathsf{CG}_y(t)\right) + (|\mathcal{C}_k| - 1) \max_{t \in \mathcal{C}_k} \mathsf{RNA}\left(0^{\ell_4}\mathsf{CG}_z(t)\right)$.

Case 3: The pair involves a letter in some $\mathsf{CG}_u(t)$, where $(u, t) \in \{(x, t_{x,1}), (x, t_{x,2}), (y, t_y), (z, t_z)\}$. The number of such pairs is upper bounded by $\max_{t,t' \in \mathcal{C}_k} \mathsf{RNA}\left(0^{\ell_4}\mathsf{CG}_x(t)0^{\ell_4}\mathsf{CG}_y(t')\right) + \max_{t,t' \in \mathcal{C}_k} \mathsf{RNA}\left(0^{\ell_4}\mathsf{CG}_x(t)0^{\ell_4}\mathsf{CG}_z(t')\right)$.

Using the above calculation and the formulas in Lemma 6, we can derive $|A| < m_1 + m_2$, as follows. Note that $m_2 \geq 6\ell_2 + 3\ell_1$.

$$|A| \leq \begin{cases} m_1 + 2\ell_2 + 4.2\ell_1 - \min(\ell_{\mathsf{CLG},1}, \ell_{\mathsf{CNG},1}) & \text{if } x = \alpha \\ m_1 + 6\ell_2 + 2.2\ell_1 - \ell_{\mathsf{CLG},0} - \ell_{\mathsf{CNG},0} & \text{if } x = \beta \\ m_1 + 6\ell_2 + 2.2\ell_1 & \text{if } x = \gamma \end{cases}$$

Note that $m_1 + m_2 \leq \mathsf{RNA}(S_G)$ by Lemma 5. □

3.4. Calculating $\mathsf{RNA}(S_G)$

In this section, we prove that $\mathsf{RNA}(S_G) = m_1 + m_2$, and finish the proof of Theorem 2. In view of Lemma 13, in the calculation of $\mathsf{RNA}(S_G)$, we can restrict our consideration to RNA foldings in $\mathcal{M}_{\alpha,\beta,\gamma}$. Based on the structural property of RNA foldings in $\mathcal{M}_{\alpha,\beta,\gamma}$, we first reduce the calculation of $\mathsf{RNA}(S_G)$ to the calculation of optimal RNA foldings of simpler strings.

Lemma 14. $\mathsf{RNA}(S_G) \leq m_1 + \max_{t_\alpha, t_\beta, t_\gamma \in \mathcal{C}_k} \mathsf{RNA}(0^{\ell_4}\mathsf{CG}_\alpha(t_\alpha)0^{\ell_4}\mathsf{CG}_\beta(t_\beta)0^{\ell_4}\mathsf{CG}_\gamma(t_\gamma))$.

Proof. By Lemma 13, there is an optimal RNA folding of S_G in $\mathcal{M}_{\alpha,\beta,\gamma}$. For any $A \in \mathcal{M}_{\alpha,\beta,\gamma}$, let $t_\alpha, t_\beta, t_\gamma \in \mathcal{C}_k$ be the three cliques in the definition of (P2). Each pair in A falls into one of the following cases.

Case 1: The pair links a letter $0'$ in some substring $0'^{\ell_3}$ to a letter 0 in some substring 0^{ℓ_4}. Note that there are exactly $3(|\mathcal{C}_k| + 1)\ell_3$ number of such pairs in A.

Case 2: The pair contains a letter in some $\mathsf{CG}_u(t)$, where $(u, t) \notin \{(\alpha, t_\alpha), (\beta, t_\beta), (\gamma, t_\gamma)\}$. Observe that any letter in such $\mathsf{CG}_u(t)$ can only be matched to letters within the same clique gadget $\mathsf{CG}_u(t)$ or some substring 0^{ℓ_4}, and so the number of pairs in A that fall into this case is upper bounded by $(|\mathcal{C}_k| - 1) \max_{t \in \mathcal{C}_k} \mathsf{RNA}(0^{\ell_4}\mathsf{CG}_\alpha(t)) + (|\mathcal{C}_k| - 1) \max_{t \in \mathcal{C}_k} \mathsf{RNA}(0^{\ell_4}\mathsf{CG}_\beta(t)) + (|\mathcal{C}_k| - 1) \max_{t \in \mathcal{C}_k} \mathsf{RNA}(0^{\ell_4}\mathsf{CG}_\gamma(t))$.

Case 3: The pair involves a letter in some $\mathsf{CG}_u(t)$, where $(u, t) \in \{(\alpha, t_\alpha), (\beta, t_\beta), (\gamma, t_\gamma)\}$. The number of such pairs is upper bounded by $\mathsf{RNA}(0^{\ell_4}\mathsf{CG}_\alpha(t_\alpha)0^{\ell_4}\mathsf{CG}_\beta(t_\beta)0^{\ell_4}\mathsf{CG}_\gamma(t_\gamma))$.

