An RNA Pseudoknot Folding Algorithm Based on Heuristic Searching

C. C. Lin, G. S. Huang and R. C. T. Lee
Department of Computer Science and Information Engineering,
National Chi Nan University, Puli, Nantou Hsien, Taiwan, 54561
{s2321905, shieng, rctlee}@ncnu.edu.tw

Abstract

The general problem for the prediction of RNA secondary structure with pseudoknots is NP-hard. For a restricted class of pseudoknots, it was traditionally solved by the dynamic programming approach. However, these algorithms are computationally very expensive, when the input sequence or the contained pseudoknot is too long. We present a heuristic algorithm for RNA secondary structures with pseudoknots prediction. We shall point out that the problem can be considered as a tree searching problems and can be solved by our algorithm effectively. It is very efficient and can be used as a practical tool especially for pseudoknots.

1 Introduction

RNA plays a key role in the biological processes, such as carriers of genetic information, catalysts and mediators. Many of them contain pseudoknots. However, the RNA secondary structure with pseudoknots is NP-hard [1, 2] and still an open problem.

An RNA sequence is a sequence \( R = r_1r_2 \cdots r_n \) where \( r_i \in \{A,U,G,C\} \). In RNA, A and U, G and C and G and U can form base pairs. According to the thermodynamic hypothesis [3], the secondary structure of an RNA sequence is the one with the minimum free energy. In nature, only stable structures can exist and a stable structure must be the one with minimum free energy. In a secondary structure of an RNA sequence, base pairs can increase the structural stability and unpaired bases can decrease the structural stability. We say that an RNA structure is optimum when the number of base pairs is maximized.

The RNA secondary structure prediction problem is informally described as follows: We are given an RNA sequence and we are asked to find the secondary structure such that the number of base pairs is maximized.

There are five kinds of possible secondary structures as shown in Fig. 1. Let \( M \) be a set of base pairs of \( R \). Figure 1(a) is an RNA secondary structure without pseudoknots, which has no distinct pairs \((i,j), (k,h)\) in \( M \) satisfy \( i < j < k \). Fig. 1(b) is the general case for an RNA secondary structure with pseudoknots. An RNA secondary structure with simple pseudoknots is as shown in Fig(c). An RNA secondary structure with H-type pseudoknots contains two stems \((S_1 \text{ and } S_2)\) and two loops \((u \text{ and } w)\) as shown in Fig. 1(d). A simple recursive pseudoknots has a H-type pseudoknots and allows the unpaired strands \(u, v\) and \(w\) to have arbitrary structures, including pseudoknots as shown in Fig. 1(e). The pseudoknots in Fig. 1(b) have recursive structures where any loop region can be folded by arbitrary structures including pseudoknots.

There are many papers discussing the prediction of secondary structures of RNA sequences...
The general problem for the prediction of RNA secondary structures with recursive pseudoknots has been proved to be NP-hard [1, 2] when an arbitrary energy function depending on adjacent base pairs is used.

Figure 1. (a) Non-pseudoknot, (b) Pseudoknots, (c) Simple Pseudoknots, (d) H-type pseudoknots, and (e) Simple Recursive pseudoknots

Nussinov et al. [4] first proposed a dynamic programming solution to predict RNA secondary structures without pseudoknots for maximizing the number of base pairs. Such an approach is also followed by Zuker et al., Waterman et al., Turner et al., Jyan et al. and Lyngsø et al. [2, 5, 6, 7, 8].

Rivas and Eddy [9] proposed a dynamic program to predict the secondary structure of an RNA sequence with pseudoknots. The algorithm has the worst case time and space complexities of $O(n^8)$ and $O(n^4)$ respectively. Akutsu [1] proposed a dynamic programming procedure to predict RNA secondary structures with pseudoknots. The algorithm has the worst case time and space complexities $O(n^4)$ and $O(n^3)$ respectively. Redder et al. [10] modified the method of Rivas and Eddy [9] and used the newest Turner energy rules to predict RNA secondary structures with simple recursive pseudoknots. Their algorithm is called pknotsRG and requires $O(n^4)$ time and $O(n^2)$ space.

