Approximation Algorithm and Scheme for RNA Maximum Weighted Stacking

Hengwu Li
School of Computer Science and Technology, Shandong Economic University, Jinan, China
Email: hengwuli@mail.sdu.edu.cn

Huijian Han and Zhenzhong Xu
School of Computer Science and Technology, Shandong Economic University, Jinan, China
Email: {hanhuijian, xuzz_001}@yahoo.com.cn

***Abstract*—Pseudoknotted RNA structure prediction is an important problem in bioinformatics. Existing polynomial time algorithms have no performance guarantee or can handle only limited types of pseudoknots. In this paper for the general problem of pseudoknotted RNA structure prediction, maximum weighted stacking problem is presented based on stacking actions, and its polynomial time approximation algorithm with *O*(*nlogn*) time and *O*(*n*) space and polynomial time approximation scheme are given. The approximate performance ratio of this approximation algorithm is 3. Compared with existing polynomial time algorithm, they have exact approximation performance and can predict arbitrary pseudoknots.**

***Index Terms*—RNA structure, appproximation algorithm, approximation scheme, pseudoknot**

I. Introduction

RNAs are versatile molecules: messenger RNAs carry genetic information and act as the intermediary agent between DNAs and proteins; ribosomal RNAs, transfer RNAs, and other non-coding RNAs play important structural, regulatory, and catalytic roles in cells [1]. To understand fully the various functions of RNAs, we need to first understand their structures. The primary structure of an RNA is the sequence of nucleotides (that is, the four different bases *A*, *C*, *G*, and *U*) in its single-stranded polymer. However, these sequences are not simply long strands of nucleotides. In RNA, complementary bases of guanine and cytosine pair (*G*, *C*) by forming a triple hydrogen bond, and these of adenine and uracil pair (*A*, *U*) by a double hydrogen bond; additionally, these of guanine and uracil can form a single hydrogen bond base pair. An RNA folds into a three-dimensional structure by these hydrogen bonds, that are nonconsecutive in the sequence. The three-dimensional arrangement of the atoms in the folded RNA molecule is its tertiary structure; the collection of base pairs in the tertiary structure is the secondary structure. Experimental test of RNA tertiary structure is too expensive and time consuming to meet practical need, so predicting RNA structure prediction by computer becomes a basic method and issue in computational biology [2][3].

The secondary structure of an RNA is the scaffold of its tertiary structure. RNA secondary structure prediction is the first step to predict RNA tertiary structure from RNA sequence. The best algorithm Zuker predicts RNA secondary structure without pseudoknots with $O(n^3)$ time and $O(n^2)$ space for a sequence of length n and is implemented by MFOLD and ViennaRNA programs. But they couldn't predict pseudoknots.

Among the most prevalent RNA structures is a motif known as the pseudoknot. Pseudoknots play a variety of diverse roles in biology [2]. Plausible pseudoknotted structures have been proposed [4] in 1985 and confirmed [5] in 1998 for the 3' end of several plant viral RNAs, where pseudoknots are apparently used to mimic tRNA structure. Recently, pseudoknots were confirmed in some RNAs of humans and many other species [6][7].

Currently pseudoknot is not included in the majority of the study for RNA secondary structure prediction. Finding the best secondary structure including arbitrary pseudoknots has been proved to be NP-hard [8].

Most methods for RNA folding which are capable of folding pseudoknots adopt heuristic search procedures and sacrifice optimality. Examples of these approaches include quasi-Monte Carlo searches and genetic algorithms. These approaches are inherently unable to guarantee that they have found the best structure, and consequently unable to say how far a given prediction is from optimality [9][10].

A different approach to pseudoknotted prediction is the maximum weighted matching algorithm, considering only the base paired action and no stacking action. The maximum weighted matching algorithm folds an optimal pseudoknotted structure in $O(n^3)$ time with low accuracy and seems best suited to folding sequences for which a previous multiple alignment exists [11]. Another approach adopts dynamic programming to predict the tractable subclass of pseudokonts based on complex thermodynamic model in $O(n^4)$-$O(n^6)$ time[12]-[14].

Adjacent base pairs form stack, stacking and base pairing actions in RNA molecules are the most primary and stable actions [8]. Maximum stacking problem has also attracted close attention in RNA secondary structure

doi:10.4304/jsw.6.2.233-240

prediction containing pseudoknots. Ieong presented the problem of maximum stacked base pairing number [8]. Lyngsø presented the problem of maximum stacking number, proved this problem belongs to NP-hard class and designed its polynomial time approximate scheme[15].

Lyngsø treat all stacks as the same. RNA structural experimental results indicate that the different types of stacks have the different energy, and the energy of stack is determined by the type of its base pairs. So we present maximum weighted stacking problem based on biological stacking and base pairing actions, and discuss its algorithm and complexity.

We give a polynomial time approximation algorithm with $O(n\log n)$ time and $O(n)$ space, and a polynomial time approximate scheme to predict arbitrary pseudoknots. The approximate performance ratio of this algorithm is 3.

Compared with existing polynomial time algorithm, which can handle only limited types of pseudoknots or have no performance guarantee, they have exact approximation performance and can predict arbitrary pseudoknots.

In section 2 we present the maximum weighted stacking problem. In section 3 we give a polynomial time approximation algorithm. In section 4 we give a polynomial time approximate scheme (PTAS). In section 5 we briefly conclude the paper.

