ELSEVIER

Discrete Applied Mathematics 104 (2000) 45–62

DISCRETE
APPLIED
MATHEMATICS

Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots[☆], [☆☆]

Tatsuya Akutsu

Human Genome Center, Institute of Medical Science, University of Tokyo, 4-6-1 Shirokanedai, Minato-ku, Tokyo 108-8639, Japan

Abstract

This paper shows simple dynamic programming algorithms for RNA secondary structure prediction with pseudoknots. For a basic version of the problem (i.e., maximizing the number of base pairs), this paper presents an $O(n^4)$ time exact algorithm and an $O(n^{4-\delta})$ time approximation algorithm. The latter one outputs, for most RNA sequences, a secondary structure in which the number of base pairs is at least $1-\varepsilon$ of the optimal, where ε, δ are any constants satisfying $0<\varepsilon, \delta<1$. Several related results are shown too. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: RNA secondary structure; Pseudoknot; Approximation algorithms; Computational biology; Dynamic programming

1. Introduction

The problem of RNA secondary structure prediction is, given an RNA sequence of length n, to compute its correct *secondary structure* (an out-planar like structure) [13,15,17]. Although it is still hard to compute (nearly) correct structures for all sequences, several methods have been developed. In most such methods, RNA secondary structure prediction is defined as an energy minimization problem, in which an *optimal secondary structure* (i.e., a secondary structure with minimum free energy) is to be computed.

For RNA secondary structure prediction *without pseudoknots*, a lot of studies have been done both from a practical viewpoint and from a theoretical viewpoint. Waterman and Smith, and Zuker and Stiegler proposed simple *dynamic programming* algorithms [16,18]. The time complexities of those dynamic programming algorithms were $O(n^3)$

[☆] A preliminary version appeared in the Seventh Workshop on Genome Informatics (GIW'97).

[☆☆] Partially supported by the Ohkawa foundation for information and telecommunications and by a Grant-in-Aid "Genome Science" from The Ministry of Education, Science, Sports and Culture in Japan.

E-mail address: takutsu@ims.u-tokyo.ac.jp (T. Akutsu).

PII: S0166-218X(00)00186-4

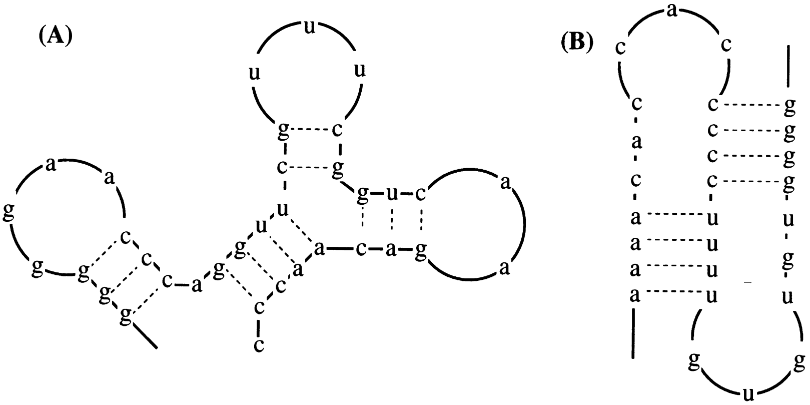

Fig. 1. Examples of RNA secondary structures: (A) structure without pseudoknots; (B) pseudoknot.

if we ignore the destabilizing energy due to loop regions, otherwise they were $\Omega(n^4)$. Although only a slight improvement had been done on global free energy minimization [3], several important improvements have been done for finding locally stabilizing substructures, where various types of destabilizing energy functions were considered [5,15].

For RNA secondary structure prediction *with pseudoknots* (see Fig. 1), several studies have been done from a practical viewpoint. Abrahams *et al.* developed a local search method [1], Akiyama and Kanehisa developed a method based on the Hopfield network [2], Brown and Wilson [4] developed a method based on the stochastic context-free grammar. However, these methods are not guaranteed to find optimal structures. On the other hand, only a few studies have been done from a theoretical viewpoint. Brown and Wilson considered a method based on an extension of the stochastic context-free grammar and mentioned that its time complexity was $\Omega(n^5)$ [4]. Uemura *et al.* proposed algorithms based on *tree adjoining grammar* [14]. The time complexities of their algorithms depend on types of pseudoknots: it is $O(n^4)$ for simple pseudoknots and $O(n^5)$ or more for the other pseudoknots. Although their algorithms can always find optimal structures, tree adjoining grammar is complicated and hard to understand and thus the proposed algorithms are not simple. Thus, we analyzed their method and we found that tree adjoining grammar was not crucial but the parsing procedure was crucial. Since the parsing procedure is intrinsically a dynamic programming procedure, we re-formulate their method as a dynamic programming procedure without tree adjoining grammar.

In this paper, we first consider a basic version of RNA secondary structure with simple pseudoknots: maximizing the number of base pairs where simple pseudoknots may appear. For this version, we show a simple $O(n^4)$ time dynamic programming algorithm. Although the time complexity is not improved from the previous one [14], it is much simpler, it is easier to understand, it is easier to modify, and it is easier to cope with various score functions (i.e., free energy functions). Next, we apply a technique developed in [3] to the proposed algorithm and obtain an $\mathrm{O}(n^{4-\delta})$ time approximation algorithm that computes, for most RNA sequences, a secondary structure whose score (the number of base pairs) is at least $1-\varepsilon$ of the optimal, where ε, δ are any fixed constant such that $0<\varepsilon, \delta<1$. Note that application of the technique is not trivial and an additional idea is introduced here. Then, we show various extensions of these algorithms. On the other hand, we show a hardness result for generalized pseudoknots, where we do not know whether or not such a structure exists in real RNAs.

Because of the high time complexity as in Ref. [14], the proposed algorithms are not yet practical. However, they might be made practical if some heuristics are combined with them. Some idea for such practical improvement is discussed in Section 7. Moreover, we emphasize that the most important contribution of this paper is that it corrects the previous misunderstanding that pseudoknots cannot be handled by a simple dynamic-programming-based approach, and it shows that the dynamic-programming-based approach is still useful for secondary structure prediction with pseudoknots.

2. Preliminaries

Here, we introduce basic versions of RNA secondary structure prediction, in which the score is defined as the total number of base pairs appearing in a secondary structure. Although the score is an extremely simple estimate of the free energy, the problems and the algorithms are to be extended for more realistic scores in Section 5.

2.1. Secondary structure without pseudoknots

Let $A = a_1 a_2 \ldots a_n$ be an *RNA sequence*. That is, A is a string over an alphabet $\Sigma = \{\mathtt{a}, \mathtt{u}, \mathtt{g}, \mathtt{c}\}$. A pair of residues (letters) (x, y) is called a (*complementary*) *base pair* if $\{x, y\} = \{\mathtt{a}, \mathtt{u}\}$ or $\{x, y\} = \{\mathtt{g}, \mathtt{c}\}$. Although the wobble pair $\{\mathtt{g}, \mathtt{u}\}$ [9] is not treated as a base pair, similar results hold for such a case. A set of pairs of indices

$$M = \{(i, j) \mid 1 \leqslant i < j \leqslant n, (a_i, a_j) \text{ is a base pair}\}$$

is called an *RNA secondary structure without pseudoknots* if no distinct pairs (a_i, a_j), (a_h, a_k) in M satisfy $i \leqslant h \leqslant j \leqslant k$ (see Fig. 1). The score of M is defined as the number of base pairs in M (i.e., $|M|$). Then, a *basic version of RNA secondary structure prediction without pseudoknots* is defined as follows: given an RNA sequence $A = a_1 a_2 \ldots a_n$, find an RNA secondary structure M with the maximum score. Such a structure is called an *optimal RNA secondary structure*, and its score is denoted by $OPT_0(A)$. In an RNA secondary structure M, a pair of consecutive subsequences $(a_i a_{i+1} \ldots a_{i+k}, a_{j-k} a_{j-k+1} \ldots a_j)$ is called a *stacked region* if (a_{i+h}, a_{j-h}) is a base pair in M for all $h \leqslant k$, and a consecutive subsequence $a_i a_{i+1} \ldots a_{i+k}$ is called a *loop region* if no a_{i+h} $(0 \leqslant h \leqslant k)$ appears in M.