Applying the formulas in Lemma 6, we have $|A| = m_1 + \mathsf{RNA}(0^{\ell_4}\mathsf{CG}_\alpha(t_\alpha)0^{\ell_4}\mathsf{CG}_\beta(t_\beta)0^{\ell_4}\mathsf{CG}_\gamma(t_\gamma))$. Hence we conclude the proof. □

For any choices of three k-cliques $t_\alpha, t_\beta, t_\gamma \in \mathcal{C}_k$, we define:

$$S_{t_\alpha,t_\beta,t_\gamma} = 1^{\ell_2} \circ S_{t_\gamma,t_\alpha} \circ 1^{\ell_2} \circ S_{t_\alpha,t_\beta,} \circ 1'^{2\ell_2} \circ S_{t_\beta,t_\gamma},$$

where

$$S_{t_\gamma,t_\alpha} = 0^{\ell_1}\mathsf{CNG}(t_\gamma)p(\mathsf{CLG}(t_\alpha)^R)0'^{\ell_1},$$

$$S_{t_\alpha,t_\beta} = 0^{\ell_1}\mathsf{CNG}(t_\alpha)p(\mathsf{CLG}(t_\beta)^R)0'^{\ell_1},$$

$$S_{t_\beta,t_\gamma} = 0'^{\ell_1}p(\mathsf{CNG}(t_\beta))\mathsf{CLG}(t_\gamma)^R0^{\ell_1}.$$

Note that $S_{t_\alpha,t_\beta,t_\gamma}$ is simply a cyclic shift of the concatenation of $\mathsf{CG}_\alpha(t_\alpha)$, $\mathsf{CG}_\beta(t_\beta)$, and $\mathsf{CG}_\gamma(t_\gamma)$ after removing the sequences of 1s and 1′s at the beginning and the end of these clique gadgets. Lemma 17, together with Lemma 14, reduces the calculation of $\mathsf{RNA}(S_G)$ to the calculation of $\mathsf{RNA}(S_{t_\alpha,t_\beta,t_\gamma})$. Lemma 15 and Lemma 16 are auxiliary lemmas.

Lemma 15. *Let* $S = S_1 \circ S_2 \circ S_3 \in \{0, 1, 0', 1'\}^*$ *be a string, where the substring* S_2 *is either* $11'$ *or* $1'1$. *Then* $\mathsf{RNA}(S) = \mathsf{RNA}(S_1 \circ S_3) + 1$.

Proof. It suffices to show that there exists an optimal RNA folding of S where the two letters in S_2 are paired up. Let A be any optimal RNA folding of S. We show that it is possible to modify A in such a way that the two letters in S_2 are paired up, and the total number of matched pairs is unchanged.

Case 1: If the two letters in S_2 are already paired up, then no modification is needed.

Case 2: If exactly one of the two letters in S_2 is matched in A, we first unmatch it, and then we pair up the two letters in S_2.

Case 3: Suppose that both two letters in S_2 are matched to letters not in S_2. Suppose that the letter $1 \in S_2$ is paired up with x in A, and the letter $1' \in S_2$ is paired up with y in A. We remove these two pairs from A, and then we add the two pairs $\{x, y\}$ and $\{1, 1'\}$ to A. □

Lemma 16. *In any optimal RNA folding of* $0^{\ell_4}\mathsf{CG}_\alpha(t_\alpha)0^{\ell_4}\mathsf{CG}_\beta(t_\beta)0^{\ell_4}\mathsf{CG}_\gamma(t_\gamma)$, *no letter within the three substrings* 0^{ℓ_4} *is matched.*

Proof. For ease of notation, we write $S^\star$ to denote $0^{\ell_4}\mathsf{CG}_\alpha(t_\alpha)0^{\ell_4}\mathsf{CG}_\beta(t_\beta)0^{\ell_4}\mathsf{CG}_\gamma(t_\gamma)$. We first state a few simple observations.

(O1) By simply matching only the letters in $\mathsf{CG}_\alpha(t_\alpha), \mathsf{CG}_\beta(t_\beta)$, and $\mathsf{CG}_\gamma(t_\gamma)$, as described in Fig. 3, we infer that $\mathsf{RNA}(S^\star) \geq 6\ell_2 + 3\ell_1$.

(O2) The total number of $0'$ and $1'$ in $S^\star$ is at most $6\ell_2 + 3.1\ell_1$, and so $\mathsf{RNA}(S^\star) \leq 6\ell_2 + 3.1\ell_1$.

(O3) The absolute difference between the number of 1 and the number of $1'$ in $S^\star$ is at most $0.1\ell_1$.

Note that $\mathsf{CG}_\gamma(t_\gamma)$ does not contain $0'$. To prove the lemma, we show that in any optimal RNA folding A of $S^\star$, no letter in 0^{ℓ_4} is matched to letters in $\mathsf{CG}_\beta(t_\beta)$ and $\mathsf{CG}_\alpha(t_\alpha)$. We write S_1, S_2, and S_3 to denote the first, second, and the third substring 0^{ℓ_4}.

We show that there is no pair in A linking a letter $0' \in \mathsf{CG}_\beta(t_\beta)$ to a letter $0 \in S_2 \cup S_3$. Recall that $\mathsf{CG}_\beta(t_\beta) = 1'^{\ell_2} p(\mathsf{CLG}(t_\beta)^R)0'^{\ell_1}1'^{2\ell_2}0'^{\ell_1} p(\mathsf{CNG}(t_\beta))1'^{\ell_2}$. If there is such a pair, then at least ℓ_2 amount of $1'$ cannot participate in the RNA folding. Therefore, by (O2), $|A| \leq (6\ell_2 + 3.1\ell_1) - \ell_2 \leq 5\ell_2 + 3.1\ell_1$. However, by (O1), $\mathsf{RNA}(S^\star) \geq 6\ell_2 + 3\ell_1 > |A|$, contradicting with the assumption that A is optimal.