II. RNA STRUCTURE PREDICTION

One single-stranded RNA molecule can be viewed as a sequence of n symbols (bases) drawn from the alphabet $\{A, C, G, U\}$. Let sequence $s=s_1s_{2\ldots}s_n$ be a single-stranded RNA molecule, where each base $s_i\in\{A, U, C, G\}$, $1\leq i\leq n$. The subsequence $s_{i,j} = s_i\ s_{i+1\ldots}s_j$ is a segment of s, $1\leq i\leq j\leq n$.

To a first approximation, one can model its secondary structure as follows. If s_i and s_j are complementary bases (A&U, C&G, U&G), then s_i and s_j may constitute a base pair (i, j). Each base can at most take part in one base pair, in other words, the set of base pairs forms a matching. It also turn out that secondary structures are noncrossing as Fig.1.

Definition1 (RNA secondary structure, S) Concretely we say that a secondary structure S on s is a set of base pairs $S=\{(i, j)\}$, where $i, j\in\{1,2,\ldots,n\}$, that satisfies the following conditions.

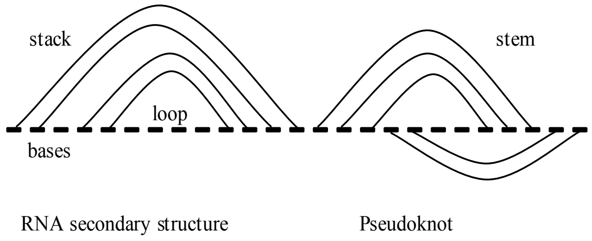

Figure 1. RNA secondary structure.and pseudoknots.

(i) (*No sharp turns*.) The ends of each pair in S are separated by at least three intervening bases; that is, if $(i, j)\in S$, then $i<j$-3.

(ii) For any pair (i, j) in S, $(i, j)\in\{(A,U), (C,G), (U,G), (U, A), (G,C), (G,U)\}$.

(iii) S is a matching: no base appears in more than one pair.

(iv) (*The noncrossin condition*.) If (i, j) and (k, l) are two pairs in S, then they are compatible, that is, they are juxtaposed (e.g. $i<j<k<l$) or nested (e.g. $i<k<l<j$).

Base pair and internal unpaired bases construct loops. If (i, j) and $(i+1, j-1)\in S$, base pairs (i, j) and $(i+1, j-1)$ constitute stack $(i, i+1{:}\ j-1, j)$, and $m(\geq 1)$ consecutive stacks form the helix $(i, i+m{:}\ j-m, j)$ with the length of $m+1$. The energy of helix $(p, p+m-1{:}\ i-m+1,i)$ is denoted as $E(p, p+m-1{:}i-m+1,i)$.

If base pairs (i, j) and (k, l) are incompatible, they constitute pseudoknots ($i<k<j<l$ or $k<i<l<j$) as Fig.1.

Stack is the only type of loops that stabilize the secondary structure [8]. Maximum stacking number problem treat all stacks as the same. In RNA, complementary bases of guanine and cytosine pair (G, C) by forming a triple hydrogen bond, and these of adenine and uracil pair (A, U) by a double hydrogen bond; additionally, these of guanine and uracil can form a single hydrogen bond base pair. RNA structural experimental results indicate that the different types of base pairs and stacks have the different energy, and the energy of stack is determined by the type of its base pairs. So we present maximum weighted stacking problem based on biological stacking and base pairing actions, and discuss its algorithm and complexity.

Definition2 (stacking fold model of pseudoknotted RNA structure prediction, SFM): For RNA sequence s, $s\in\{A, U, C, G\}^*$, a secondary structure S is a set of base pairs such that if $(i, j)\in S$ then

(i) The ends of each pair in S are separated by at least four intervening bases; that is, if $(i, j)\in S$, then $i<j$-3.

(ii) For any pair (i, j) in S, $(i, j)\in\{(A,U), (C,G), (U,G), (U, A), (G,C), (G,U)\}$.

(iii) S is a matching: no base appears in more than one pair.

(iv) If $(i+1, j-1)\in S$, then (i, j) and $(i+1, j-1)$ form stack with the weight of $w(i, i+1{:}j-1, j)$.

(v) If $(i+1, j-1),(i', j'),(i'+1, j'-1)\in S$, $s_i=s_{i'}$, $s_j=s_{j'}$ and $s_{i+1}=s_{i'+1}$, $s_{j-1}=s_{j'-1}$, then $w(i,i+1{:}j-1,j)=w(i', i'+1{:}\ j'-1,j')$. That is, the size of stacking force is determined by base pair itself and adjacent bases pair.

(vi) If $(i+1, j-1)\in S$, then the weight of S is $W(S)=\sum_{1\leq i<j\leq n}w(i, i+1{:}j-1, j)$.

So the determinant problem of maximum weighted stacking is to determine if there is a secondary structure S under SFM model with $W(S)\geq K$ for given RNA sequence s and constant K. The optimal problem of maximum weighted stacking is to find a secondary structure S with maximal weight for given RNA sequence s under SFM model.

If $W(i, i+1{:}j-1, j)=-E(i, i+1{:}j-1, j)$, then the solution of maximum weighted stacking is the minimal energy structure. Not only the weight maybe the value of energy, but also maybe the results of phylogeny analysis, or other auxiliary information.

Definition3: If (i, j), $(i+1, j-1)$, …, (k, l) are all base pair in $s_{i,j}$, $i<k<l<j$, then the structure enclosed by (i, j) and (k, l) is denoted as stem $S[i,j]$, and the length of $S[i,j]$ is denoted as $LS[i, j] = (k-i+1)$.

