Available online at www.sciencedirect.com
ScienceDirect
Journal of Bionic Engineering 7 (2010) 382–389

Simulating the Folding Pathway of RNA Secondary Structure Using the Modified Ant Colony Algorithm

Jun Yu[1,2], Changhai Zhang[1,2], Yuanning Liu[1,2], Xin Li[1,2]

1. *College of Computer Science and Technology*, *Jilin University*, *Changchun* 130012, *P. R. China*
2. *Key Laboratory of Symbolic Computation and Knowledge Engineering* (*Ministry of Education*, *China*) *Jilin University*, *Changchun* 130012, *P. R. China*

Abstract

A new method for simulating the folding pathway of RNA secondary structure using the modified ant colony algorithm is proposed. For a given RNA sequence, the set of all possible stems is obtained and the energy of each stem is calculated and stored at the initial stage. Furthermore, a more realistic formula is used to compute the energy of multi-branch loop in the following iteration. Then a folding pathway is simulated, including such processes as construction of the heuristic information, the rule of initializing the pheromone, the mechanism of choosing the initial and next stem and the strategy of updating the pheromone between two different stems. Finally by testing RNA sequences with known secondary structures from the public databases, we analyze the experimental data to select appropriate values for parameters. The measure indexes show that our procedure is more consistent with phylogenetically proven structures than software RNAstructure sometimes and more effective than the standard Genetic Algorithm.

Keywords: RNA, secondary structure, folding pathway, ant colony algorithm

doi: 10.1016/S1672-6529(10)60270-3

1 Introduction

A ribonucleic acid (RNA) molecule consists of a chain of ribonucleotides linked together by covalent chemical bonds and each ribonucleotide contains one of the four bases: adenine (A), cytosine (C), guanine (G) or uracil (U). For many decades, most of the researches have been focused on identifying genes which encode protein. It has been generally accepted that genetic information flows from DNA to protein by means of RNA. Recently, more and more evidences show that the functions of RNAs have been underestimated. RNAs play an important role in many biological processes such as affecting transcription and the chromosome structure, taking part in RNA modification and regulating mRNA translation. Until now, surprisingly many functions of RNAs have been found, but there are still many functions unknown.

The secondary structure is formed by RNA folding back on itself with base pairs like A-U, G-C and G-U. Research indicates that the secondary structure determines the biochemical activity of the RNA molecule. With an increasing number of sequences, experimental approaches including X-Ray crystallography and nuclear magnetic resonance spectroscopy are helpless due to the fact that they are extremely costly and time consuming. Therefore, prediction of RNA secondary structure using computer technology is one of the fundamental issues in bioinformatics and has become very popular during the last few decades. A rather reliable approach for RNA secondary structure prediction is comparative sequence analysis, in which a group of homologous sequences are aligned to reveal the common base pair model[1,2]. But if there is only one sequence, comparative sequence analysis can not be applied. The number of possible structures increases exponentially with the length of an RNA sequence, so it is not feasible to search the entire space. One of the original methods is to compute the maximum number of base pairs in an RNA sequence[3]. Currently the general

Corresponding author: Yuanning Liu
E-mail: liuyn@jlu.edu.cn

approach is to find the structure which has the minimum free energy since the researchers believe that it is the most thermodynamically stable structure[4,5]. These methods can be grossly divided into two classes: dynamic programming method and combinatorial optimization algorithm. RNAstructure, one of the famous procedures, implemented by Zuker using dynamic programming method does not include pseudoknots and its time complexity is $O(n^3)$ where n is the length of an RNA sequence[6,7]. Especially, it adopted a simplified formula to calculate the energy of multi-branch loop, which affects the accuracy of the finial structure. Furthermore, when general pseudoknots are allowed in an RNA secondary structure, the computation becomes intractable since it has been proven to be Non-deterministic Polynomial-time-hard (NP-hard)[8]. Several algorithms were restricted to some special types of pseudoknots for solvability in $O(n^4)$ to $O(n^6)$[9–11]. There are two steps when combinatorial optimization algorithm is applied to predict RNA secondary structure. The first step is to obtain all possible stems and the next step is to find the optimal stem combination whose corresponding secondary structure has the minimum free energy. The typical algorithms include helical region distribution[12] and the Genetic Algorithm (GA)[13,14]. In the process of generating next solutions, it reduces the iterative efficiency when GA selects stems randomly and does not utilize the relationship information between different stems. In addition, a stem in the next solution obtained by the crossover operator may overlap with another one. The own characteristics of GA restrict its application in RNA secondary structure prediction domain. Also suboptimal RNA secondary structures are given to avoid the inaccuracy of the free energy and ensure that they contain the real secondary structure with the greatest probability[15,16].