Next, we show that there is no pair in A linking a letter $0' \in \mathsf{CG}_\beta(t_\beta)$ to a letter $0 \in S_1$. Recall that $\mathsf{CG}_\alpha(t_\alpha) = 1'^{2\ell_2} p(\mathsf{CLG}(t_\alpha)^R)0'^{\ell_1}1^{\ell_2}0^{\ell_1}\mathsf{CNG}(t_\alpha)1^{\ell_2}$. Suppose that there is such a pair. Then the $3\ell_2$ amount of $1'$ in (i) the substring $1'^{2\ell_2}$ of $\mathsf{CG}_\alpha(t_\alpha)$ and (ii) the first substring $1'^{\ell_2}$ of $\mathsf{CG}_\beta(t_\beta)$ can only be matched to letters in $\mathsf{CG}_\alpha(t_\alpha)$. However, the amount of 1 in $\mathsf{CG}_\alpha(t_\alpha)$ is at most $2.1\ell_1$, and so at least $0.9\ell_2$ amount of $1'$ are not matched. Therefore, by (O1) and (O2), $|A| \leq (6\ell_2 + 3.1\ell_1) - 0.9\ell_2 < \mathsf{RNA}(S^\star)$, contradicting with the assumption that A is optimal.

Lastly, we show that there is no pair in A linking a letter $0' \in \mathsf{CG}_\alpha(t_\alpha)$ to a letter $0 \in S_1 \cup S_2 \cup S_3$. Suppose that there is such a pair. We show that at least ℓ_2 amount of $1'$ cannot participate in the RNA folding. Then, by (O1) and (O2), $|A| \leq (6\ell_2 + 3.1\ell_1) - \ell_2 < \mathsf{RNA}(S^\star)$, contradicting with the assumption that A is optimal. We divide the analysis into cases.

Case 1: A letter $0' \in \mathsf{CG}_\alpha(t_\alpha)$ is matched to a letter $0 \in S_1$. Then all letters in the substring $1'^{2\ell_2}$ of $\mathsf{CG}_\alpha(t_\alpha)$ cannot participate in the RNA folding.

Case 2: A letter $0' \in \mathsf{CG}_\alpha(t_\alpha)$ is matched to a letter $0 \in S_2$. Then all letters in the two substrings 1^{ℓ_2} of $\mathsf{CG}_\alpha(t_\alpha)$ can only be matched to letters within $p(\mathsf{CLG}(t_\alpha)^R)$. Therefore, at least $2\ell_2 - |p(\mathsf{CLG}(t_\alpha)^R)| > 2\ell_2 - 0.1\ell_1$ amount of 1 are unmatched. By (O3), the absolute difference between the number of 1 and the number of $1'$ in $S^\star$ is at most $0.1\ell_1$, and so at least $2\ell_2 - 0.2\ell_1 > \ell_2$ amount of $1'$ cannot participate in the RNA folding.

Case 3: A letter $0' \in \mathsf{CG}_\alpha(t_\alpha)$ is matched to a letter $0 \in S_3$. Then all $1'$ in $\mathsf{CG}_\beta(t_\beta)$ can only be matched to letters in $\mathsf{CG}_\alpha(t_\alpha)$. Observe that the number of $1'$ in $\mathsf{CG}_\beta(t_\beta)$ is at least ℓ_2 more than the number of 1 in $\mathsf{CG}_\alpha(t_\alpha)$, and so at least ℓ_2 amount of $1'$ cannot participate in the RNA folding. □

Lemma 17. $\mathsf{RNA}(0^{\ell_4}\mathsf{CG}_\alpha(t_\alpha)0^{\ell_4}\mathsf{CG}_\beta(t_\beta)0^{\ell_4}\mathsf{CG}_\gamma(t_\gamma)) = 4\ell_2 + \mathsf{RNA}(S_{t_\alpha,t_\beta,t_\gamma})$.

Proof. We bound $\mathsf{RNA}(0^{\ell_4}\mathsf{CG}_\alpha(t_\alpha)0^{\ell_4}\mathsf{CG}_\beta(t_\beta)0^{\ell_4}\mathsf{CG}_\gamma(t_\gamma))$ as follows.

$$\begin{aligned}
&\mathsf{RNA}(0^{\ell_4}\mathsf{CG}_\alpha(t_\alpha)0^{\ell_4}\mathsf{CG}_\beta(t_\beta)0^{\ell_4}\mathsf{CG}_\gamma(t_\gamma)) \\
&\quad = \mathsf{RNA}(\mathsf{CG}_\alpha(t_\alpha)\mathsf{CG}_\beta(t_\beta)\mathsf{CG}_\gamma(t_\gamma)) && \text{(Lemma 16)}
\end{aligned}$$