Lemma1： Let the length of stem $A[i, j]$ is $LA[i, j]$. We split $A[i, j]$ into k segments: $a_1, a_2, \ldots, a_k$, such that $La_1+La_2+\ldots+La_k=LA$. The number of stacks in the k segments is $\sum La_i-k=LA-k$.

Proof：

By the definition of stem, the base pairs in $A[i, j]$ are (i, j), $(i+1, j-1)$, …, $(i+LA[i, j]-1, l-LA[i, j]+1)$, and the number of stacks in $A[i, j]$ *is* $LA[i, j]-1$.

Similarly, the number of stacks in a_{i} is La_{i} -1, $1 \leq i \leq k$.

So the number of stacks in the k segments is $\sum La_i-k=LA-k$.

Theorem1 ： The maximum weighted stacking problem belongs to NP-hard.

Proof：

If $W(i, i+1{:}j-1, j)=1$, the maximum weighted stacking problem become the maximum stacking number problem, which belongs to NP-hard [15]. The proof is by reduction from the BIN PACKING problem, known to be strongly NP hard [16].

In the BIN PACKING problem we are given k items of sizes $a_1, \ldots, a_k$ and B bins each with capacity C, and have to determine whether the items fit into the bins. Or in more mathematical terms, we need to determine whether the k elements $a_1, \ldots, a_k$ can be partitioned into B sets, with the sum of elements in any set at most C. Given an instance of BIN PACKING we construct the RNA sequence s and the target K.

$$s = C^{a_1} AC^{a_2} A\ldots AC^{a_k} AAA\underbrace{G^C AG^C A\ldots AG^C}_{B \text{ substring of } G^C}$$

$$K = \sum_{i=1}^{k} a_i - k$$

As A’s can only form base pairs with U’s in SFM model, all base pairs in a legal structure for s will be (C,G) base pairs and s clearly meets the assumptions discussed above. Furthermore, any C in s is separated from any G in s by at least three other bases, so any otherwise unpaired C can form a legal base pair with any otherwise unpaired G in s.

Hence, we can find a structure S with $W(S) = K$ iff we can partition the k substrings of C’s of lengths $a_1, \ldots, a_k$ into B groups that can each be fully base paired using one substring of C consecutive G’s; i.e. the total length of the substrings of C’s in any group can be at most C. Clearly this is possible iff the original BIN PACKING problem has a solution.

The length of s is $\sum a_i+k+BC+B+1$. As BIN PACKING is strongly hard we can assume that are all $a_1, \ldots, a_k$, B, C polynomially bounded by the size of the original BIN PACKING instance.

Hence, $|s|$ is also polynomially bounded by the size of the original BIN PACKING instance. Clearly the same holds for a fair representation of the target K. Constructing s and K in time polynomial in the size of their representations is trivial.

So the theorem is true, and the maximum stacking number problem is a special case of the maximum weighted stacking problem. In fact, the maximum weighted stacking problem is a special case of the maximum weighted independent set.

III. Approximation Algorithm

We further simplify sequences s, change G into A, change U and C into B, then base pair (A,U), (C,G) and (G,U) into (A,B), the type of stacking is reduced into three classes $((A,A{:}\ B,B), (A,B{:}\ A,B), (B,A{:}\ B,A))$.

A. Approximation Algorithm

Definition4: Given arbitrary subsequences $s_{i,j}$ and $s_{k,l}$ of s, if $1 \leq i<k \leq j<l$ or $1 \leq k<i \leq l<j$, then $s_{i,j}$ conflicts with $s_{k,l}$.

Lemma2: If we group all stacks of s into two sets $((A,B{:}\ A,B),(B,A{:}\ B,A))$ and $(A,A{:}B,B)$ by the type of stack, then $W(s) \leq 2*\max(A_1, A_2)$, A_1 is the weight of maximum stacks formed by $\{(A,A{:}\ B,B)\}$ set, and A_2 is the weight of maximum stacks formed by $\{(A,B{:}\ A,B),(B,A{:}\ B,A)\}$set.

Proof：

For stacking structure, there are properties of the relative independence and partition, which make the element only form stack with the element in the same set. The stacks from the different set may conflict, so they can’t stay in S at the same time. So $W(s) \leq A_1+A_2 \leq 2*\max(A_1,A_2)$.

According to above theory, we design an approximation algorithm SA for the maximum weighted stacking problem as follows.

B. Performance

Lemma3: Given the subsequence $SB=B^k$ and $SA=\{s_{i,j}=A^x|2 \leq x \leq k\}$, we descending sort SA by the length of $s_{i,j}$, take out the subsequence from SA in turn and match that from SB, then the number of matched stacks is the biggest one.

Proof :

Let $\max(x)$ be M. When $k \leq M$, the lemma is obviously right.

When $k=M+1$, the subsequence A^M will match with B^k, then the number of stacks is k-2. If we use the subsequence $s_{i,j}$ with the length of less M , then the number of stacks is less than k-2 according to lemma1.

We suppose that the lemma is right when $k=m>M$. When $k=m+1$, the length of stacks is more than m-2 in our algorithm. If we replace the selected subsequences with the other subsequences in SA, then the length of stacks would less than or equal to the replaced one by lemma1. So the general matched stacks is less than or equal to our result.

Therefore the lemma is right.

Lemma4: For any instance I for the maximum weighted stacking problem, let the optimal solution be $OPT(I)$ with the weight of $W(OPT(I))$, and the solution of

SA is *SA*(*I*) with the weight of *W*(*SA*(*I*)), then $W(OPT(I)\ )/W(SA(I))\le 3$.