Apart from the previously discussed approaches based on dynamic programming, heuristic approaches can be more efficient, but cannot guarantee finding an optimal solution with the minimum free energy. Ruan et al. [11] proposed a heuristic algorithm to predict RNA secondary structures with pseudoknots, which extends the loop matching algorithm (LM) [4] to consider pseudoknots. It method iterates using LM to choose a highest promising region as the predicted structure, called the iterated loop matching (ILM) algorithm. Huang et al. proposed a heuristic approach for detecting RNA H-type pseudoknots, called HPknotter, based on structural matching and dynamic programming kernels [13]. Sperschneider et al. [12] uses a hybrid of sequence matching and free energy minimization approach to predict RNA secondary structures with pseudoknots. This method is called KnotSeeker.

In this paper, we proposed an algorithm based on the idea of A* algorithm [14, 15] for predicting optimal RNA secondary structure with pseudoknots. It is very efficient and can be used as a practical tool especially for long sequences and long pseudoknots.

2 Fragments of an RNA Sequence

An RNA secondary structure consists of some fragments which are defined now. A fragment $f$ is a longest sequence of the form
equals 0, it denotes that \( r_i \) cannot be paired with \( r_j \). After all pairs are found, we scan all diagonals in the forward slash direction from lower left to upper right to find all fragments.

In the following, we give the algorithm to find all fragments of an RNA sequence.

**Algorithm 1** To find all fragments of an RNA sequence

Input: An RNA sequence \( R = r_1 r_2 \cdots r_n \) where \( r_p \in \{A, U, G, C\} \) for \( 1 \leq p \leq n \).

Output: All fragments of the RNA sequence

**Step 1:** /* Initialize \( \mathcal{W} \) and the index \( k \) of fragments */
\[
\mathcal{W} = \{(A, U), (U, A), (G, C), (C, G), (G, U), (U, G)\};
\]
\( k = 1 \);

**Step 2:** /*Find all fragments */
\( F = \phi \);
for \( p = 1 \) to \( n-4 \) do
\( a = 0 \);
while \((p-a \geq 1 \land q+a \leq n)\) do
if \((r_{p-a}, r_{q+a}) \in \mathcal{W}\) then
\( F = F \cup \{(r_{p-a}, r_{q+a})\}; \)
else
if \( F_k \neq \phi \) then
\( F = F \cup \{F_k\}; \)
\( k = k + 1 ; \)
\( F_k = \phi ; \)
endwhile
if \( F_k \neq \phi \) then
\( F = F \cup \{F_k\}; \)
\( k = k + 1 ; \)
\( F_k = \phi ; \)
endfor
endfor

**Step 3:** Output Sorted \( F \);

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>A</td>
<td>U</td>
<td>C</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>G</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
</tr>
</tbody>
</table>

**Figure 2.** Illustrating the order of fragments
3 The Conditions on Fragments

Consider the above fragments in Fig. 2. The secondary structure of an RNA sequence would be a subset of fragments identified by Algorithm 1. If we select a set of fragments, it may contain pseudoknots. For example, in Fig. 2, if we selected \( f_2 \) and \( f_3 \), they form a pseudoknot as shown in Fig. 3. Therefore, we shall consider two kinds of conditions on fragments for our heuristic searching: without pseudoknots and with pseudoknots.

3-1 The Conditions for Fragment Sets without Pseudoknots

In the following, we define five rules to avoid the selection of a set of fragments containing pseudoknots.

An RNA secondary structure is a set \( S \) of fragments, \( \{f_1, f_2, \ldots, f_k\} \), where \( f_i > f_{i-1} \) for \( i = 1, 2, \ldots, k-1 \), and satisfies one of the following five conditions.