$$
\begin{aligned}
&= \mathsf{RNA}(1'^{2\ell_2} p(\mathsf{CLG}(t_\alpha)^R) 0'^{\ell_1} 1^{\ell_2} 0^{\ell_1} \mathsf{CNG}(t_\alpha) 1^{\ell_2} 1'^{\ell_2} p(\mathsf{CLG}(t_\beta)^R) 0'^{\ell_1} 1'^{2\ell_2} && \text{(by definition)}\\
&\quad 0'^{\ell_1} p(\mathsf{CNG}(t_\beta)) 1'^{\ell_2} 1^{\ell_2} \mathsf{CLG}(t_\gamma)^R 0^{\ell_1} 1^{\ell_2} 0^{\ell_1} \mathsf{CNG}(t_\gamma) 1^{2\ell_2})\\
&= \mathsf{RNA}(1^{\ell_2} 0^{\ell_1} \mathsf{CNG}(t_\gamma) 1^{2\ell_2} 1'^{2\ell_2} p(\mathsf{CLG}(t_\alpha)^R) 0'^{\ell_1} 1^{\ell_2} 0^{\ell_1} \mathsf{CNG}(t_\alpha) 1^{\ell_2} 1'^{\ell_2} && \text{(cyclic shift)}\\
&\quad p(\mathsf{CLG}(t_\beta)^R) 0'^{\ell_1} 1'^{2\ell_2} 0'^{\ell_1} p(\mathsf{CNG}(t_\beta)) 1'^{\ell_2} 1^{\ell_2} \mathsf{CLG}(t_\gamma)^R 0^{\ell_1})\\
&= 4\ell_2 + \mathsf{RNA}(1^{\ell_2} 0^{\ell_1} \mathsf{CNG}(t_\gamma) p(\mathsf{CLG}(t_\alpha)^R) 0'^{\ell_1} 1^{\ell_2} 0^{\ell_1} \mathsf{CNG}(t_\alpha) p(\mathsf{CLG}(t_\beta)^R) 0'^{\ell_1} && \text{(Lemma 15)}\\
&\quad 1'^{2\ell_2} 0'^{\ell_1} p(\mathsf{CNG}(t_\beta)) \mathsf{CLG}(t_\gamma)^R 0^{\ell_1})\\
&= 4\ell_2 + \mathsf{RNA}(S_{t_\alpha,t_\beta,t_\gamma}). && \text{(by definition)}
\end{aligned}
$$

For the third equality, we move $1^{\ell_2} 0^{\ell_1} \mathsf{CNG}(t_\gamma) 1^{2\ell_2}$ from the end of the sequence to the beginning. The fourth equality follows by applying Lemma 15 iteratively to remove the substrings $1^{2\ell_2} 1'^{2\ell_2}$, $1^{\ell_2} 1'^{\ell_2}$, and $1'^{\ell_2} 1^{\ell_2}$. □

Lemma 18 shows that $\mathsf{RNA}(S_G) = m_1 + m_2$ by calculating the exact value of $\mathsf{RNA}(S_{t_\alpha,t_\beta,t_\gamma})$.

Lemma 18. $\mathsf{RNA}(S_G) = m_1 + m_2$.

Proof. By Lemma 5, we already have $\mathsf{RNA}(S_G) \geq m_1 + m_2$. By Lemma 14 and Lemma 17, we have $\mathsf{RNA}(S_G) \leq m_1 + 4\ell_2 + \max_{t_\alpha,t_\beta,t_\gamma \in \mathcal{C}_k} \mathsf{RNA}(S_{t_\alpha,t_\beta,t_\gamma})$. Thus, to prove the lemma, it suffices to show that $\mathsf{RNA}(S_{t_\alpha,t_\beta,t_\gamma}) = 2\ell_2 + 3\ell_1 + \frac{3}{2}\ell_0 - Q(t_\alpha, t_\beta, t_\gamma)$. Recall that $Q(t_\alpha, t_\beta, t_\gamma)$ is defined as $\frac{1}{2}\big(\delta_{\mathsf{LCS}}(\mathsf{CLG}(t_\alpha), \mathsf{CNG}(t_\beta)) + \delta_{\mathsf{LCS}}(\mathsf{CLG}(t_\alpha), \mathsf{CNG}(t_\gamma)) + \delta_{\mathsf{LCS}}(\mathsf{CLG}(t_\beta), \mathsf{CNG}(t_\gamma))\big)$.

First of all, we calculate a simple lower bound on $\mathsf{RNA}(S_{t_\alpha,t_\beta,t_\gamma})$. Pairing up letters not residing in clique node gadgets and clique list gadgets yields an RNA folding of $S_{t_\alpha,t_\beta,t_\gamma}$ with cardinality $2\ell_2 + 3\ell_1$, and so $\mathsf{RNA}(S_{t_\alpha,t_\beta,t_\gamma}) \geq 2\ell_2 + 3\ell_1$. We claim that for any optimal RNA folding A of $S_{t_\alpha,t_\beta,t_\gamma}$, the following two statements are true.

(S1) For each of the two substrings 1^{ℓ_2}, there is a letter 1 paired up with a letter 1′ in the substring $1'^{2\ell_2}$.
(S2) For each $S' \in \{S_{t_\gamma,t_\alpha}, S_{t_\alpha,t_\beta}, S_{t_\beta,t_\gamma}\}$, there is a pair linking a letter 0′ in $0'^{\ell_1} \subseteq S'$ and a letter 0 in $0^{\ell_1} \subseteq S'$.

To prove the statement (S1), suppose that a substring 1^{ℓ_2} does not have any letter matched to a letter in the substring $1'^{2\ell_2}$. We show that this leads to a contradiction. Observe that the number of 1′ in $S_{t_\alpha,t_\beta,t_\gamma}$ that does not belong to $1'^{2\ell_2}$ is at most $0.1\ell_1$. Thus, $|A|$ is at most the total number of 0′ plus the total number of 1 minus $(\ell_2 - 0.1\ell_1)$. By a simple calculation, $|A| \leq (3\ell_1 + 0.1\ell_1) + (2\ell_2 + 0.1\ell_1) - (\ell_2 - 0.1\ell_1) = \ell_2 + 3.3\ell_1 < 2\ell_2 + 3\ell_1$, contradicting with the known lower bound of $\mathsf{RNA}(S_{t_\alpha,t_\beta,t_\gamma})$.