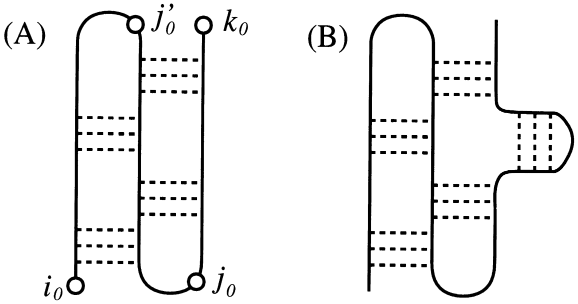

Fig. 2. (A) Simple pseudoknot, and (B) recursive pseudoknot.

It is well known that $OPT_0(A)$ can be computed in $O(n^3)$ time using the following simple dynamic programming procedure [15]:

$$S(i,j) = \max \begin{cases} S(i+1,j-1)+v(a_i,a_j), \\ \max\limits_{i<k\leqslant j}\{S(i,k-1)+S(k,j)\}, \end{cases}$$

where we let $S(i,j)=0$ for all $i\geqslant j$, and $v(x,y)=1$ if (x,y) is a base pair, otherwise $v(x,y)=0$. Note that $OPT_0(A)$ is given by $S(1,n)$. Note also that an optimal structure can be computed in $O(n^3)$ time from $S(i,j)$ using the *traceback* technique [15]. Similarly, we only describe the procedures for computing scores or free energies in this paper, *all of which can be modified for computing secondary structures without increasing the orders of the time complexities, using the traceback technique.*

2.2. Secondary structure with pseudoknots

Since there is no established definition of pseudoknot, we define it based on Refs. [1,4,14].

First, we define *simple pseudoknots* (see Fig. 2(A)). Consider a consecutive subsequence $a_{i_0}a_{i_0+1}\dots a_{k_0}$ of an RNA sequence A where i_0 and k_0 are arbitrarily chosen positions. We call a set of base pairs M_{i_0,k_0} a *simple pseudoknot* if there exist positions j_0, j_0' $(i_0<j_0'<j_0<k_0)$ for which the following conditions are satisfied:

- Each i $(i_0\leqslant i\leqslant k_0)$ appears at most once in M_{i_0,k_0}.
- Each $(i,j)\in M_{i_0,k_0}$ satisfies either $i_0\leqslant i<j_0'\leqslant j<j_0$ or $j_0'\leqslant i<j_0\leqslant j\leqslant k_0$.
- If pairs (i,j) and (i',j') in M_{i_0,k_0} satisfy either $i<i'<j_0'$ or $j_0'\leqslant i<i'$, then $j>j'$ holds.

Although usual substructures qualify as simple pseudoknots if either $\{(i,j)\in M_{i_0,k_0} \mid i<j_0'\}=\emptyset$ or $\{(i,j)\in M_{i_0,k_0} \mid j_0'\leqslant i\}=\emptyset$ holds, all the algorithms can be modified so that these cases are excluded without increasing the order of the time complexity.

Then, a set of base pairs M is called an *RNA secondary structure with simple pseudoknots* if the following conditions are satisfied:

- $M=M'\cup M_{i_1,k_1}\cup M_{i_2,k_2}\cup\cdots\cup M_{i_t,k_t}$ where t is a non-negative integer and $1\leqslant i_1<k_1<i_2<k_2<\cdots<i_t<k_t\leqslant n$.
- Each M_{i_h,k_h} is a simple pseudoknot for a consecutive subsequence $a_{i_h}a_{i_h+1}\dots a_{k_h}$.

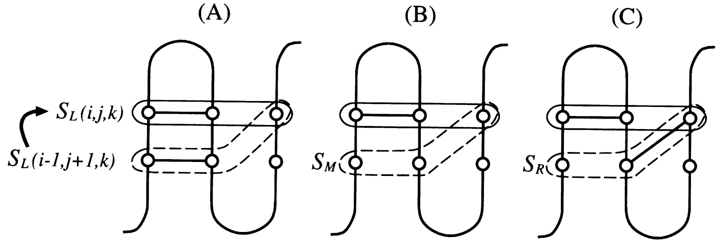

Fig. 3. Illustration of the recurrence used in the dynamic programming procedure: (A) corresponds to $S_L(i,j,k) = v(a_i,a_j) + S_L(i-1,j+1,k)$, (B) corresponds to $S_L(i,j,k) = v(a_i,a_j) + S_M(i-1,j+1,k)$, and (C) corresponds to $S_L(i,j,k) = v(a_i,a_j) + S_R(i-1,j+1,k)$.

- M' is a secondary structure without pseudoknots for a sequence A', where A' is obtained by deleting all $a_{i_h}a_{i_h+1}\dots a_{k_h}$ from A (i.e., $A' = a_1a_2\dots a_{i_1-1}a_{k_1+1}\dots a_{i_2-1}a_{k_2+1}\dots\dots a_{i_t-1}a_{k_t+1}\dots a_n$).

Note that a secondary structure without pseudoknots corresponds to the case of $t=0$ and qualifies as a secondary structure with simple pseudoknots.

As in Section 2.1, a *basic version of RNA secondary structure prediction with simple pseudoknots* is defined as a problem of finding an RNA secondary structure with simple pseudoknots that has the maximum score (i.e., the maximum number of base pairs). In this case, the score of an optimal structure is denoted by $OPT_1(A)$.

Generally, pseudoknots can have recursive structures (see Fig. 2(B)): any loop region can be replaced by another secondary structure (with/without pseudoknots). The definition of an RNA secondary structure with recursive pseudoknots is given by replacing “Each M_{i_h,k_h} is a simple pseudoknot” in the above definition with “Each M_{i_h,k_h} is a recursive pseudoknot”.

3. Dynamic programming algorithm for simple pseudoknots

In this section, we show a simple dynamic programming algorithm for the basic version of RNA secondary structure prediction with simple pseudoknots.

For finding a simple pseudoknot substructure whose endpoints are i_0th and k_0th residues, we consider triplets (i,j,k) $(i_0 - 1 \leqslant i < j \leqslant k \leqslant k_0)$ instead of (i,j) in Section 2.1. Moreover, we consider three types of triplets $S_L(i,j,k)$,$S_M(i,j,k)$, and $S_R(i,j,k)$. $S_L(i,j,k)$ (resp. $S_R(i,j,k)$) corresponds to a case where ith and jth (resp. jth and kth) residues make a base pair. Then, each triplet can be computed by the following recurrence (see Fig. 3):

$$S_L(i,j,k) = v(a_i,a_j) + \max \left\{ \begin{array}{c} S_L(i-1,j+1,k), \\ S_M(i-1,j+1,k), \\ S_R(i-1,j+1,k) \end{array} \right\}, \qquad (1)$$

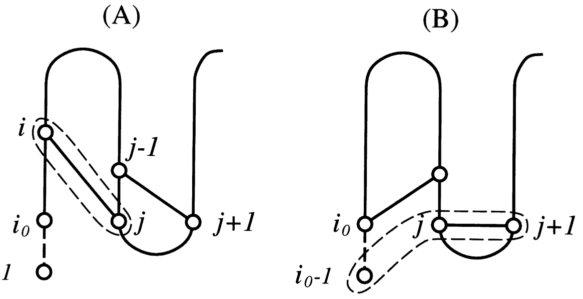

Fig. 4. Illustration of the initialization procedure: (A) corresponds to $S_L(i,j,j)=v(a_i,a_j)$ and (B) corresponds to $S_R(i_0-1,j,j+1)=v(a_j,a_{j+1})$.