**Condition 1:**
\[
x(\text{start}(f_i)) > x(\text{end}(f_i)) \quad \text{and} \quad y(\text{start}(f_i)) < y(\text{end}(f_i))
\]

**Condition 2:**
\[
x(\text{start}(f_i)) > x(\text{end}(f_i)) \quad \text{and} \quad y(\text{start}(f_i)) < y(\text{end}(f_i))
\]

**Condition 3:**
\[
x(\text{start}(f_i)) > x(\text{start}(f_j)), \quad y(\text{start}(f_i)) < y(\text{end}(f_j)) \quad \text{and} \quad x(\text{end}(f_i)) > x(\text{end}(f_j))
\]

**Condition 4:**
\[
y(\text{start}(f_i)) < x(\text{start}(f_j))
\]

**Condition 5:**
\[
y(\text{start}(f_i)) < x(\text{end}(f_j)) \quad \text{and} \quad y(\text{end}(f_i)) < x(\text{start}(f_j))
\]

We illustrate above five conditions in Fig. 4.

3-2 The Conditions for Fragment Sets with Pseudoknots

In the following, we define some rules to select a set of fragments with simple pseudoknots.

An RNA secondary structure with pseudoknots is also a set \( S \) of fragments, \( S = \{f_{a_1}, f_{a_2}, \ldots, f_{a_{k-1}}\} \) where \( f_{a_i} > f_{a_{i+1}} \) for \( i = 1, 2, \ldots, k-1 \), and satisfy one of the following seven conditions.

**Condition 6:**
\[
x(\text{end}(f_i)) < x(\text{start}(f_j)) \quad \text{and} \quad y(\text{end}(f_i)) > y(\text{end}(f_j))
\]

**Condition 7:**
\[
x(\text{end}(f_i)) < x(\text{start}(f_j)), \quad y(\text{end}(f_i)) > x(\text{start}(f_j)), \quad y(\text{end}(f_i)) < x(\text{end}(f_j)), \quad y(\text{start}(f_i)) > x(\text{end}(f_j)) \quad \text{and} \quad y(\text{start}(f_j)) < y(\text{end}(f_j))
\]

**Condition 8:**
\[
x(\text{end}(f_i)) < x(\text{start}(f_j)), \quad y(\text{end}(f_i)) > x(\text{end}(f_j)), \quad y(\text{end}(f_i)) < y(\text{end}(f_j)), \quad y(\text{start}(f_i)) > y(\text{end}(f_j)) \quad \text{and} \quad y(\text{start}(f_j)) < y(\text{start}(f_j))
\]

Figure 4. The five conditions for fragment sets without pseudoknots
Figure 5. The seven conditions for fragment sets with pseudoknots

Condition 9:
\[ x(\text{start}(f_i)) < x(\text{start}(f_j)), \]
\[ x(\text{end}(f_i)) \geq x(\text{start}(f_j)), \]
\[ x(\text{end}(f_i)) < x(\text{end}(f_j)), \]
\[ y(\text{end}(f_i)) > y(\text{end}(f_j)) \]
\[ y(\text{start}(f_i)) < y(\text{start}(f_j)) \]

We illustrate the above seven conditions in Fig. 5.

Condition 10:
\[ x(\text{start}(f_i)) < x(\text{start}(f_j)), \]
\[ x(\text{end}(f_i)) \geq x(\text{start}(f_j)), \]
\[ x(\text{end}(f_i)) < x(\text{end}(f_j)), \]
\[ y(\text{end}(f_i)) > y(\text{end}(f_j)), \]
\[ y(\text{end}(f_i)) < y(\text{end}(f_j)), \]
\[ y(\text{start}(f_i)) \geq y(\text{end}(f_j)) \]
\[ y(\text{start}(f_i)) < y(\text{start}(f_j)) \]

Condition 11:
\[ x(\text{start}(f_i)) < x(\text{start}(f_j)), \]
\[ x(\text{end}(f_i)) \geq x(\text{start}(f_j)), \]
\[ y(\text{end}(f_i)) \leq y(\text{end}(f_j)), \]
\[ y(\text{start}(f_i)) > y(\text{end}(f_j)) \]
\[ y(\text{start}(f_i)) < y(\text{end}(f_j)) \]