To prove the statement (S2) suppose that there is a substring $S' \in \{S_{t_\gamma,t_\alpha}, S_{t_\alpha,t_\beta}, S_{t_\beta,t_\gamma}\}$ that has no pair linking a letter 0′ in $0'^{\ell_1} \subseteq S'$ and a letter 0 in $0^{\ell_1} \subseteq S'$. Due to (S1), any pair in A involving letters in $0'^{\ell_1} \subseteq S'$ or $0^{\ell_1} \subseteq S'$ are confined to be within S'. Therefore, the number of pairs in A involving letters in S' is at most $|S| - 2\ell_1 \leq 0.1\ell_1$. This is certainly not optimal, since simply matching all 0′ in $0'^{\ell_1}$ to all 0 in 0^{ℓ_1} gives us ℓ_1 amount of pairs.

We can infer from the above two statements that for each $S' \in \{S_{t_\gamma,t_\alpha}, S_{t_\alpha,t_\beta}, S_{t_\beta,t_\gamma}\}$, letters within S' are only matched to letters within S in any optimal RNA folding of $S_{t_\alpha,t_\beta,t_\gamma}$. As a result,

$$
\begin{aligned}
\mathsf{RNA}(S_{t_\alpha,t_\beta,t_\gamma}) &= \mathsf{RNA}(1^{\ell_2} \circ 1^{\ell_2} \circ 1'^{2\ell_2}) + \mathsf{RNA}(S_{t_\gamma,t_\alpha}) + \mathsf{RNA}(S_{t_\alpha,t_\beta}) + \mathsf{RNA}(S_{t_\beta,t_\gamma})\\
&= 2\ell_2 + 3\ell_1 + \mathsf{RNA}(\mathsf{CNG}(t_\gamma) p(\mathsf{CLG}(t_\alpha)^R)) + \mathsf{RNA}(\mathsf{CNG}(t_\alpha) p(\mathsf{CLG}(t_\beta)^R))\\
&\quad + \mathsf{RNA}(p(\mathsf{CNG}(t_\beta)) \mathsf{CLG}(t_\gamma)^R)\\
&= 2\ell_2 + 3\ell_1 + \frac{3}{2}\ell_0 - \frac{1}{2}\big(\delta_{\mathsf{LCS}}(\mathsf{CLG}(t_\alpha), \mathsf{CNG}(t_\beta)) + \delta_{\mathsf{LCS}}(\mathsf{CLG}(t_\alpha), \mathsf{CNG}(t_\gamma))\\
&\quad + \delta_{\mathsf{LCS}}(\mathsf{CLG}(t_\beta), \mathsf{CNG}(t_\gamma))\big). \qquad \square
\end{aligned}
$$

We are ready to prove Theorem 2.

Proof of Theorem 2. Throughout the proof, k is treated as a constant. Given a graph G, we construct the string S_G. According to Lemma 1 and Lemma 4, the length of S_G is $O(n^{k+1} \log n)$, and S_G can be constructed in time $O\left(n^{k+1} \log n\right)$. We let $t_\alpha, t_\beta, t_\gamma \in \mathcal{C}_k$ be chosen to minimize

$$
Q(t_\alpha, t_\beta, t_\gamma) = \frac{1}{2}\big(\delta_{\mathsf{LCS}}(\mathsf{CLG}(t_\alpha), \mathsf{CNG}(t_\beta)) + \delta_{\mathsf{LCS}}(\mathsf{CLG}(t_\alpha), \mathsf{CNG}(t_\gamma)) + \delta_{\mathsf{LCS}}(\mathsf{CLG}(t_\beta), \mathsf{CNG}(t_\gamma))\big).
$$

By Lemma 3, there exists a number c_1 meeting the following conditions.

- The number c_1 depends only on n and k, and $Q(t_\alpha, t_\beta, t_\gamma) \geq 3c_1/2$.
- If $Q(t_\alpha, t_\beta, t_\gamma) = 3c_1/2$, then each of $t_\alpha \cup t_\beta$, $t_\alpha \cup t_\gamma$, and $t_\beta \cup t_\gamma$ is a $2k$-clique; in other words, $t_\alpha \cup t_\beta \cup t_\gamma$ is a $3k$-clique.
- If $Q(t_\alpha, t_\beta, t_\gamma) > 3c_1/2$, then the graph has no $3k$-clique.

According to Lemma 18, $\mathsf{RNA}(S_G) = m_1 + m_2$. By its definition, m_1 only depends on n and k; and $m_2 = 6\ell_2 + 3\ell_1 + \frac{3}{2}\ell_0 - \min_{t_\alpha, t_\beta, t_\gamma \in \mathcal{C}_k} Q(t_\alpha, t_\beta, t_\gamma)$. Hence we are able to infer whether G has a $3k$-clique from the value of $\mathsf{RNA}(S_G)$, which can be calculated in time $T\left(O\left(n^{k+1}\log n\right)\right)$. □

4. Hardness of Dyck edit distance problem

In this section, we consider the Dyck edit distance problem. The goal of this section is to present a simple reduction from RNA folding problem (with alphabet size 4) to Dyck edit distance problem (with alphabet size 10).

Dyck edit distance. Recall that the Dyck edit distance problem asks for the minimum number of edits to transform a given string to well-balanced parentheses of s different types. The formal definition of the problem is as follows. Given $S \in (\Sigma \cup \Sigma')^n$, the goal of the Dyck edit distance problem is to find a minimum number of edit operations (insertion, deletion, and substitution) that transform S into a string in the Dyck context free language.