Proof:

The set of A_x and B_x is corresponding to the stacking class of (*A*, *A*: *B*, *B*). Let the optimal solution to the stacking class of (*A*, *A*: *B*, *B*) is $OPT(A_1)$ with the weight of $W(OPT(A_1))$.

The stack formed by Step1 is belong to perfect match, so the number of matched stack ≥ that of stack in $OPT(A_1)$ formed by corresponding elements.

The number of stack formed by step2 ≥ that of stack in $OPT(A_1)$ formed by corresponding elements by lemma3.

Let the number of stack in split B_x be N_x, and the number of stack formed by corresponding elements in $OPT(A_1)$ be O_x. Let the number of stack formed by *AZ* be *NAZ*, the number of stack in split *BZ* be *NBZ*, and the number of stack formed by corresponding elements in $OPT(A_1)$ be *OZ*.

1) If $AZ \ge BZ$ and $B_2 \ne \varnothing$, then $A_2 = \varnothing$, and the length of A_x matched with B_2 is at least 3.

Let the element number of B_2 is *k*, then $N_2 \ge 2k/3$. But $O_2 \le k$, so $N_2 \ge (2/3)\ O_2$.

Let the element number of B_3 is *k*, then the length of A_x corresponding to B_3 is more than 3. $N_3 \ge 4k/3$, and $O_3 \le 2k$, so $N_3 \ge (2/3)\ O_3$.

Let the element number of B_4 is *k*, then the length of A_x corresponding to B_4 is more than 2. $N_4 \ge 2k$, and $O_4 \le 3k$, so $N_4 \ge (2/3)\ O_4$.

When the length of B_x is more than 4, each split can at most reduced one stack. Let $BZ = k$, then $NAZ \ge 2k/3$. The number of reduced stack in splitting $BZ \le k/5$, so $NBZ \ge (NAZ - k/5) \ge (7/10\ )\ NAZ \ge (7/10\ )\ OZ$.

2) If $AZ \ge BZ$ and $B_2 = \varnothing$, then the length of A_x corresponding to B_3 is equal to 2, or more than 3.

When the length of A_x is equal to 2, each of B_3 can formed stack only with one A_2. Let the number of element in B_3 is *k*, the number of A_2 corresponding to B_3 is greater than *k*, therefore the optimal value can be get after segmentation.

When the length of A_x is greater than 3, $N_3 \ge 4k/3$. In addition, $O_3 \le 2k$, so $N_3 \ge (2/3)\ O_3$.

When the length of B_x is greater than 3, each split can at most reduce one stack. Let $BZ = k$, there are four cases as follows.

a) If the length of corresponding A_x is equal to 2, and B_x is an even number, then the splitting of *BZ* can not reduce the number of stack.

b) If the length of corresponding A_x is equal to 2, and B_x is an odd number, then $NAZ \ge 3k/5$. Because the reduced number $\le k/5$, $NBZ \ge (NAZ - k/5) \ge (2/3)\ NAZ \ge (2/3)OZ$.

c) If the length of corresponding A_x is greater than 2, and the length of B_x is equal to four, then $N_4 \ge 2k$, which the number of elements in B_4 is equal to *k*. In addition, $O_4 \le 3k$, so $N_4 \ge (2/3)\ O_4$.

```
//Given $s=s_1s_2\ldots s_n$, let the weight of $(A,A:B,B)$、$(A,B:A,B)$ and $(B,A:B,A)$ is $W(A,A:B,B)$ 、 $W(A,B:A,B)$ and $W(B,A:B,A)$ respectively.
Approximation algorithm SA:
step1: Search all $A^k$ subsequences and $B^k$ subsequences ($2\le k\le n$) in
    $s$, put them into sets $A_2,A_3,..,A_n$ and $B_2,B_3,.,B_n$ by the length.
    Then count $AZ=2|A_2|+3|A_3|+..+n\ |A_n|$ and $BZ=\ 2|B_2|+3|B_3|+..+n|B_n|$. At last match all subsequences of $A_x$ and $B_x$ one by
    one, delete matched elements and record the number of
    matched elements $P_x$.
step2: If $AZ\ge BZ$, descending sort the elements in $B_n,B_{n-1},.,B_2$, put
    the elements of $A_n,A_{n-1},., A_2$ on them in turn by descending sort.
    Then cut the sequence according to $B_n,B_{n-1},., B_2$ to make best of
    forming stacks and calculate the number of stacks $S_{12}$.
     If $AZ<BZ$, descending sort the elements in $A_n,A_{n-1},., A_2$, put the
     elements of $B_n,B_{n-1},., B_2$ on then in turn by descending sort.
     Then cut the sequence according to $A_n,A_{n-1},., A_2$ to make best
     of forming stacks and calculate the number of stacks $S_{12}$.
     $S_1= S_{11}+ S_{12}=\sum_{2\le x\le n}(x-1)*P_x+ S_{12}$.
Step3: Search all subsequences of $AB$ and $BA$ in $s$, then built graph
    $C_1$ and $C_2$ using the subsequences as vertexes, and evaluate the
    vertexes with $W(A,B: A,B)$ or $W(B,A:B,A)$ by the type of stack.
    If the subsequence corresponding to $C_1$ conflicts with the
    subsequence corresponding to $C_2$, draw a line between the
    vertexes.
Step4: Calculate the maximum weighted independent set formed
    by $C_1$ and $C_2$, then delete the vertexes which not belong to the
    maximum weighted independent set, and count the number of
    stacks $S_2=|C_1|/2$, $S_3= |C_2|/2$
Step5: If $W(A, A: B,B)S_1\ge W(A,B: A,B)\ S_2+ W(B,A:B,A)\ S_3$, output
$W(A, A: B,B)\ S_1$, otherwise output $W(A,B: A,B)\ S_2+W(B,A:B,A)\ S_3$.
```