Condition 12:
\[ x(\text{end}(f_i)) < x(\text{start}(f_j)), \]
\[ y(\text{end}(f_i)) \geq x(\text{start}(f_j)), \]
\[ y(\text{end}(f_i)) \leq x(\text{end}(f_j)), \]
\[ y(\text{start}(f_i)) \geq y(\text{end}(f_j)) \]

4 The Principle of the Pseudoknot Prediction Method

The basic principle of our pseudoknot prediction method is as follows: Suppose we are given a set of fragments of an RNA sequence. For the \(A^*\) tree searching algorithm, every node on the searching tree, it is associated with a selected fragment. Thus the path from the root of the tree to a node \(A\) constitutes a set of fragments constituting a partial solution of the problem. If node \(A\) has the largest estimated cost among all of the nodes of the searching tree, we expand it. Otherwise, node \(A\) is a terminal. Thus, according to the \(A^*\) algorithm, we have found an optimal solution which has maximal base pairs. However, this optimal solution follows the above twelve rules.
4-1 The Principle of the A* Algorithm

The A* algorithm is a good tree searching strategy and often used to solve optimization problems [14, 15]. It uses the best first strategy that finds the least cost path from the root node to a goal node for a minimization problem. For the A* tree searching algorithm, every node on the searching tree it maintains a cost function. Let \( g(A) \) denote the path length from the root node to node \( A \). Let \( h(A) \) denote the estimated path length from node \( A \) to a goal node.

Let \( f(A) \) denote the cost function of node \( A \),
\[
f(A) = g(A) + h(A)
\]
Among all of the nodes of the searching tree, the node with least cost will be selected as the next node to be expanded. The A* algorithm stops when the selected node is a goal node. The node represents an optimal solution when the estimated path length is no more than the actually length.

4-2 Descendants of a Node in the Searching Tree

Let \( S(A) \) denote the currently selected fragment set on node \( A \). Each node in our searching tree is associated with such a set \( S(A) \) of fragments \( S(A) = \{f_{a_1}, f_{a_2}, \ldots, f_{a_n}\} \) such that \( f_{a_1} > f_{a_2} > \cdots > f_{a_n} \) and every pair of \( f_{a_i} \) and \( f_{a_j} \) satisfies one of the twelve conditions described in Section 3, for \( 1 \leq a_i \leq k \). The children of node \( A \) are fragments \( f_h, f_h, \ldots, f_h \) that are smaller than \( \{f_{a_1}, f_{a_2}, \ldots, f_{a_n}\} \) and furthermore every pair of \( f_{h_i} \) and \( f_{h_j} \) satisfies one of the twelve conditions \( 1 \leq i \leq l \) and \( 1 \leq j \leq k \). Let \( Es(A) \) denote the set of legal child nodes of node \( A \). Let \( As(A) \) denote the set of expansion child nodes of node \( A \). The expansion of Node \( A \) is to create child nodes where each child node corresponds to a \( f_{a_i} \in As(A) \). Let \( Z \) be the parent of node \( A \). To obtain the children of node \( A \), we shall consider the right-most position of all fragments between \( y(start(f_{a_i})) \) to 1 and each fragment and \( f_{a_i} \) satisfies one of the twelve conditions that denote \( As(A) = As(Z) \cap Es(A) \), where \( 1 \leq j \leq k \).

In the following, we present the algorithm to find the descendants of a node in the searching tree.

Algorithm 2 To find all children of a node

Input: A set \( F \) of fragments of an RNA sequence \( F = \{f_1, f_2, \ldots, f_m\} \), a fragment \( f_{a_i} \in F \) and \( As(Z) \).

Output: A descendants of \( f_{a_i} \), \( As(A) = \{f_{h_1}, f_{h_2}, \ldots, f_{h_k}\} \)

Step 1: /* Initialization of parameters*/
\[ As(A) = \phi; \quad Es(A) = \phi \]

Step 2: /*Find the descendants of Node A*/
for \( i = a_i - 1 \) to 1 do
if \( y(start(f_j)) < y(start(f_{a_i})) \) then
    \[ Es(A) = Es(A) \cap f_j; \]
Endfor
Step 3: Output \( As(A) = As(Z) \cap Es(A); \)

After we have found the children of a node, we shall computer their cost values for the tree searching approach in order to solve the RNA secondary structure prediction problem. For A* algorithm, we need to evaluate two functions: \( g(A) \) and \( h(A) \).