Given Σ and its corresponding Σ', the Dyck context free language is defined by the grammar with following production rules: $\mathbf{S} \rightarrow \mathbf{SS}$, $\forall x \in \Sigma, \mathbf{S} \rightarrow x\mathbf{S}x'$, and $\mathbf{S} \rightarrow \epsilon$ (empty string). Note that for each $x \in \Sigma$, the two symbols x and x' represent one type of parenthesis.

An alternate formulation. An alternative definition of the Dyck edit distance problem is as follows. Given a sequence $S \in (\Sigma \cup \Sigma')^n$, find a minimum cost set $A \subseteq \{(i, j) \mid 1 \leq i < j \leq n\}$ satisfying the following conditions:

- $A = A_M \cup A_S$ has no crossing pair.
- A_M contains only pairs of the form (x, x'), $x \in \Sigma$ (i.e. for all $(i, j) \in A_M$, we have $S[i] = x$, $S[j] = x'$, for some $x \in \Sigma$). A_M corresponds to the set of matched pairs.
- A_S does not contain any pair of the form (y', x), $x, y \in \Sigma$ (i.e. for all $(i, j) \in A_S$ we have either $S[i] \in \Sigma$ or $S[j] \in \Sigma'$). A_S corresponds to the set of pairs that can be fixed by one substitution operation per each pair.
- Let D be the set of letters in S that do not belong to any pair in A. Each letter in D requires one deletion/insertion operation to fix.

The cost of A is then defined as $|A_S| + |D|$, and the Dyck edit distance of the string S is the cost of a minimum cost set meeting the above conditions.

Discussion. Dyck edit distance problem can be thought of as an asymmetric version of the RNA folding problem that also handles substitution, in addition to deletion and insertion. Though these two problems look similar, they can behave quite differently. For example, in Section 1 we describe a simple reduction from LCS to RNA folding; since LCS is basically the edit distance problem without substitution, one might feel that the same reduction also reduces the edit distance problem to the Dyck edit distance problem. However, this is not true. The following example shows that the edit distance between two strings X and Y cannot be inferred from the Dyck edit distance of $X \circ p(Y^R)$. Both the two strings $X_1 = ababa$ and $X_2 = abbaa$ require 4 edit operations to transform into the string $Y = caaac$; but the Dyck edit distance of $X_1 \circ p(Y^R) = ababac'a'a'a'c'$ is 4 (by deleting all b and c'), while the Dyck edit distance of $X_2 \circ p(Y^R) = abbaac'a'a'a'c'$ is 3 (by deleting all c' and substituting the second b with b').

Intuitively, the substitution operation makes Dyck edit distance more complicated than RNA folding. Indeed, the conditional lower bound for Dyck edit distance shown in [15] requires a big alphabet size of 48 and a longer proof. In the remainder of this section, we prove Theorem 3 by demonstrating a simple reduction from RNA folding problem (with alphabet size 4) to Dyck edit distance problem (with alphabet size 10). This improves upon the hardness result in [15], and justifies the intuition that Dyck edit distance is at least as hard as RNA folding.

Proof of Theorem 3. For notational simplicity, we let the alphabet for the RNA folding problem be $\Sigma \cup \Sigma' = \{0, 0', 1, 1'\}$ instead of $\{A, C, G, U\}$. Let S be any string in $(\Sigma \cup \Sigma')^n$. We define the string S_{Dyck} as the result of applying the following operations on S:

- Replace each letter 0 with the sequence $S_0 = aeb'aeb'$.
- Replace each letter $0'$ with the sequence $S_{0'} = bba'a'$.
- Replace each letter 1 with the sequence $S_1 = ced'ced'$.
- Replace each letter $1'$ with the sequence $S_{1'} = ddc'c'$.

The sequence S_{Dyck} is of length at most $6n$ on the alphabet $\{a, b, c, d, e\} \cup \{a', b', c', d', e'\}$, though the letter e' is not used. We claim that the Dyck edit distance of S_{Dyck} is $\frac{|S_{\text{Dyck}}|}{2} - 2\mathsf{RNA}(S)$.

Upper bound. We show that the Dyck edit distance of S_{Dyck} is at most $\frac{|S_{\text{Dyck}}|}{2} - 2\text{RNA}(S)$. Given an optimal RNA folding of S, we construct a crossing-free matching A with cost $\frac{|S_{\text{Dyck}}|}{2} - 2\text{RNA}(S)$ as follows.

For matched pairs in the RNA folding of S:

- For each matched pair $(0, 0')$ in the RNA folding of S, we add two pairs $(a, a'), (a, a')$ to A_M, and add three pairs $(e, b'), (e, b'), (b, b)$ to A_S in its corresponding pair of substrings $(S_0 = \mathbf{a}(eb')\mathbf{a}(eb'), S_{0'} = (bb)\mathbf{a}'\mathbf{a}')$ in S_{Dyck}.
- For each matched pair $(0', 0)$ in the RNA folding of S, we add two pairs $(b, b'), (b, b')$ to A_M, and add three pairs $(a', a'), (a, e), (a, e)$ to A_S in its corresponding pair of substrings $(S_{0'} = \mathbf{bb}(a'a'), S_0 = (ae)\mathbf{b}'(ae)\mathbf{b}')$ in S_{Dyck}.
- Similarly, for each matched pair $(1, 1'), (1', 1)$ in the RNA folding of S, we add two pairs to A_M and three pairs to A_S.