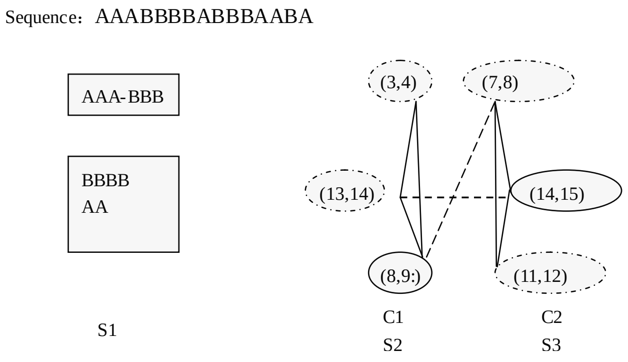

Figure 2. Example of approximation algorithm

d) If the length of corresponding A_x is greater than 2, and the length of B_x is more than four, then $NAZ \geq 2k/3$. Because the number of reduced stack $\leq k/5$, $NBZ \geq (NAZ - k/5) \geq (7/10) NAZ \geq (7/10) OZ$.

Therefore in step 2, NBZ≥ (2/3) OZ.

Similarly, when $AZ < BZ$, $NBZ \geq (2/3)$ OZ in step2.

That is, $W(A, A\text{: } B, B)S_1 \geq (2/3)\ W(OPT(A_1))$

2) Fig.C_1 and C_2 is corresponding to the class of stack (A, B: A, B) and (B, A: B, A). Let the optimal value of Fig. C_1 and C_2 be $OPT(A_2)$.

If two vertices are both in Fig.C1 or C2, and they can constitute a stack, then a line is added to connect them. Due to the relative characteristics of independence in stack structure and classification: there is no conflict between any two vertices in C1 or C2. If we don't consider the connect lines in C1 and C2, C1 and C2 will form two cliques, and arbitrary vertex in the clique can add to the optimal structure with the same probability, as shown in Fig2.

Step4 select the maximal weighted stack from the stacks formed by conflict fragments. If the C1 or C2 is an odd number, you may lose a stack.

By removing the greatest degree vertex from the vertexes with large weight and adding its adjacent vertex, the adjusted stacks have the greatest weight, as shown in Fig1. After deleting the conflict lines and vertexes, C1 and C2 still constitute two cliques with the stack of $S_2 = |C_1|/2$ and $S_3 = |C_2|/2$ respectively.

So the calculated value of weight is $W(A, B\text{: } A, B)S_2 + E(B, A\text{: } B, A)S_3 \geq W(OPT(A_2))$.

By lemma2, $W(OPT(I)) \leq 2{*}\max(W(OPT(A_1)), W(OPT(A_2))) \leq 2{*}\max((3/2)\ W(A, A\text{: } B, B)\ S_1, W(A, B\text{: } A, B)S_2 + W(B, A\text{: } B, A)S_3) \leq 3\ W(SA(I))$.

Therefore $W(OPT(I)\ )/W(SA(I)) \leq 3$.

Lemma5: Given sequence s with the length of n, algorithm SA has $O(nlogn)$ time and $O(n)$ space .

Proof:

Because the sum of A^k and B^k $<n$, so the time complexity for searching and counting in Step1 is $O(n)$, and the space complexity is $O(n)$.

The time complexity for sorting and selecting in Step2 is $O(nlogn)$, and the space complexity is $O(n)$.

Because the sum of AB segment and BA segment $<n$, so the time complexity for searching and constructing figure in Step3 is $O(n)$, and the space complexity is $O(n)$.

In step4, linear time is used to calculate the maximal weighted independent set with the degree less than 2 by dynamic programming, moreover the time and space complexity for deleting and counting is $O(n)$.

The time complexity of Step3 is $O(1)$. So the time and space complexity of this algorithm is reduce to $O(nlogn)$ and $O(n)$ respectively.

Theorem2: For any instance I for the maximum weighted stacking problem, algorithm SA is the approximation one with the polynomial time and the approximate performance ratio of 3.

Proof:

By lemma 4 and lemma 5, the theorem is right.

IV. APPROXIMATION SCHEME

A. Approximation Scheme

We divide sequence into single base, adjacent double bases, and adjacent K ($K\in$integer and $K\geq 2$) bases in all possible ways, then built graph using them as vertexes. If two subsequences can form a stem, a line is added to connect them, and the stem energy is assigned to it as its weight, we compute the maximum weighted matching for each partition, and choose the maximum weighted matching of all the partitions as the result.

As each base belongs to adjacent i bases or single base, the number of partitions is K^n, $2\leq i\leq K$. For each partition, $O(n^3)$ time is required to compute the maximum weighted matching, so the time complexity is $O(n^3K^n)$ to compute maximum weight matching of all the partitions.

But we need only consider the type and energy of paired adjacent i bases, not paired adjacent i bases themselves, $2\leq i\leq K$. So we represent the energy of paired adjacent i bases as weight, and save the number of unpaired adjacent i bases for each type of adjacent i bases in order to pair with back complementary ones. For each type of unpaired adjacent i bases, if two partitions all have the same the number of this type of unpaired adjacent i bases, and they have the same paired weight, then they have the same results.