$$S_R(i,j,k)=v(a_j,a_k)+\max\left\{\begin{array}{c} S_L(i,j+1,k-1), \\ S_M(i,j+1,k-1), \\ S_R(i,j+1,k-1) \end{array}\right\}, \tag{2}$$

$$S_M(i,j,k)=\max\left\{\begin{array}{l} S_M(i-1,j,k),\ S_M(i,j+1,k),\ S_M(i,j,k-1), \\ S_L(i-1,j,k),\ \ S_L(i,j+1,k), \\ S_R(i,j+1,k),\ \ S_R(i,j,k-1) \end{array}\right\}, \tag{3}$$

where $v(a_i,a_j)=1$ if (a_i,a_j) is a base pair, otherwise $v(a_i,a_j)=-\infty$, and the initialization is done by letting (see Fig. 4):

$$S_L(i,j,j)=v(a_i,a_j) \quad \text{for all } i<j,$$

$$S_R(i_0-1,j,j+1)=v(a_j,a_{j+1}) \quad \text{for all } j,$$

$$S_L(i_0-1,j,k)=S_R(i_0-1,j,k)=S_M(i_0-1,j,k)=0$$

$$\text{for the other } j,k \text{ satisfying } k=j \text{ or } k=j+1.$$

For each (fixed) pair (i_0, k_0), we compute the above scores and we obtain the score of a pseudoknot whose endpoints are (i_0, k_0) by

$$S_{\text{pseudo}}(i_0,k_0)=\max_{i_0\leqslant i<j<k\leqslant k_0}\{S_L(i,j,k),\ S_M(i,j,k),\ S_R(i,j,k)\}.$$

Finally, we compute the optimal score $S(1,n)$ for the whole structure by the following recurrence:

$$S(i,j)=\max\left\{S_{\text{pseudo}}(i,j),\ S(i+1,j-1)+v(a_i,a_j),\ \max_{i<k\leqslant j}\{S(i,k-1)+S(k,j)\}\right\}.$$

Theorem 1. *An RNA secondary structure with simple pseudoknots which has the maximum number of base pairs can be computed by a simple dynamic programming algorithm in* $O(n^4)$ *time using* $O(n^3)$ *space.*

Proof. First we show that an optimal structure is computed using the dynamic programming procedure shown above. Since $S_{\mathrm{pseudo}}(i_0, k_0)$ is newly introduced, it suffices to show that the procedure for $S_{\mathrm{pseudo}}(i_0, k_0)$ computes the score of an optimal simple pseudoknot for $a_{i_0}a_{i_0+1}\ldots a_{k_0}$.

As in pairwise sequence alignment [15], we associate the shortest path problem (precisely, the maximum weight path problem for acyclic directed graphs) with the dynamic programming procedure. We construct an acyclic directed graph $G(V,E)$ as follows. The vertex set V is defined by

$$V = \{v_s\} \cup \{v_L(i,j,k), v_R(i,j,k), v_M(i,j,k) \mid i_0 - 1 \leqslant i < j \leqslant k \leqslant k_0\},$$

where v_s is introduced as the start point. The edge set and the weights of edges are defined by the following rules:

- $(v_s, v_d(i,j,k)) \in E$ and $w(v_s, v_d(i,j,k)) = S_d(i,j,k)$ if $S_d(i,j,k)$ is used in the initialization where $d \in \{L,M,R\}$,
- $(v_d(i',j',k'), v_L(i,j,k)) \in E$ and $w(v_d(i',j',k'), v_L(i,j,k)) = v(a_i, a_j)$ if $v_d(i',j',k')$ and $v_L(i,j,k)$ appear in recurrence (1),
- $(v_d(i',j',k'), v_R(i,j,k)) \in E$ and $w(v_d(i',j',k'), v_R(i,j,k)) = v(a_j, a_k)$ if $v_d(i',j',k')$ and $v_R(i,j,k)$ appear in recurrence (2),
- $(v_d(i',j',k'), v_M(i,j,k)) \in E$ and $w(v_d(i',j',k'), v_M(i,j,k)) = 0$ if $v_d(i',j',k')$ and $v_M(i,j,k)$ appear in recurrence (3).

Let M_{i_0,k_0} be an arbitrary simple pseudoknot for sequence $a_{i_0}a_{i_0+1}\ldots a_{k_0}$. From the definition of a simple pseudoknot, there exist integers j_0' and j_0 (although (j_0', j_0) is not necessarily unique, we can use any pair). We let $(i,j) \prec (i',j')$ if either one of the following holds: $i' < j' < i < j$, $i < i' < j < j'$, $i < i' < j' < j < j_0$ and $j_0' \leqslant i' < i < j < j'$. Then, the elements of M_{i_0,k_0} are totally ordered by '$\prec$'.

Let $(i_{\mathrm{top}}, j_{\mathrm{top}})$ be the highest element of M_{i_0,k_0} under '$\prec$'. From the construction of $G(V,E)$ and the total ordering of M_{i_0,k_0}, we can see that there exists a path from v_s to $v_d(i_{\mathrm{top}}, j_{\mathrm{top}}, k)$ or $v_d(i, i_{\mathrm{top}}, j_{\mathrm{top}})$ for some i, k and $d \in \{L,M,R\}$ with the total weight $|M_{i_0,k_0}|$.

Conversely, assume that there exists a path from v_s to $v_d(i,j,k)$ with the total weight $W \geqslant 0$. Then we can obtain a secondary structure M_{i_0,k_0} with score W by using the following rule: $(i,j) \in M_{i_0,k_0}$ if either $v_L(i,j,k)$ $(i \geqslant i_0)$ or $v_R(i',i,j)$ $(i' \geqslant i_0)$ appears in the path. It should be noted that each index i can appear at most once in M_{i_0,k_0} from the structure of the graph.

Since $G(V,E)$ is an acyclic directed graph, the weight of the maximum weight path from v_s to $v_d(i,j,k)$ can be computed by dynamic programming and is equal to $S_d(i,j,k)$. Therefore, it is proven that an optimal pseudoknot is computed.

Next, we analyze the time complexity. For each pair (i_0, k_0), we must compute scores of $O(n^3)$ triplets. Therefore, scores of $O(n^5)$ triplets should be computed in total. However, we do not need to compute $O(n^3)$ scores for each pair (i_0, k_0). Since k_0 does not appear in the recurrences (1)–(3) and $S_d(i,j,k)$ does not depend on $S_{d'}(i',j',k')$ such that $k' > k$, $S_d(i,j,k)$ does not depend on k_0 (although it depends on i_0). Therefore, if scores for (i_0, k_0) are known, we only need to compute $O(n^2)$ scores $S_d(i,j,k_0+1)$

for (i_0, k_0+1). Moreover, we can compute scores in the following order:

```
for $i = i_0$ to $n-2$ do
    for $j = n-1$ downto $i+1$ do
        for $k = j+1$ to $n$ do
```

Therefore, we only need to compute $\mathrm{O}(n^3)$ scores for each i_0. Since each score can be computed in constant time, $\mathrm{O}(n^4)$ time is sufficient in total, which matches the result of Ref. [14].

Finally, we consider the space complexity. In order to reduce the space complexity, we compute all $S_{\mathrm{pseudo}}(i_0, k_0)$ before computing all $S(i,j)$. Although $\mathrm{O}(n^3)$ space is required for computing all $S_{\mathrm{pseudo}}(i_0, k_0)$ for fixed i_0, the same memory space can be used for computing the other $S_{\mathrm{pseudo}}(i'_0, k'_0)$. Since $\mathrm{O}(n^2)$ space is sufficient for storing all values of $S_{\mathrm{pseudo}}(i_0, k_0)$, the total space complexity is $\mathrm{O}(n^3)$. □

4. Approximation algorithm for simple pseudoknots

Since the time complexity shown in Section 3 is too high, it should be reduced. Although some heuristic technique was used in Ref. [14], they did not succeed to reduce the worst-case time complexity. In this section, we show an $\mathrm{O}(n^{4-\delta})$ time algorithm that computes, for most RNA sequences, a secondary structure with simple pseudoknots whose score is at least $1-\varepsilon$ of the optimal, where ε, δ are any constants such that $0 < \varepsilon, \delta < 1$.