For unmatched letters in S:

- For each unmatched letter 0 in S, we add three pairs $(a, b'), (e, b'), (a, e)$ to A_S in its corresponding substring $S_0 = (a(eb')(ae)b')$. Similarly, for each unmatched letter 1, we add three pairs to A_S.
- For each unmatched letter $0'$ in S, we add two pairs $(b, b), (a', a')$ to A_S in its corresponding substring $S_0 = (bb)(a'a')$. Similarly, for each unmatched letter $1'$, we add two pairs to A_S.

The set A_M has size $2\text{RNA}(S)$, the set A_S has size $\frac{|S_{\text{Dyck}}|-4\text{RNA}(S)}{2}$, and D is an empty set. Therefore, the cost of A is $\frac{|S_{\text{Dyck}}|-4\text{RNA}(S)}{2} = \frac{|S_{\text{Dyck}}|}{2} - 2\text{RNA}(S)$.

Lower bound. We show that the Dyck edit distance of S_{Dyck} is at least $\frac{|S_{\text{Dyck}}|}{2} - 2\text{RNA}(S)$. Given a crossing-free matching A (on the string S_{Dyck}) of cost C, we recover an RNA folding of S that has $\geq \frac{|S_{\text{Dyck}}|}{4} - \frac{C}{2}$ number of matched pairs.

We build a multi-graph $G = (V, E)$ such that V is the set of all substrings S_0, $S_{0'}$, S_1, and $S_{1'}$ that constitute S_{Dyck}, and the number of edges between two substrings in V is the number of pairs in A_M linking letters between these two substrings. Note that $|V| = n$ and $|E| = A_M$. It is clear that $C \geq \frac{|S_{\text{Dyck}}|-2|E|}{2}$, since $|A_S| + |D| \geq \frac{|S_{\text{Dyck}}|-2|A_M|}{2} = \frac{|S_{\text{Dyck}}|-2|E|}{2}$. We show that we can obtain an RNA folding of S that has size $\geq \frac{|E|}{2}$. Note that $\frac{|E|}{2} \geq \frac{|S_{\text{Dyck}}|}{4} - \frac{C}{2}$. We make the following three observations.

(O1) G has degree at most 2. The reason is that at most two letters in each substring S_0, $S_{0'}$, S_1, $S_{1'}$ can participate in pairings of the form (x, x'), $x \in \{a, b, c, d\}$, without crossing.

(O2) In the graph G, each edge either (i) links a substring S_0 with a substring $S_{0'}$, or (ii) links a substring S_1 with a substring $S_{1'}$. The reason is that any pairing of the form (x, x'), $x \in \{a, b, c, d\}$, must be made between S_0 and $S_{0'}$, or between S_1 and $S_{1'}$.

(O3) G does not contain any cycle of odd length. This is due to (O2).

In view of (O2), a (graph-theoretic) matching $M \subseteq E$ of G naturally corresponds to a size-$|M|$ RNA folding of S, as follows. For each edge, which is a pair of substrings in S_{Dyck}, in M, we add its corresponding pair of letters in S to the RNA folding. By (O1) and (O3), G in a graph of maximum degree 2 without odd cycles, and a maximum matching in such a graph has size at least $\frac{|E|}{2}$, and so we conclude the proof. □

The reason that the letter e is essential in the proof is briefly explained as follows. Suppose that e is removed. For each matched pair $(0, 0')$ in the RNA folding of S, after adding two pairs (a, a') and (a, a') to A_M, the letter b' between two letters a in $S_0 = ab'ab'$ cannot participate in any matching. Hence some letters have to be in D according to our construction of the crossing-free matching A, which implies that our construction might not be optimal.

Consider the case $S = (00'0')$. We would have $S_{\text{Dyck}} = ab'ab'bba'a'bba'a'$ after removing e. If we match the two pairs (a, a') and (a, a') in $\mathbf{a}b'\mathbf{a}b'bb\mathbf{a}'\mathbf{a}'bba'a'$, then the cost will be at least 5 (three substitutions and two deletions are needed). However, there is a solution that uses only 4 substitutions: $\mathbf{a}(b'\mathbf{a}(b'(bb)a')\mathbf{a}'(bb)a')\mathbf{a}'$.

Note that if substitution is not allowed in the definition of Dyck edit distance, then the letter e in the above proof is not needed, and this lowers the alphabet size requirement from 10 to 8.

5. Conclusion

In this paper we present a conditional lower bound of RNA folding problem with alphabet size 4, and demonstrate a simple reduction from RNA folding problem to Dyck edit distance problem. One open problem that still remains is whether it is possible to reduce Dyck edit distance problem to RNA folding problem (i.e., the reverse of Theorem 3). The “standard” RNA folding problem only finds an optimal pseudoknot-free fold for an RNA sequence; however, the “real world” RNA folding includes pseudoknots, and is more complicated. There are variants of RNA folding problem that consider pseudoknots; see e.g., [23] and the citations therein. It would be interesting to see whether the techniques presented in this paper and [15,13] can be adapted to provide meaningful lower bounds for these problems.

Acknowledgements

The author thanks Seth Pettie for helpful discussions and comments.

References

[1] Y.-J. Chang, Hardness of RNA folding problem with four symbols, in: R. Grossi, M. Lewenstein (Eds.), Proceedings of 27th Annual Symposium on Combinatorial Pattern Matching, CPM, in: Leibniz International Proceedings in Informatics (LIPIcs), vol. 54, Schloss Dagstuhl–Leibniz-Zentrum Fuer Informatik, Dagstuhl, Germany, 2016, pp. 13:1–13:12, URL http://drops.dagstuhl.de/opus/volltexte/2016/6089.