Moreover for each type of unpaired adjacent i bases, if the partitions all have the same the number of this type of unpaired adjacent i bases, we need only choose the one with maximal weight from these partitions according to the theory of optimization.

Let $dk=\sum_{2\leq i\leq K}4^i$, matrices $S_{[x_1][x_2]\ldots[x_{dk}]}$, $SA_{[x_1]\ [x_2]\ldots[x_{dk}]}$ and $SB_{[x_1][x_2]\ldots[x_{dk}]}$ represent respectively the maximal energy of sequences $s_{1,i}$, $s_{1,i\text{-}1}$, $s_{1,i\text{-}2}$ with x_i unpaired adjacent y_{i} bases in the ith type ($1\leq i\leq dk$, $0\leq x_i\leq n_i$). Because each partition has at most $n/2$ stack, then we can reduce computation by branch-bound method. Base on above principle, we give an approximation scheme for pseudoknotted RNA secondary structure prediction.

B. Performance

Lemma6: Let $OPT(I)$ be the maximal energy that can be formed by any secondary structure of sequence I. Let $SAA[I]$ be the output by algorithm SAA. Then, $OPT(I)$ / $SAA[I] \leq 1{+}1/(K\text{-}1)$, $K\in$integer and $K\geq 2$.

```
//Let $s=s_1s_{2...}s_n$ be the input sequence, $K\in$integer and $E(S)$ is the output energy of the algorithm.
//Initially, $E(S)=\varnothing$, matrices $S=0$, $SA=0$, and $SB=0$.
$SAA(s)$
1. for $m$=2 to $K$ do
       Divide sequence $s$ into $n$-$m$+1 adjacent $m$ bases in all possible ways.
       Compute the number of each types of adjacent $m$ bases.
   end for
2. Sort all type of adjacent bases such that $n_1\le n_2\le...\le n_{dk}$, $dk=\sum_{2\le i\le K}4^i$. $q_i=n_i+1$.
3. for $i$=2 to $n$ do
       for $m$=2 to $K$ do
           Assuming the type of adjacent $m$ bases $s_{i-m+1...}s_{i-1}s_i$ is the $k$th and that of adjacent $m$ bases $s_p\ s_{p+1...}s_{p+m-1}$ paired with $s_{i-m+1...}\ s_{i-1}s_i$ is the $l$th.
           1) $S_{[x_1]...[x_k+1]...[x_{dk}]} = SB_{[x_1]...[x_k]...[x_{dk}]}$, if $S_{[x_1]...[x_k+1]...[x_{dk}]} < SB_{[x_1]...[x_k]...[x_{dk}]}$ and $x_1y_1+x_2y_2+...+x_{dk}y_{dk}\le i$-$m$. That is, $s_{i-m+1...}s_{i-1}\,s_i$ is adjacent $m$ bases waiting for pair.
           2) $S_{[x_1]...[x_l-1]...[x_{dk}]} = SB_{[x_1]...[x_l]...[x_{dk}]}+E(i-m+1,i\colon j, j+m-1)$, if $S_{[x_1]...[x_l-1]...[x_{dk}]} < SB_{[x_1]...[x_l]...[x_{dk}]} + E(p, p+m-1\colon i-m+1,i)$ and $x_1y_1+x_2y_2+...+x_{dk}y_{dk}\le i$-$m$. That is, $s_{i-m+1...}\ s_{i-1}s_i$ forms helix with adjacent $m$ bases waiting for pair.
       end for
       $SB=SA$, $SA=S$, if $x_1y_1+x_2y_2+...+x_{dk}y_{dk}\le i$.
   end for
4. $E(S)=\max(S_{[x_1][x_2]...[x_{dk}]})$, if $x_1y_1+x_2y_2+...+x_{dk}y_{dk}\le i$.
```

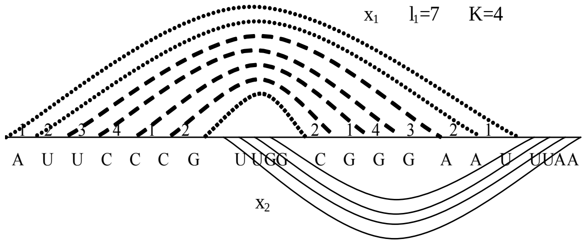

Figure 3. Example of approximation algorithm

Proof:

Let the stem in $OPT(I)$ are $x_1,x_2,...,x_m$ with the length of $l_1,l_2,...,l_m$ and the energy of Ex_1, Ex_2,..., Ex_m, $m\ge1$.

$\forall x_q\in OPT(I)$,$1\le q\le m$, if $l_q\le K$, then we choose that $E_q= Ex_q$; otherwise we divide x_q into stem with the length of 2, and group these stems into K set $X_{q1},X_{q2},...,X_{qK}$.

X_{q1}={ (i,i+1: j-1,j), (i+K+1,i+K+2: j-K-2,j-K- 1),...}

X_{q2}={ (i+1,i+2: j-2,j-1), (i+K+2,i+K+3: j-K-3, j- K-2), ... }

....

X_{qk}={ (i+K,i+K+1:j-K-1,j-K), (i+2K+1,i+2K+2: j-2K-2, j-2K-1) ,.... }

Let the energy of $X_{q1},X_{q2},...,X_{qK}$ is EX_{q1}, $EX_{q2},...,EX_{qK}$ respectively, then $Ex_q= EX_{q1} + EX_{q2} +...+ X_{qK}$.