First we show that $OPT_1(A)$ is $\Theta(n)$ for most RNA sequences.

Proposition 2. *If* $\mathtt{a}, \mathtt{u}, \mathtt{g}, \mathtt{c}$ *occur uniformly and independently in a random RNA sequence of length* n, $OPT_1(A) \geqslant n/4$ *with probability at least* $\geqslant 1 - 4/\mathrm{e}^{n/32}$.

Proof. Let #a, #u, #g, #c be the numbers of occurrences of a, u, g, c in an RNA sequence, respectively. In Ref. [3], $OPT_0(A) \geqslant \min(\#\mathtt{a}, \#\mathtt{u}) + \min(\#\mathtt{g}, \#\mathtt{c})$ was proven. Since $OPT_1(A) \geqslant OPT_0(A)$ holds from the definitions, $OPT_1(A) \geqslant \min(\#\mathtt{a}, \#\mathtt{u}) + \min(\#\mathtt{g}, \#\mathtt{c})$ holds.

On the other hand, the probability that #a is less than $n/8$ is at most $\mathrm{e}^{-(n/32)}$ from the Chernoff bound [10]. Since the probability that $\min(\#\mathtt{a}, \#\mathtt{u}) + \min(\#\mathtt{g}, \#\mathtt{c}) < n/4$ is bounded above by the probability that $\min\{\#\mathtt{a}, \#\mathtt{u}, \#\mathtt{g}, \#\mathtt{c}\} < n/8$, the proposition holds. □

Next we show an approximation algorithm which outputs a good approximate secondary structure with simple pseudoknots if $OPT_1(A)$ is $\Theta(n)$. It is obtained by modifying the exact algorithm shown in Section 3. Although the modification is simple and based on the technique developed in Ref. [3], it is not straightforward and an additional idea is required.

Recall that, in the exact algorithm in Section 3, we compute simple pseudoknots for all pairs of (i_0, k_0). In the modified algorithm, we compute simple pseudoknots for

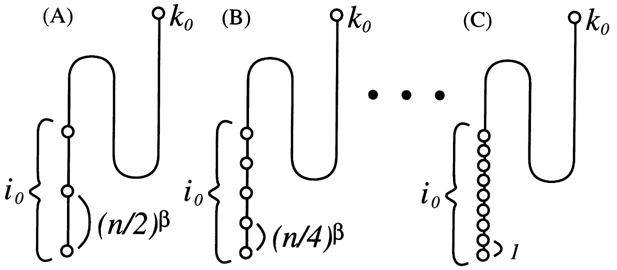

Fig. 5. Scores $S_{\text{pseudo}}(i_0,k_0)$ are not computed for all i_0 but for restricted i_0. (A) $n/2<k_0-i_0\leqslant n$, (B) $n/4<k_0-i_0\leqslant n/2,\ldots$, (C) $k_0-i_0\leqslant n^\alpha$.

restricted pairs of (i_0,k_0). Let α,β be some fixed reals such that $0<\alpha,\beta<1$. Let I_h $(h\geqslant 1)$ be a set of pair of indices defined by

$$I_h=\left\{(i_0,k_0)\,|\,i_0 \bmod \left\lceil\left(\frac{n}{2^h}\right)^\beta\right\rceil=0 \text{ and } \frac{n}{2^h}<k_0-i_0\leqslant\frac{n}{2^{h-1}}\right\}.$$

Let I be a set of pairs of indices defined by

$$I=I_1\cup I_2\cup\cdots\cup I_{H-1}\cup I_H\cup\{(i_0,k_0)\,|\,0<k_0-i_0\leqslant n^\alpha\},$$

where H is the smallest number satisfying $n/2^{H-1}\leqslant n^\alpha$. In the modified algorithm, we only compute pseudoknots for (i_0,k_0) in I (see Fig. 5) and we use the same procedure for $S(i,j)$ as in Section 3. We denote this modified algorithm by $\mathscr{APR}$.

Lemma 3. *$\mathscr{APR}$ works in* $O(n^{4-\beta}+n^{1+3\alpha})$ *time.*

Proof. Since the time complexity for the other parts is $O(n^3)$, we only consider the part of computing simple pseudoknots.

The time for computing simple pseudoknots for all (i_0,k_0) in I_h is

$$O\left(\frac{n}{(n/2^h)^\beta}\left(\frac{n}{2^h}\right)^3\right)=O(n^{4-\beta}2^{(\beta-3)h}).$$

The time for computing simple pseudoknots for (i_0,k_0) not in any I_h is $O(n\times(n^\alpha)^3)=O(n^{1+3\alpha})$. Since

$$n^{4-\beta}\sum_{h=1}^{\infty}2^{(\beta-3)}h\leqslant n^{4-\beta},$$

the total time for computing simple pseudoknots is $O(n^{4-\beta}+n^{1+3\alpha})$. □

Next we define the *error* of an approximate secondary structure to the optimal one to be the difference between their scores, where the optimal score is always greater than or equal to the approximate score.

Lemma 4. *The error of a secondary structure computed by $\mathscr{APR}$ is* $O(n^{1+\alpha\beta-\alpha})$.

Proof. Note that the error due to a pseudoknot with endpoints $(i_0, k_0) \in I_h$ is at most $O((n/2^h)^{\beta})$ because the distance between adjacent i_0's is $O((n/2^h)^{\beta})$.

Therefore, it is easy to verify that the worst case error is bounded above by the error of a case in which $\Theta(n/n^{\alpha})$ pseudoknots of length $\Theta(n^{\alpha})$ occur. In that case, the error due to each pseudoknot is $O(n^{\alpha\beta})$. Therefore, the total error is bounded above by $O((n/n^{\alpha})n^{\alpha\beta}) = O(n^{1+\alpha\beta-\alpha})$. □

From the above lemma, the score of an approximate structure is at least $1-\varepsilon$ of $OPT_1(A)$ for sufficiently large n (precisely, $n > N$ where N is a constant depending on δ and ε) if $OPT_1(A)$ is $\Theta(n)$ and $1+\alpha\beta-\alpha < 1$. Therefore, we have:

Theorem 5. *If* a, u, g, c *occur uniformly and independently in a random RNA sequence of length n, an RNA secondary structure with simple pseudoknots, in which the number of base pairs is at least* $1-\varepsilon$ *of the optimal, can be computed in* $O(n^{4-\delta})$ *time with probability at least* $1 - 4/e^{n/32}$, *where* ε, δ *are any fixed constants such that* $0 < \varepsilon, \delta < 1$.

Note that, unlike usual polynomial time approximation scheme (PTAS), the time complexity does not depend on ε although threshold N depends on δ and ε. Note also that $\mathscr{APR}$ always outputs a secondary structure with simple pseudoknots whose score is at least $1-\varepsilon$ of the optimal for sufficiently large n if $\min(\#\mathtt{a}, \#\mathtt{u}) + \min(\#\mathtt{g}, \#\mathtt{c}) = \Theta(n)$.

5. Extensions

In this section, we show that the proposed dynamic programming algorithms can be extended in various ways.