Ant colony algorithm inspired by the observation of the behavior of natural ant colonies was first systematically introduced by Italian scholar Dorigo in his doctoral thesis. A set of artificial ants cooperate to find the optimal solution by continuously updating the pheromone between two different nodes. Ant colony algorithm has been widely applied to combinatorial optimization problems, such as the traveling salesman problem[17], the quadratic assignment problem[18] and the graph coloring problem[19]. Experimental results show that ant colony algorithm is very effective and it has an inherent advantage in solving NP-hard problems. In our procedure the energy of each stem is calculated and stored at the first stage. A new strategy is presented to select the first stem node and the pheromone between any two stems in the solution is updated after iteration. Moreover, appropriate values for parameters are selected by testing RNA sequences with known secondary structures. What is more important is that a more realistic formula is used to calculate the energy of multi-branch loop. The modified ant colony algorithm is more appropriate to simulate the folding pathway of RNA secondary structure.

The organization of this paper is as follows. Section **2** introduces the four possible relationships between two different stems and illustrates the formula for calculating the energy of RNA secondary structure. After calculating the energy of each stem, we construct the heuristic information, the rule of initializing the pheromone, the mechanism of choosing the first and next stem and the strategy of updating the pheromone between two different stems in section **3**. In section **4**, first, appropriate values for parameters are selected by testing RNA sequences with known secondary structures. Then we compare our procedure with software RNAstructure and GA. Experimental results show that our procedure is more accurate than RNAstructure sometimes and is more effective than GA. Conclusion and our future work are given in section **5**.

2 Basic concepts

2.1 The relationship between two stems

A stem is formed by two complementary ribonucleotide segments and its length is restricted to equal to or greater than three Base Pairs (BP). A stem u, based on the traditional representation, has three components where $u(1)$, $u(2)$ and $u(3)$ denote the initial and final ribonucleotide position, and the length of the stem respectively. For two stems u and w satisfying $u(1) \le w(1)$, the topological relationship has four possibilities:

(1) u includes w: $u(1) + u(3) \le w(1)$ and $w(2) \le u(2) - u(3)$;

(2) u excludes w: $u(2) < w(1)$;

(3) u crosses w: $u(1) + u(3) \le w(1)$, $w(1) + w(3) \le u(2) - u(3) + 1$, and $u(2) \le w(2) - w(3)$;

(4) u overlaps w: there exists at least a ribonucleotide at the same position in u and w simultaneously.

The stems with relationships except (4) can coexist

in the same structure. Especially, it should be pointed out that pseudoknots will be formed in one structure if there are two stems with relationship (3). Only if the relationship between two stems is (1) or (2), it is called consistence. A stem set is consistent if the relationship between any two stems is consistent.