After that, we sort $EX_{q1},EX_{q2},..,EX_{qK}$ such that $EX_{qa1}\ge EX_{qa2}\ge...\ge EX_{qaK}$ and delete the energy EX_{qaK} in order to just divide x_q into stems whose length is not more than K.

For example, for $x_1,x_2\in OPT(I)$ in Fig.3, when K=4, we divide x_1 into four groups of 1-4, then delete the energy of the second group so that x_1 is divided into two stems with the length of 2 and 4.

Let the sum of left energy is E_q, then

$E_q\ge(EX_{q1}+EX_{q2}+...+EX_{qK})(K-1)/K=(K-1)\,Ex_q/K$.

After above handle, all helices in $OPT(I)$ become the structures formed by the stems whose length is not more than K, then $\sum_{1\le q\le m}E_q\ge\sum_{1\le q\le m}(K-1)Ex_q/K=(K-1)OPT(I)/K$.

Also the length of sequence $s_{1,i}$ is i, so each partition of s meets the condition $x_1y_1 + x_2y_2 + ...+ x_{dk}y_{dk}\le i$. Obviously $SAA[I]$ is the optimal structure formed by stems whose length is not more than K.

Therefore, $SAA[I]\ge\sum_{1\le q\le m}E_q\ge(K-1)OPT(I)/K$

$OPT(I)/SAA[I]\le K/(K-1)=1+1/(K-1)$.

Lemma7: Given an RNA sequence s of length n, algorithm SAA computes the maximal energy that can be formed by s in $O((n/2dk)^{dk+1})$time and $O((n/dk)^{dk})$space.

Proof:

The time complexity of Step1 is $O(Kn)$.

The time complexity of Step2 is $O(Kn\log Kn)$.

The time complexity of Step3 is $O(K\sum_{2\le i\le n}(x_1+1)(x_2+1)....(x_{dk}+1))$.

The time complexity of Step4 is $O((x_1+1)(x_2+1)....(x_{dk}+1))$.

By the condition $x_1y_1+x_2y_2+...+x_{dk}y_{dk}\le I$, we can see that when *i is big enough*, $x_1x_2...x_{dk}\le(i/2dk)^{dk}$.

So the time complexity of algorithm SAA is $O(K\sum_{2\le i\le n}(x_1+1)(x_2+1)....(x_{dk}+1))=O(K\sum_{2\le i\le n}(i/2dk)^{dk})=O((n/2dk)^{dk+1})$.

By the condition $n_1+n_2+...+n_{dk}\le(K-1)n$ and $n_1\le n_2\le....\le n_{dk}$, we can see that the space complexity of algorithm SAA is SAA $O(q_1q_2....q_{dk})=O((n/dk)^{dk})$.

Theorem3: The Algorithm SAA is a 1+ε approximation algorithm for the problem of constructing a secondary structure S with maximal energy for given RNA sequence s under SFM model, $\varepsilon=1/(K-1)$, $K\in$integer and $K\ge2$.

Proof:

By Lemmas 6 and 7, the result follows.

V. CONCLUSION

In this paper for the general problem of pseudoknotted RNA structure prediction, maximum weighted stacking problem is presented based on stacking, and its polynomial time approximation algorithm with O ($nlogn$) time and $O(n)$ space and polynomial time approximation scheme are given. The approximate performance ratio of this approximation algorithm is 3. Compared with existing polynomial time algorithm, they have exact approximation performance and can predict arbitrary pseudoknots

ACKNOWLEDGMENT

This work was supported by NSFC under grant NO.60603077, Shan Dong Province Natural Science Foundation of China under grant Y2008G37, Y2008G29, Y2007G59, and the Research Foundation of Shandong Economic University (08009).

We thank the reviewers for their detailed comments.

REFERENCES

[1] C. Dennis, "The brave new world of RNA," *Nature*, vol.418, 122-12, 2002.

[2] D. W. Staple and S. E. Butcher, "Pseudoknots: RNA structures with diverse functions," *PLoS Biol.*, vol.3, pp. e213, 2005.

[3] D. Mathews, J. Sabina, M. Zuker, and D. Turner, "Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure," *Journal of Molecular Biology*, vol.288, 911-940, 1999.

[4] C. W. A. Pleij, K. Rietveld, and L. Bosch, A new principle of RNA folding based on pseudoknotting. *Nucl Acids Res.*, vol.3, 1721-1731, 1985.

[5] M. H. Kolk, M. van der Graff, S. S. Wijmenga, C. W. A. Pleij, H. A. Heus, and C. W. Hilbers, "NMR structure of a classical pseudoknots: interplay of single- and double-stranded RNA," *Science*, vol.280, pp. 434-438 ,1998.

[6] D. H. Mathews, and D. H. Turner, "Prediction of RNA secondary structure by free energy minimization," *Current Opinion in Structural Biology*, vol.16, pp.270-278, 2006.

[7] I. Barette, G. Poisson, and P. Gendron, "Pseudoknots in prion protein mRNAs confirmed by comparative sequence analysis and pattern searching," *Nucleic Acids Research*, vol. 29, pp. 753-758, 2001.

[8] S. Ieong, M. Y. Kao, T. W. Lam, "Predicting RNA secondary structures with arbitrary pseudoknots by maximizing the number of stacking pairs," *Journal of Computational Biology*, vol.6, pp.981-995, 2003 [2nd Symposium on Bioinformatics and Bioengineering, p.183–190, 2001].