5.1. Energy function depending on adjacent base pairs

Although we have considered the problem of maximizing the number of base pairs so far, energy functions depending on adjacent base pairs are widely used (i.e., an energy function is a function from $\Sigma \times \Sigma \times \Sigma \times \Sigma$ to the set of reals) [7,13,14]. Moreover, free energy for such pairs usually takes negative values and the problem is defined as a minimization problem. In this subsection, we consider an energy function $\mu(a_{i-1}a_i, a_j a_{j+1})$ depending on adjacent base pairs (a_{i-1}, a_{j+1}) and (a_i, a_j), instead of an energy function $v(a_i, a_j)$ depending on a base pair (a_i, a_j).

The algorithm shown in Section 3 can be modified for energy functions depending on adjacent base pairs. Since the modification for a secondary structure without pseudonots is already known [15,18], we only show the modification for computing $S_{\mathrm{pseudo}}(i_0, k_0)$.

For that purpose, we use tables $S_{LL}(i,j,k)$ and $S_{RR}(i,j,k)$ in addition to $S_L(i,j,k)$, $S_M(i,j,k)$ and $S_R(i,j,k)$. $S_{LL}(i,j,k)$ and $S_{RR}(i,j,k)$ correspond to the occurrences of adjacent base pairs $((a_{i-1}, a_{j+1}), (a_i, a_j))$ and $((a_{j+1}, a_{k-1}), (a_j, a_k))$ respectively. Note also that ‘max’ must be replaced by ‘min’ in this case because the problem is defined as a minimization problem. The modified dynamic programming procedure is shown below:

$$S_{LL}(i,j,k) = \mu(a_{i-1}a_i, a_ja_{j+1}) + \min\left\{\begin{array}{l} S_L(i-1,j+1,k), \\ S_{LL}(i-1,j+1,k) \end{array}\right\},$$

$$S_L(i,j,k) = \min\left\{\begin{array}{ll} S_M(i-1,j+1,k), & S_R(i-1,j+1,k), \\ S_{RR}(i-1,j+1,k) & \end{array}\right\},$$

$$S_{RR}(i,j,k) = \mu(a_ja_{j+1}, a_{k-1}a_k) + \min\left\{\begin{array}{l} S_R(i,j+1,k-1), \\ S_{RR}(i,j+1,k-1) \end{array}\right\},$$

$$S_R(i,j,k) = \min\left\{\begin{array}{ll} S_L(i,j+1,k-1), & S_M(i,j+1,k-1), \\ S_{LL}(i,j+1,k-1) & \end{array}\right\},$$

$$S_M(i,j,k) = \min\left\{\begin{array}{lll} S_M(i-1,j,k), & S_M(i,j+1,k), & S_M(i,j,k-1), \\ S_L(i-1,j,k), & S_L(i,j+1,k), & S_R(i,j+1,k), \\ S_R(i,j,k-1), & S_{LL}(i-1,j,k), & S_{LL}(i,j+1,k), \\ S_{RR}(i,j+1,k), & S_{RR}(i,j,k-1) & \end{array}\right\}.$$

Since the time complexity increases by a constant factor owing to this modification, we obtain an $O(n^4)$ time exact algorithm for computing a secondary structure with simple pseudoknots under an energy function depending on adjacent base pairs.

Since the number of possible combinations of adjacent base pairs is finite (4^4), we can assume that free energy for adjacent base pairs $\mu(a_{i-1}a_i, a_ja_{j+1})$ is bounded by a constant, i.e., $0 \geqslant \mu(a_{i-1}a_i, a_ja_{j+1}) \geqslant -E$ holds for some constant E. In such a case, the technique and the analysis in Section 4 can also be applied in a straightforward way. Therefore, we have the following theorem.

Theorem 6. *Under an energy function depending on adjacent base pairs, an optimal RNA secondary structure with simple pseudoknots can be computed in* $O(n^4)$ *time, and an approximate RNA secondary structure with simple pseudoknots whose free energy is at most* $1-\varepsilon$ *of the minimum can be computed in* $O(n^\delta)$ *time for most random RNA sequences, where* ε, δ *are any constants such that* $0 < \varepsilon, \delta < 1$.

5.2. Destabilizing energy

Although we did not consider free energy for loop regions, loop regions are also important determinants of RNA stability. For loop regions in a secondary structure without pseudoknots, destabilizing energy functions depending on the length of the loop are used in many references [1,13,15,17] although sequence dependence is unclear owing to the lack of data [13]. Most destabilizing energy functions are determined from experimental data on short loops and extrapolation [1,13]. Since much fewer properties are known about destabilizing energy for pseudoknots [1], we use a simple energy function such that the destabilizing energy of each loop is determined from the length of the loop. In the following, $\xi(x)$ denotes the destabilizing energy for a loop of length x.

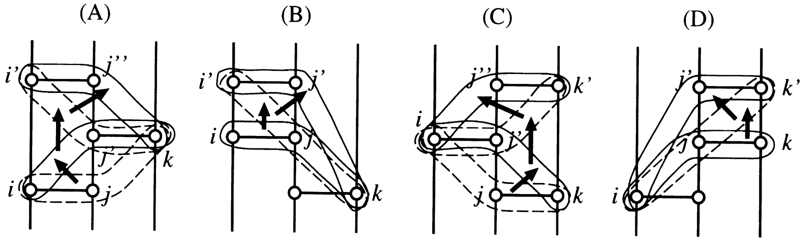

Fig. 6. Four transition patterns for computing an optimal pseudoknot structure with destabilizing energy.

Since an $O(n^4)$ time algorithm for computing an optimal secondary structure without pseudoknots under energy functions including destabilizing energies is known [15], we only consider the computation of an optimal pseudoknot.

In order to take destabilizing energy into account, it suffices to consider the four transition patterns shown in Fig. 6, where the exceptional case in which either $\{(i,j) \in M_{i_0,k_0} \mid i < j'_0\} = \emptyset$ or $\{(i,j) \in M_{i_0,k_0} \mid j'_0 \leqslant i\} = \emptyset$ holds can be covered by the algorithm without pseudoknots. Let $b(i)$ denote the base pair such that either one of the endpoints is a_i. Instead of $S_d(i,j,k)$ in Section 3, we use triplets $S_{xyz}(i,j,k)$ $(i<j<k)$ where $x,y,z \in \{L,M,H\}$. $S_{xyz}(i,j,k)$ means that each of a_i, a_j and a_k must be an endpoint of some base pair and that the ordering of $b(i)$, $b(j)$ and $b(k)$ is the same as that of x, y and z, where we use the ordering defined in Section 3 for $b(i)$, $b(j)$ and $b(k)$ and we let $L \prec M \prec H$. For example, $S_{LMH}(i,j,k)$ and $S_{HLL}(i,j,k)$ mean $b(i) \prec b(j) \prec b(k)$ and $b(j) = b(k) \prec b(i)$, respectively. We do not use all types of $S_{xyz}(i,j,k)$ but use the following types: $S_{HHL}(i,j,k)$, $S_{HLL}(i,j,k)$, $S_{HML}(i,j,k)$, $S_{LLH}(i,j,k)$, $S_{LHH}(i,j,k)$ and $S_{LMH}(i,j,k)$.