[2] Y. Frid, D. Gusfield, A simple, practical and complete $O(\frac{n^3}{\log n})$-time algorithm for RNA folding using the four-Russians speedup, Algorithms Mol. Biol. 5 (1) (2010) 13, https://doi.org/10.1186/1748-7188-5-13.

[3] R. Durbin, S.R. Eddy, A. Krogh, G. Mitchison, Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids, Cambridge University Press, 1998.

[4] T. Akutsu, Approximation and exact algorithms for RNA secondary structure prediction and recognition of stochastic context-free languages, J. Comb. Optim. 3 (2) (1999) 321–336, https://doi.org/10.1023/A:1009898029639.

[5] T. Pinhas, D. Tsur, S. Zakov, M. Ziv-Ukelson, Edit distance with duplications and contractions revisited, in: R. Giancarlo, G. Manzini (Eds.), Combinatorial Pattern Matching, CPM, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 441–454.

[6] T. Pinhas, S. Zakov, D. Tsur, M. Ziv-Ukelson, Efficient edit distance with duplications and contractions, Algorithms Mol. Biol. 8 (1) (2013) 27, https://doi.org/10.1186/1748-7188-8-27.

[7] B. Venkatachalam, D. Gusfield, Y. Frid, Faster algorithms for RNA-folding using the four-Russians method, in: A. Darling, J. Stoye (Eds.), Algorithms in Bioinformatics, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 126–140.

[8] K. Bringmann, F. Grandoni, B. Saha, V.V. Williams, Truly sub-cubic algorithms for language edit distance and RNA-folding via fast bounded-difference min-plus product, in: Proceedings of IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS, 2016, pp. 375–384.

[9] R. Williams, Faster all-pairs shortest paths via circuit complexity, in: Proceedings of the 46th Annual ACM Symposium on Theory of Computing, STOC, ACM, New York, NY, USA, 2014, pp. 664–673, URL http://doi.acm.org/10.1145/2591796.2591811.

[10] M. Pătraşcu, R. Williams, On the possibility of faster SAT algorithms, in: Proceedings of the 21th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2010, pp. 1065–1075, URL http://dl.acm.org/citation.cfm?id=1873601.1873687.

[11] A. Abboud, A. Backurs, V.V. Williams, Tight hardness results for LCS and other sequence similarity measures, in: Proceedings of IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS, 2015, pp. 59–78.

[12] A. Backurs, P. Indyk, Edit distance cannot be computed in strongly subquadratic time (unless SETH is false), in: Proceedings of the 47th Annual ACM Symposium on Theory of Computing, STOC, ACM, New York, NY, USA, 2015, pp. 51–58, URL http://doi.acm.org/10.1145/2746539.2746612.

[13] K. Bringmann, M. Kunnemann, Quadratic conditional lower bounds for string problems and dynamic time warping, in: Proceedings of IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS, 2015, pp. 79–97.

[14] L. Roditty, V.V. Williams, Fast approximation algorithms for the diameter and radius of sparse graphs, in: Proceedings of the 45th Annual ACM Symposium on Theory of Computing, STOC, ACM, New York, NY, USA, 2013, pp. 515–524, URL http://doi.acm.org/10.1145/2488608.2488673.

[15] A. Abboud, A. Backurs, V.V. Williams, If the current clique algorithms are optimal, so is valiant's parser, in: Proceedings of IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS, 2015, pp. 98–117.

[16] F. Eisenbrand, F. Grandoni, On the complexity of fixed parameter clique and dominating set, Theoret. Comput. Sci. 326 (1) (2004) 57–67, https://doi.org/10.1016/j.tcs.2004.05.009, URL http://www.sciencedirect.com/science/article/pii/S030439750400372X.

[17] V. Vassilevska, Efficient algorithms for clique problems, Inform. Process. Lett. 109 (4) (2009) 254–257, https://doi.org/10.1016/j.ipl.2008.10.014, URL http://www.sciencedirect.com/science/article/pii/S0020019008003293.

[18] A. Amir, G.M. Landau, Fast parallel and serial multidimensional approximate array matching, Theoret. Comput. Sci. 81 (1) (1991) 97–115, https://doi.org/10.1016/0304-3975(91)90318-V, URL http://www.sciencedirect.com/science/article/pii/030439759190318V.

[19] A. Amir, T.M. Chan, M. Lewenstein, N. Lewenstein, On hardness of jumbled indexing, in: J. Esparza, P. Fraigniaud, T. Husfeldt, E. Koutsoupias (Eds.), Automata, Languages, and Programming, ICALP, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014, pp. 114–125.

[20] T.M. Chan, M. Lewenstein, Clustered integer 3SUM via additive combinatorics, in: Proceedings of the 47th Annual ACM Symposium on Theory of Computing, STOC, ACM, New York, NY, USA, 2015, pp. 31–40, URL http://doi.acm.org/10.1145/2746539.2746568.

[21] B. Saha, The Dyck language edit distance problem in near-linear time, in: Proceedings of IEEE 55th Annual Symposium on Foundations of Computer Science, FOCS, 2014, pp. 611–620.

[22] B. Saha, Language edit distance and maximum likelihood parsing of stochastic grammars: faster algorithms and connection to fundamental graph problems, in: Proceedings of IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS, 2015, pp. 118–135.

[23] Y. Frid, D. Gusfield, Speedup of RNA pseudoknotted secondary structure recurrence computation with the four-Russians method, in: G. Lin (Ed.), Combinatorial Optimization and Applications, COCOA, Springer, Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 176–187.