[9] J. Ren, B. Rastegari, and H. H. Hoos, "HotKnots: heuristic prediction of RNA secondary structures including pseudoknots," *RNA*, vol.11, pp. 1494-1504, 2005.

[10] J. Ruan, G. D. Stormo,and W. Zhang, "An iterated loop matching approach to the prediction of RNA secondary structures with pseudoknots," *Bioinformatics*, vol.20, pp. 58-66, 2004.

[11] J. E. Tabaska R. B. Cary, H. N. Gabowad, and G. D. Stormo, "An RNA folding method capable of identifying pseudoknots and base triples," *Bioinformatics*, vol.14, pp. 691-699, 1998.

[12] S. R. Rivas and Eddy, "A dynamic programming algorithm for RNA structure prediction includdimg pseudoknots," *Journal of Molecular Biology*, vol. 285, pp. 2053-2068, 1999.

[13] J. Reeder, and R. Giegerich, "Design, implementation and evaluation of a practical pseudoknot folding algorithm based on thermodynamics," *BMC Bioinformatics*, vol. 5, pp.104, 2004.

[14] L. Hengwu, Z. Daming, L. Zhendong, and L. Hong, "Prediction for RNA planar pseudoknots," *Progress in Nature Science*, vol. 17(6), pp. 717-724, 2006 .

[15] R. B. Lyngsø, "Complexity of Pseudoknot Prediction in Simple Models," *LNCS*, vol.3142, p.919-931, 2004 [The 31st International Colloquium on Automata, Languages and Programming sponsored by the European Association of Theoretical Computer Science, 2004].

[16] D..Sankoff, "Simultaneous solution of the RNA folding, alignment, and protosequence problems," *SIAM J.Appl. Math*, vol.45, 810-825, 1985.

Hengwu Li, Male, was born in Jiaonan city, China, in 1969, Ph. D. in Computer Software Theory and Techniques of Shandong University, China, in 2008. Li's major field of study is analysis and design of algorithm.

He is an ASSOCIATE PROFESSOR in School of Computer Science & Technology of Shandong Economic University and Shandong Prov. He participated in the work in 1991 and has long been engaged in computer technology in teaching and research work. He has published more than 20 papers in academic journals at home and abroad. For example: Prediction for RNA Planar Pseudoknots (Journal of Progress in natural science, 2007); New Algorithm for Predicting Ribonucleic Acid Secondary Structure Including Pseudoknots (Journal of tongji university, 2004); A Polynomial Algorithm to Compute the Minimum Degree Spanning Trees of Directed Acyclic Graphs with Applications to the Broadcast Problem. (Journal of Discrete mathematics, 2008). His current research interests include analysis and design of algorithm, biological computation and electronic commerce.

Dr. Li is a member of Computer Society of Shandong Province, Shandong Prov. Key Lab of Digital Media Technology, and IEEE. Prof. Han received third prize of Shandong Province Natural Science Award in 2007.

Huijian Han, Male, was born in HeZe city, China, in December 19, 1971. In 2000, Han is a master majoring in Computer Software Theory and Techniques of Shandong University, China, and now as Computer Applied Technique Ph. D. candidate in Shandong University. Han's major field of study is texture mapping of CG.

He is a PROFESSOR, MASTER TUTOR in School of Computer Science & Technology of Shandong Economic University and Shandong Prov. Key Lab of Digital Media Technology, in Jinan city, China. He participated in the work in 1992 and has long been engaged in computer technology in teaching and research work. He has published more than 30 papers in academic journals at home and abroad and participated in the preparation of two books. For example: Computer Graphics (Beijing, China: Science Press,2005); Concise Guide to Computer Graphics (Beijing, China, Higher Education Press,2007); Determining Knots by Minimizing Energy(Beijing , China, Journal of Computer Science and Technology, 2006). His current research interests include CG&CAGD, computer simulation and algorithm design.

Prof. Han is a member of Chinese Association for System Simulation, Computer Society of Shandong Province, and Academic Committee of Shandong Prov. Key Lab of Digital Media Technology. Prof. Han received third prize of Shandong Province Natural Science Award in 2005, received first prize of Science and Technology Progress Award of People's Republic of China Ministry of Education in 2007 and received Outstanding Contribution Award from the Shandong Provincial Science and Technology Association in 2008.

Zhenzhong Xu, Male, was born in Qingzhou city, China, in January 1, 1956, Bachelor in Computer Science and Technology of Shandong University, China, in 1982. Xu's major field of study is soft engineering.

He is a PROFESSOR, MASTER TUTOR in School of Computer Science & Technology of Shandong Economic University in Jinan city, China. He participated in the work in 1982 and has long been engaged in computer technology in teaching and research work. He has published more than 20 papers in academic journals at home and abroad and participated in the preparation of three books. For example: Date Structure (Beijing, China: Science Press, 2004); Concise Guide to Computer Graphics (Beijing, China, Higher Education Press, 2007); Prediction for RNA Planar Pseudoknots (Journal of Progress in natural science, 2007). His current research interests include soft engineering, biological computation, MIS, CAD, computer simulation and control.

Prof. Xu is a member of Computer Society of Shandong Province, and Academic Committee of Shandong Prov. Key Lab of Digital Media Technology. Prof. Xu received third prize of Science and Technology Progress Award of People's Republic of China Ministry of Gym in 2005, received Outstanding Contribution Award from the Shandong Provincial Education Association in 2007, and received Outstanding Contribution Award from the Shandong Provincial Science and Technology Association in 2005.