As in Section 3, we consider a directed acyclic graph whose vertex set consists of $v_{xyz}(i,j,k)$, v_s and v_t, where v_t is the end point and the weight of the minimum weight path from v_s to $v_{xyz}(i,j,k)$ is equal to $S_{xyz}(i,j,k)$. Then, transition patterns (A)–(D) in Fig. 6 are represented by the following directed paths:

$$\begin{aligned}
&(\text{A})\ v_{LLH}(i,j,k) \rightarrow v_{LHH}(i,j',k) \rightarrow v_{HLL}(i',j',k) \rightarrow v_{HHL}(i',j'',k),\\
&(\text{B})\ v_{HHL}(i,j,k) \rightarrow v_{HML}(i',j,k) \rightarrow v_{HHL}(i',j',k),\\
&(\text{C})\ v_{HLL}(i,j,k) \rightarrow v_{HHL}(i,j',k) \rightarrow v_{LLH}(i,j',k') \rightarrow v_{LHH}(i,j'',k'),\\
&(\text{D})\ v_{LHH}(i,j,k) \rightarrow v_{LMH}(i,j,k') \rightarrow v_{LHH}(i,j',k').
\end{aligned}$$

Weights of edges are defined by

$$\begin{aligned}
&w(v_{LLH}(i,j,k), v_{LHH}(i,j',k)) = v(a_{j'}, a_k) + \xi(j - j' - 1),\\
&w(v_{LHH}(i,j,k), v_{HLL}(i',j,k)) = \xi(i' - i - 1),\\
&w(v_{HLL}(i,j,k), v_{HHL}(i,j',k)) = v(a_i, a_{j'}) + \xi(j - j' - 1),\\
&w(v_{HHL}(i,j,k), v_{HML}(i',j,k)) = \xi(i' - i - 1),\\
&w(v_{HML}(i,j,k), v_{HHL}(i,j',k)) = v(a_i, a_{j'}) + \xi(j - j' - 1),\\
&w(v_{HHL}(i,j,k), v_{LLH}(i,j,k')) = \xi(k' - k - 1),
\end{aligned}$$

$$w(v_{LHH}(i,j,k), v_{LMH}(i,j,k')) = \xi(k'-k-1),$$
$$w(v_{LMH}(i,j,k), v_{LHH}(i,j',k)) = v(a_{j'}, a_k) + \xi(j-j'-1).$$

It should be noted that the value of either one of i, j and k changes through each edge. Weights of edges from v_s and weights of edges into v_t are defined by

$$w(v_s, v_{LLH}(i,j,k)) = v(a_i, a_j) + \xi(i-i_0) + \xi(k-j-1),$$
$$w(v_s, v_{HLL}(i,j,k)) = v(a_j, a_k) + \xi(i-i_0) + \xi(k-j-1),$$
$$w(v_{HHL}(i,j,k), v_t) = \xi(j-i-1) + \xi(k_0-k),$$
$$w(v_{LHH}(i,j,k), v_t) = \xi(j-i-1) + \xi(k_0-k).$$

Then, it is seen that the free energy of an optimal pseudoknot is equal to the weight of the minimum weight path from v_s to v_t.

As in Section 3, the minimum weight path can be computed by a dynamic programming procedure. Since there are $O(n^4)$ edges in this case, $O(n^4)$ time is required in order to compute an optimal pseudoknot for fixed (i_0, k_0). As in Section 3, scores computed for (i_0, k_0) can also be used for computing scores for (i_0, k_0+1). Note that special care is required in this case since $\xi(k_0-k)$ must be taken into account. For that purpose, we compute the following values for each k:

$$S_{i_0}(k) = \min_{i,j}\{S_{HHL}(i,j,k) + \xi(j-i-1), S_{LHH}(i,j,k) + \xi(j-i-1)\},$$

where these values do not depend on k_0. And, we let $S_{\text{pseudo}}(i_0, k_0) = \min_{k \leqslant k_0}\{S_{i_0}(k) + \xi(k_0-k)\}$.

Since $O(n^4)$ time is required for each i_0, $O(n^5)$ time is required for computing all values of $S_{\text{pseudo}}(i_0, k_0)$. Since $O(n^4)$ time is sufficient for the other parts, the total time complexity is $O(n^5)$.

We can modify this algorithm so that energies depending on adjacent base pairs are taken into account without increasing the order of the time complexity, by combining the method mentioned in Section 5.1.

Theorem 7. *Under an energy function consisting of stacking energy depending on adjacent base pairs and destabilizing energy depending on the length of the loop, an optimal RNA secondary structure with simple pseudoknots can be computed in* $O(n^5)$ *time.*

5.3. Recursive pseudoknots

Although we have considered simple pseudoknots so far, recursive pseudoknot structures may appear in real RNAs. We can obtain an algorithm for recursive pseudoknots by modifying the algorithm in Section 5.2 (see Fig. 7).

We use the same directed graph as in Section 5.2. But, weights of edges are changed by replacing $\xi(i'-i-1)$, $\xi(j-j'-1)$, $\xi(k'-k-1)$, $\xi(i-i_0)$, $\xi(k-j-1)$, $\xi(j-i-1)$ and $\xi(k_0-k)$ with $S(i+1, i'-1)$, $S(j'+1, j-1)$, $S(k+1, k'-1)$, $S(i-1, i_0)$, $S(j+1, k-1)$, $S(i+1, j-1)$ and $S(k+1, k_0)$, where $S(x,y)$ was defined in Section 3.

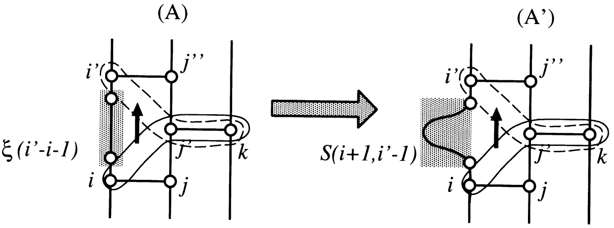

Fig. 7. Modification of weights for recursive pseudoknots. $\xi(i' - i - 1)$ in Fig. 6(A) is replaced by $S(i + 1, i' - 1)$.

For example, $w(v_{LHH}(i,j,k), v_{HLL}(i',j,k)) = \xi(i' - i - 1)$ in Section 5.2 is replaced by $w(v_{LHH}(i,j,k), v_{HLL}(i',j,k)) = S(i + 1, i' - 1)$.

We can arrange the dynamic programming procedure so that the required values of $S(x, y)$ are already determined before determining $S_{xyz}(i,j,k)$. As in Section 5.2, we can show that the total time complexity is $O(n^5)$, and the algorithm can be modified so that energies depending on adjacent base pairs are taken into account.

Theorem 8. *Under an energy function depending on adjacent base pairs, an optimal RNA secondary structure with recursive pseudoknots can be computed in* $O(n^5)$ *time.*

6. Hardness result for generalized pseudoknots

In Section 5, we showed that dynamic programming algorithms can cover most types of pseudoknots (we do not know what extent we should cover because no established definition of a pseudoknot is known). However, the forms of secondary structures cannot be extended to the entire class of planar graphs. In such a case, we can prove an NP-hardness result. In the following, an RNA secondary structure with *generalized pseudoknots* means a structure having any planar graph structure under the condition that each residue can be connected with at most one residue except adjacent residues. Note that adjacent residues are assumed to be connected by an edge in a planar graph.

Theorem 9. *RNA secondary structure prediction with generalized pseudoknots is* NP-*hard, where we assume that an arbitrary energy function depending on adjacent base pairs can be used.*

Proof. We use a reduction from longest common subsequence (LCS), which is known to be NP-complete [6,8].

The decision problem version of LCS is, given a set of strings $L = \{s_1, s_2, \ldots, s_m\}$ ($|s_1| = |s_2| = \cdots = |s_m| = n$) over an alphabet Σ and an integer k, to decide whether or not there exists a string s_c of length k that is a (not necessarily consecutive) subsequence of each s_i. In this proof, we consider a case of $\Sigma = \{\mathrm{a}, \mathrm{u}\}$, for which LCS remains NP-complete [8].