4-3 Evaluating \( g(A) \)

Let \( A \) be a node and let \( X \) be the parent node of \( A \). Let \( f_j \) be the fragment added to \( X \) and \( f_i \) be the fragment added to \( A \). Let \( RL(f_i, f_j) \) denote a relationship between \( f_i \) and \( f_j \). If \( f_i \) and \( f_j \) satisfy one of twelve conditions, we set \( RL(f_i, f_j) \) equal to the serial number of the conditions; otherwise we set
Let \( RL(f_i, f_j) \) denote the length of fragment \( f_i \). Let \( CL_{i,j} \) denote the length of common bases between \( f_i \) and \( f_j \).

According to the definition of the twelve cases of conditions, we have the following rules:

**Case 1**: \( RL(f_i, f_j) = 0, 1, 4, \text{ or } 6 \),

\[ CL_{i,j} = 0; \]

**Case 2**: \( RL(f_i, f_j) = 2 \),

\[ CL_{i,j} = (y(start(f_i)) - y(end(f_j))) + 1; \]

**Case 3**: \( RL(f_i, f_j) = 3 \),

\[ CL_{i,j} = (x(end(f_i)) - x(start(f_j))) + 1; \]

**Case 4**: \( RL(f_i, f_j) = 5 \),

\[ CL_{i,j} = (y(start(f_i)) - x(start(f_j))) + 1; \]

**Case 5**: \( RL(f_i, f_j) = 7 \),

\[ CL_{i,j} = (x(end(f_i)) - y(end(f_j))) + 1; \]

**Case 6**: \( RL(f_i, f_j) = 8 \),

\[ CL_{i,j} = (y(start(f_i)) - y(end(f_j))) + 1; \]

**Case 7**: \( RL(f_i, f_j) = 9 \),

\[ CL_{i,j} = (x(end(f_i)) - x(start(f_j))) + 1; \]

**Case 8**: \( RL(f_i, f_j) = 10 \),

\[ CL_{i,j} = \min\{x(end(f_i)) - x(start(f_j)) + y(start(f_i)) - y(end(f_j)) + 2, L(f_i)\}; \]

**Case 9**: \( RL(f_i, f_j) = 11 \),

\[ CL_{i,j} = \max\{x(end(f_i)) - x(start(f_j)) + 1, x(end(f_i)) - y(end(f_j)) + 1\}; \]

**Case 10**: \( RL(f_i, f_j) = 12 \),

\[ CL_{i,j} = \max\{x(end(f_i)) - y(end(f_j)) + 1, (y(start(f_j)) - y(end(f_j)) + 1)\}. \]

For a Node \( A \) be characterized by \( \{f_{a_1}, f_{a_2}, \ldots, f_{a_n}\} \), the \( g(A) \) of Node \( A \) is

\[ g(A) = g(X) + L(f_{a_i}) - \sum_{j=a_{i-1}}^{a_i-1} CL_{a_{i,j}}, \text{ if } L(f_{a_i}) > \]

\[ \sum_{j=a_{i-1}}^{a_i-1} CL_{a_{i,j}} \]; otherwise \( g(A) = g(X) \).

### 4-4 Estimating \( h(A) \)

Let \( nopk(A) \) denote the number of maximum possible baser pairs of a secondary structure without pseudoknots from \( y(end(f_i)) \) to 1. Let \( wp(A) \) denote the number of maximum possible baser pairs of a secondary structure with pseudoknots from \( y(end(f_i)) \) to 1. In the following, we explain how to estimate \( nopk(A) \) and \( wp(A) \) for an RNA sequence.