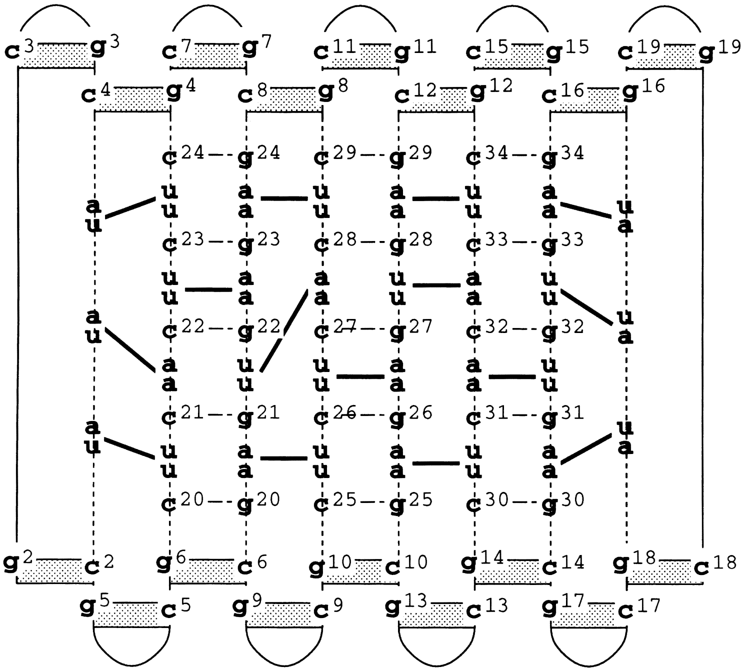
c3 g3 c7 g7 c11 g11 c15 g15 c19 g19
c4 g4 c8 g8 c12 g12 c16 g16
c24 g24 c29 g29 c34 g34
c23 g23 c28 g28 c33 g33
c22 g22 c27 g27 c32 g32
c21 g21 c26 g26 c31 g31
c20 g20 c25 g25 c30 g30
g2 c2 g6 c6 g10 c10 g14 c14 g18 c18
g5 c5 g9 c9 g13 c13 g17 c17

Fig. 8. An optimal secondary structure with generalized pseudoknots corresponding to LCS instance: {uauu, uuau, uaau}, $k = 3$. This structure is not qualified as a secondary structure with simple pseudoknots.

From an instance of LCS, we construct a large RNA sequence A in the following way (see Fig. 8). Let $s_{i,j}$ denote the jth letter of string s_i. Let x^i denote $\overbrace{xx\ldots x}^{i}$. For $i = 1,\ldots,m$, we construct sequences D_i and E_i by

$$D_i = \mathrm{c}^{I_i} s_{i,1}\, s_{i,1}\, \mathrm{c}^{I_i+1} s_{i,2}\, s_{i,2} \mathrm{c}^{I_i+2} \ldots \mathrm{c}^{I_i+n-1} s_{i,n}\, s_{i,n} \mathrm{c}^{I_i+n},$$

$$E_i = \mathrm{g}^{I_i+n}\overline{s_{i,n}}\,\overline{s_{i,n}}\mathrm{g}^{I_i+n-1} \ldots \mathrm{g}^{I_i+2}\overline{s_{i,2}}\,\overline{s_{i,2}}\mathrm{g}^{I_i+1}\overline{s_{i,1}}\,\overline{s_{i,1}}\mathrm{g}^{I_i},$$

respectively, where $\bar{x}$ denotes the complementary residue of x and $I_i = 4m + 8 + (i-1)(n+1)$. For $i = 1,\ldots,m$, we construct B_i by

$$B_i = \mathrm{c}^{4i+1}\mathrm{g}^{4i+2} D_i \mathrm{g}^{4i}\mathrm{c}^{4i+3}\mathrm{g}^{4i+3}\mathrm{c}^{4i+4} E_i \mathrm{c}^{4i+2}\mathrm{g}^{4i+5}.$$

We construct B_0 and B_{m+1} by

$$B_0 = \mathrm{g}^2\mathrm{c}^3\mathrm{g}^3\mathrm{c}^4(\mathrm{au})^k\mathrm{c}^2\mathrm{g}^5,$$

$$B_{m+1} = \mathrm{c}^{4m+5}\mathrm{g}^{4m+6}(\mathrm{au})^k\mathrm{g}^{4m+4}\mathrm{c}^{4m+7}\mathrm{g}^{4m+7}\mathrm{c}^{4m+6},$$

respectively. Finally, A is constructed by $A = B_0B_1B_2\ldots B_{m+1}$.

The energy function is defined by $\mu(\mathrm{aa},\mathrm{uu}) = \mu(\mathrm{uu},\mathrm{aa}) = \mu(\mathrm{cc},\mathrm{gg}) = \mu(\mathrm{gg},\mathrm{cc}) = \mu(\mathrm{au},\mathrm{aa}) = \mu(\mathrm{au},\mathrm{uu}) = \mu(\mathrm{ua},\mathrm{aa}) = \mu(\mathrm{ua},\mathrm{uu}) = \mu(\mathrm{aa},\mathrm{au}) = \mu(\mathrm{uu},\mathrm{au}) = \mu(\mathrm{aa},\mathrm{ua}) = \mu(\mathrm{uu},\mathrm{ua}) = -1.0$, otherwise $\mu(xy,zw) = 0.0$. As in Section 5.1, $\mu(xy,zw)$ denotes the energy for consecutive base pairs (x,w) and (y,z). Note that we assume an arbitrary energy function here and thus $(\mathtt{a},\mathtt{u})$, $(\mathtt{u},\mathtt{u})$ and $(\mathtt{a},\mathtt{a})$ can be base pairs.

We show that the following property holds: there exists a common subsequence s_c of length k if and only if there exists a secondary structure of A with score (energy): $-nm-k-\sum_{i=2}^{nm+5m+7}(i-1)$.

First, we show that if there exists a common subsequence of length k, then we can construct a secondary structure with score $-nm-k-\sum_{i=2}^{nm+5m+7}(i-1)$.

Let s_c be a common subsequence of length k. For each s_i, let $s_{i,f_i(1)},s_{i,f_i(2)},\ldots,s_{i,f_i(k)}$ be a subsequence of s_i such that $s_{i,f_i(1)}s_{i,f_i(2)}\ldots\ s_{i,f_i(k)}=s_c$. We make a secondary structure as follows (see Fig. 8):

(i) For each of $h=2,\ldots,nm+5m+7$, $\mathtt{c}^h$ is paired with $\mathtt{g}^h$ (note that $\mathtt{c}^h$ (resp. $\mathtt{g}^h$) appears exactly once in A for each h).

(ii) For all $j=1,\ldots,k$, $s_{1,f_1(j)}s_{1,f_1(j)}$ in D_1 is paired with $\mathtt{au}$ in B_0, and $\overline{s_{n,f_n(j)}}\,\overline{s_{n,f_n(j)}}$ in E_n is paired with $\mathtt{au}$ in B_{n+1}.

(iii) For all $i=1,\ldots,m$, each $s_{i,j}s_{i,j}$ in D_i such that $j\notin\{f_i(1),\ldots,f_i(k)\}$ is paired with $\overline{s_{i,j}}\,\overline{s_{i,j}}$ in E_i.

(iv) For all $i=1,\ldots,m-1$ and for all $j=1,\ldots,k$, $\overline{s_{i,f_i(j)}}\,\overline{s_{i,f_i(j)}}$ in E_i is paired with $s_{i+1,f_{i+1}(j)}s_{i+1,f_{i+1}(j)}$ in D_{i+1}.

By these pairings, every residue in A is paired with another residue in A. Since the score due to (i) is $-\sum_{i=2}^{nm+5m+7}(i-1)$, the score due to (ii) is $-2k$, the score due to (iii) is $-(n-k)m$ and the score due to (iv) is $-k(m-1)$, the total score of the constructed secondary structure is $-nm-k-\sum_{i=2}^{nm+5m+7}(i-1)$.

It can also be seen that the secondary structure constructed as above is an optimal secondary structure because every residue is paired with another residue in the structure and adjacent residues a_ia_{i+1} cannot contribute to the score unless $a_i=a_{i+1}$, $a_ia_{i+1}=\mathtt{au}$ or $a_ia_{i+1}=\mathtt{ua}$.

Next we show that if there exists an RNA secondary structure with score $-nm-k-\sum_{i=2}^{nm+5m+7}(i-1)$, then we can construct a common subsequence s_c of length k.

Let M be a secondary structure with score $-nm-k-\sum_{i=2}^{nm+5m+7}(i-1)$. Since M is an optimal secondary structure, each $\mathtt{c}^h$ must be paired with $\mathtt{g}^h$. Since the graph shown in Fig. 9 is 3-connected, this graph has a unique planar embedding except the mirror image and the choice of the outer face [11], where this graph corresponds to a partial structure determined by pairings due to $\mathtt{c}^h$ and $\mathtt{g}^h$ in $\mathtt{c}^{4i}\mathtt{g}^{4i+1}B_i\mathtt{c}^{4i+5}\mathtt{g}^{4i+4}$. From this embedding and the fact that $\mu(\mathtt{aa},xy)=\mu(xy,\mathtt{aa})\neq 0$ (resp. $\mu(\mathtt{uu},xy)=\mu(xy,\mathtt{uu})\neq 0$) only when $xy=\mathtt{uu}$, $xy=\mathtt{au}$ or $xy=\mathtt{ua}$ (resp. $xy=\mathtt{aa}$, $xy=\mathtt{au}$ or $xy=\mathtt{ua}$), we can see the following:

- $s_{1,j}s_{1,j}$ in D_1 must be paired with either $\mathtt{au}$ in B_0 or $\overline{s_{1,j}}\,\overline{s_{1,j}}$ in E_1.
- $\overline{s_{m,j}}\,\overline{s_{m,j}}$ in E_m must be paired with either $\mathtt{au}$ in B_{m+1} or $s_{m,j}s_{m,j}$ in D_m.
- For $i=1,\ldots,m-1$, $\overline{s_{i,j}}\,\overline{s_{i,j}}$ in E_i must be paired with either $s_{i,j}s_{i,j}$ in D_i or $s_{i+1,j'}s_{i+1,j'}$ in D_{i+1} such that $s_{i+1,j'}=s_{i,j}$.
- If $\overline{s_{i,j}}\,\overline{s_{i,j}}$ and $\overline{s_{i,h}}\,\overline{s_{i,h}}$ in E_i $(j<h)$ are paired with $s_{i+1,j'}s_{i+1,j'}$ and $s_{i+1,h'}s_{i+1,h'}$ in D_{i+1}, respectively, then $j'<h'$ must hold.

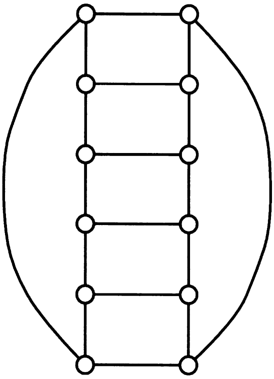

Fig. 9. This graph is 3-connected and thus a planar embedding is essentially unique.

Since every residue in an optimal secondary structure must be paired with another residue, each au in B_0 must be paired with $s_{1,j}s_{1,j}$ in D_1 for some j. Let $s_{1,f_1(j)}s_{1,f_1(j)}$ $(f_1(1) < f_1(2) < \cdots < f_1(k))$ be residues in D_1 paired with au in B_0. Then, $s_c = s_{1,f_1(1)}s_{1,f_1(2)}\ldots s_{1,f_1(k)}$ is a subsequence of s_1. Moreover, from the above constraints on pairings, s_c must be a common subsequence of $s_1, s_2, \ldots, s_m$.

Since the reduction can be done in polynomial time, the theorem holds. □

7. Concluding remarks

In this paper, we have shown that dynamic programming is still useful for the RNA secondary structure prediction with pseudoknots. As mentioned in Section 1, the most important contribution of this paper is that it corrects the previous misunderstanding that pseudoknots cannot be handled by a simple dynamic programming-based approach.

As shown in Section 5, dynamic programming algorithms can cover most types of pseudoknots. However, the time complexity increases to $O(n^5)$ or more if complex pseudoknots are handled. Therefore, improvements on the time complexities should be done. Since there is a close relationship between the prediction problem with pseudoknots and the parsing problem of tree-adjoining grammars [14], the technique developed for the recognition of tree-adjoining grammars [12] might be useful for such improvements. From a practical viewpoint, the following approach seems useful: first we enumerate candidates of stacked regions, and then we combine candidates so that combined regions do not violate the constraints. Although similar approaches have been already employed [1,2], combining candidates can also be done using a dynamic programming procedure.

Another important problem is that no established definition of a pseudoknot is known. Although the proposed dynamic programming algorithms can cover a wide class of pseudoknots, we do not know what class we should cover. Thus, it would be helpful if a formal definition of a pseudoknot is discussed and given by biologists.

The other important problem is that no established energy function is known for pseudoknots, especially for loop regions [14]. For making accurate predictions, development of such energy function is very important and should be studied.

References

[1] J.P. Abrahams, M. Berg, E. Batenburg, C. Pleij, Prediction of RNA secondary structure, including pseudoknotting by computer simulation, Nucleic Acids Res. 18 (1990) 3035–3044.

[2] Y. Akiyama, M. Kanehisa, NeuroFold: an RNA secondary structure prediction system using a Hopfield neural network, Proceedings of the Genome Informatics Workshop III, Universal Academy Press, Tokyo, 1992, pp. 199–202 (in Japanese).

[3] T. Akutsu, Approximation and exact algorithms for RNA secondary structure prediction and recognition of stochastic context-free languages, J. Combin. Optim. 3 (1999) 321–336.

[4] M. Brown, C. Wilson, RNA pseudoknots modeling using intersections of stochastic context free grammars with applications to database search, in: L. Hunter, T.E. Klein (Eds.), Pacific Symposium on Biocomputing'96, World Scientific, Singapore, 1996, pp. 109–125.

[5] Z. Galil, K. Park, Dynamic programming with convexity, concavity and sparsity, Theoret. Comput. Sci. 92 (1992) 49–76.

[6] M.R. Garey, D.S. Johnson, Computers and Intractability. A Guide to the Theory of NP-completeness, Freeman, New York, 1979.

[7] M. Gribskov, J. Devereux (Eds.), Sequence Analysis Primer, Sptockton Press, New York, 1991.

[8] R. Maier, The complexity of some problems on subsequences and supersequences, J. ACM 25 (1978) 322–336.

[9] H. Mizuno, M. Sundaralingam, Stacking of Crick Wobble pair and Watson–Crick pair: stability rules of G-U pairs at ends of helical stems in tRNAs and the relation to codon-anticodon Wobble interaction, Nucleic Acids Res. 5 (1978) 4451–4461.

[10] R. Motowani, P. Raghavan, Randomized Algorithms, Cambridge University Press, New York, 1995.

[11] T. Nishizeki, N. Chiba, Planar Graphs: Theory and Algorithms, Elsevier, Amsterdam, 1988.

[12] S. Rajasekaran, S. Yooseph, TAL recognition in $O(M(n^2))$ time, J. Comput. System Sci. 56 (1998) 83–89.

[13] D.H. Turner, N. Sugimoto, S.M. Freier, RNA structure prediction, Ann. Rev. Biophys, Biophys. Chem. 17 (1988) 167–192.

[14] Y. Uemura, A. Hasegawa, S. Kobayashi, T. Yokomori, Grammatically modeling and predicting RNA secondary structures, in: M. Hagiya et al. (Eds.), Proceedings of the Genome Informatics Workshop 1995, Universal Academy Press, Tokyo, 1995, pp. 67–76.

[15] M.S. Waterman, Introduction to Computational Biology, Chapman & Hall, London, 1995.

[16] M.S. Waterman, T.F. Smith, RNA secondary structure: a complete mathematical analysis, Math. Biosci. 41 (1978) 257–266.

[17] M. Zuker, D. Sankoff, RNA secondary structures and their prediction, Bull. Math. Biol. 46 (1984) 591–621.

[18] M. Zuker, P. Stiegler, Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information, Nucleic Acids Res. 9 (1981) 133–148.