For example, suppose we given an RNA sequence with length \( 14 \). We shall estimate \( nopk(A) \) as follows:

\[ nopk(A) = \begin{cases} \frac{n-3}{2} & \text{if } n > 4, \\ 0, & \text{otherwise}. \end{cases} \]

\( wp(A) \) is as following:

\[ wp(A) = \begin{cases} \frac{n}{2} & \text{if } n > 8, \\ nopk(A) + n - 2 \times nopk(A) - 3, & \text{if } 5 \leq n \leq 8, \\ 0, & \text{otherwise}. \end{cases} \]

For example, suppose we given an RNA sequence with length 14 and Node \( A \) is the root. \( nopk(A) \) is equal to 5 as shown in Fig. 6. \( wp(A) \) is equal to 7 as shown in Fig. 7.

![Figure 6](image-url)  
Figure 6. The maximum possible baser pairs of a secondary structure without pseudoknots
Figure 7. The maximum possible base pairs of a secondary structure with pseudoknots

Suppose that a fragment $f_i$ add on Node $A$. There are two cases shall be considered, is illustrated in Fig. 8. One is that there are not a fragment crossing $f_i$, and another is a fragment crossing $f_i$.

Case 1: A fragment crossing $f_i$

Consider Fig. 8. The first term shall consider that the maximum length of fragments contained in $f_i$, the secondary term is the maximum length of fragments between $x(start(f_i))$ and 1.

Let $L_1$ denote the length between $y(end(f_i))$ and $x(end(f_i))$ in the first term. Let $L_2$ denote the length between $x(start(f_i))$ and 1 in the second term. Let $term_1$ denote the maximum number of base pairs in the first term. Let $term_2$ denote the maximum number of base pairs in the second term.

Let $nop(A)$ and $wp(A)$ denote the number of base pairs in $A$.

$$L_1 = y(end(f_i)) - x(end(f_i)) - 1,$$

$$nop(A) = \begin{cases} L_1 - \frac{3}{2} \text{ if } L_1 > 4, \\ 0 \text{ otherwise.} \end{cases}$$

$$wp(A) = nop(A) + L_1 - 2 \times nop(A) - 3, \text{ if } 5 \leq L_1 \leq 8,$$

$$\text{otherwise.}$$

$term_1 = \max\{nop(A), wp(A)\}.$

$term_2$ is as following:

$$L_2 = x(start(f_i)) - 1,$$

$$nop(A) = \begin{cases} L_2 - \frac{3}{2} \text{ if } L_2 > 4, \\ 0 \text{ otherwise.} \end{cases}$$

$$wp(A) = (nop(A) + L_2 - 2 \times nop(A) - 3, \text{ if } 5 \leq L_2 \leq 8,$$

$$0, \text{ otherwise.}$$

$term_2 = \max\{nop(A), wp(A)\}.$

The maximum number of base pairs in Case 1 is equal to $term_1 + term_2$.

Case 2: A fragment crossing $f_i$

Let $term_3$ denote the maximum number of base pairs between $y(end(f_i))$ and 1. Let $ml$ denote a minimum length of loop in secondary structure. We shall estimate $term_3$ as following:

$$term_3 = \begin{cases} \left\lfloor \frac{L_1 - 3}{2} \right\rfloor + \left\lfloor \frac{L_2 - 3}{2} \right\rfloor, \\ \min\{L_1 - 2, \left\lfloor \frac{L_1 - 3}{2} \right\rfloor - L_2 - 2 \times \left\lfloor \frac{L_2 - 3}{2} \right\rfloor - ml\}; \end{cases}$$

$$\text{if } \left\lfloor \frac{L_2 - 3}{2} \right\rfloor > 0 \text{ then } ml = 3; \text{ otherwise } ml = 0.$$

The first term, $\left\lfloor \frac{L_1 - 3}{2} \right\rfloor$ is the maximum length of fragments contained in the fragment $f_i$, the secondary term $\left\lfloor \frac{L_2 - 3}{2} \right\rfloor$ is the maximum length of fragments between $x(start(f_i))$ and 1 and the third term $\min\{L_1 - 2, \left\lfloor \frac{L_1 - 3}{2} \right\rfloor - L_2 - 2 \times \left\lfloor \frac{L_2 - 3}{2} \right\rfloor - ml\}$ is the maximum length of a fragment crossing $f_i$.

According the above two cases, we estimate $h(A) = \max\{term_1 + term_2, term_3\}$ except $RL(f_i, f_j) = 11$. For $RL(f_i, f_j) = 11$,
Table 2. Summary of the predictions of three algorithms

<table>
<thead>
<tr>
<th>Sequence ID</th>
<th>Length</th>
<th>Sensitivity</th>
<th>Specificity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>pknotsRG</td>
<td>HPknotter</td>
</tr>
<tr>
<td>Ec_PK2</td>
<td>59</td>
<td>73.7</td>
<td>47.4</td>
</tr>
<tr>
<td>Ec_PK4</td>
<td>52</td>
<td>68.4</td>
<td>31.6</td>
</tr>
<tr>
<td>T2_gene32</td>
<td>33</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>T4_gene32</td>
<td>28</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Ec_S15</td>
<td>67</td>
<td>76.5</td>
<td>-</td>
</tr>
<tr>
<td>Ec_16S-PK505/526</td>
<td>46</td>
<td>76.7</td>
<td>33.3</td>
</tr>
<tr>
<td>TRV-PKG2_PK1</td>
<td>28</td>
<td>55.6</td>
<td>-</td>
</tr>
</tbody>
</table>

we set $h(A) = \text{term}2$.

Figure 8. Illustrate Case 1 and Case 2

4-5 Predicting RNA Secondary Structure with Pseudoknots by Using A* algorithm

In the following, we present the entire algorithm.

**Algorithm 3.** Predicting RNA secondary structure with pseudoknots

**Input:** An RNA sequence $R = r_1r_2\cdots r_n$

**Output:** Find a secondary structure of RNA with the maximum number of base pairs

**Step 1:** /* Initialization of parameters */

$A = 1$, $E(n) = \emptyset$ and $\text{Sol} = \emptyset$

**Step 2:** Find all fragments of RNA sequence by using Algorithm 1;

**Step 3:** Find the descendants of Node $A$ by using Algorithm 2;

**Step 4:** Expand all branches of Node $A$ which are associated with $E(A)$;

**Step 6:** Computer $g(A)$ and $h(A)$ for all branching nodes;

**Step 7:** If we can choose a node $A$ which has the largest estimated cost among all of the nodes of the searching tree to expand it then goto Step 8; otherwise goto Step 9.

**Step 8:** If the selected Node $A$ is a goal node then $\text{Sol} = \text{Sol} \cup A$ goto Step 7; otherwise goto Step 3.

**Step 9:** Output $\text{Sol}$

5. Experiments

We tested our method on seven sequences covering various RNA classes. The sequences were taken from PseudoBase [16] which is a database for RNA pseudoknots.

The results of prediction are measured by the sensitivity and specificity [17]. Let $RS$ be the number of base pairs in a published reference structure. Let $TP$ be the number of correctly predicted base pairs and $FP$ be the number of predicted base pairs that do not exist in the reference structure. Then the sensitivity is defined as $TP / RS$, and specificity is defined as $TP / (TP + FP)$.

We also compare with other two algorithms: pknotsRE [9] and HPknotter [13]. Table 2 shows the values of sensitivity and specificity for pknotsRE, HPknotter and our method on seven testing sequences. HPknotter responded only a partial result in Ec_S15 and did not response any result in TRV-PSG2_PK1. This could not measure them. These experimental results show that our method is quite well.

6. Conclusions

In this paper, we proposed an algorithm for predicting the secondary structure of an RNA
sequence based upon a heuristic matching. Through the experiment results, we conclude that our approach can be used to predict the secondary structure with pseudoknots quite well. At present, there are not an algorithm can really handle rather long sequences with pseudoknots. We believe that we have to use more complicated optimization techniques to handle this kind of sequences.

Reference