2.2 The free energy of RNA secondary structure

Nearest neighbor energy rule is adopted to calculate the free energy of RNA secondary structure. That is, free energies are assigned to loops rather than base pairs. These have also been called loop dependent energy rules. In order to keep this article briefly, we ignore the details which are available on website (http://www.bioinfo.rpi.edu/zukerm). The total free energy of a secondary structure is:

$$E = E_{\text{stem}} + E_{\text{hair}} + E_{\text{inter}} + E_{\text{bulge}} + E_{\text{multi}} + E_{\text{exter}}, \quad (1)$$

in which E_{stem}, E_{hair}, E_{inter}, E_{bulge}, E_{multi} and E_{exter} denote the free energy of stem, hairpin loop, interior loop, bulge loop, multi-branch loop and exterior loop, respectively. RNAstructure adopts a simplified linear formula to find optimal multi-branch loops in time proportional to n^3. Here the following more realistic formula is used to calculate the free energy of multi-branch loop:

$$E_{\text{multi}} = \text{a} + 6\text{b} + 1.75 \times \text{RT} \times \ln(l_s / 6) + \text{c} \times l_d + \delta\delta G_{\text{stack}} \quad (2)$$

where a, b, c, R and T are constants, l_s and l_d denote the number of single stranded bases and base pairs in the loop, and $\delta\delta G_{\text{stack}}$ is the sum of stem interactions energy[20,21]. The complexity of dynamic programming algorithm will take exponentially increasing time to use this more appropriate energy formula that grows logarithmically with l_s. Compared with RNAstructure, the advantage of ant colony algorithm is that the complexity is independent of the formula.

3 Methods

After obtaining the set $\boldsymbol{\Omega}$ containing all possible stems, the energy of each stem is calculated and stored at the first stage. Simulation of RNA folding pathway based on ant colony algorithm can be described as finding a consistent subset $\boldsymbol{\Omega}_{min} \subseteq \boldsymbol{\Omega}$ which fulfills the requirement:

$$E(\boldsymbol{\Omega}_{min}) = \min_{\boldsymbol{\Omega}_{subset} \subseteq \boldsymbol{\Omega}} E(\boldsymbol{\Omega}_{subset}), \quad (3)$$

where $E(\boldsymbol{\Omega}_{subset})$ is the energy of the corresponding secondary structure composed of all stems in $\boldsymbol{\Omega}_{subset}$.

3.1 The heuristic information

From the view of biological evolution, a relatively long stem is more easily formed in the early stage. So a general tendency is to choose a long stem as the next stem added into the current structure. The heuristic information from stem u to w is described as follows:

$$\eta_{uw} = \begin{cases} \dfrac{w(3)}{u(3)} \times w(3), & u \text{ and } w \text{ are consistent} \\ 0, & \text{otherwise} \end{cases}, \quad (4)$$

where η_{uw} denotes the transition strength from stem u to w. $w(3)/u(3)$ and $w(3)$ respectively reflect the relative and own importance of stem w in the transition process.

3.2 The initialization of pheromone

The initial pheromone is always defined as a constant in many other problems. Here, according to the number (expressed as $num(u)$) of the consistent stems in $\boldsymbol{\Omega}$ with stem u, a new rule is presented:

$$\tau_{uw}(0) = \begin{cases} \dfrac{1}{num(u)}, & u \text{ and } w \text{ are consistent} \\ 0, & \text{otherwise} \end{cases}, \quad (5)$$

where $\tau_{uw}(0)$ is the initial pheromone between u and w. Especially, $\tau_{uw}(i)$ represents the pheromone between u and w after the ith iteration.

3.3 The choice of the first stem

The first stem directly influences the final result and the total iteration number. We randomly generate 1000 RNA sequences whose lengths range from about 100 to 400 ribonucleotides and the statistical data demonstrates that the number of stems with length l is about four times more than the number of stems with length $l + 1$. The stability of RNA secondary structure mainly depends on the contribution of stems. Since a long stem is relatively more easily formed in the early stage, the strategy of choosing the first stem is:

$$\sum_{l_1 \geq IL} num(l_1) \geq \lambda \times \sum_{l_2 \geq 3} num(l_2), \quad (6)$$

in which $num(l)$ is the total number of stems with length l in $\boldsymbol{\Omega}$ and IL denotes the maximum integral value which satisfies Eq. (6). The reasonable value of parameter λ, which affects the convergence speed of iteration, will be

obtained from experimental data in subsection **4.1**. Then the value of *IL* can be gained. The first stem is randomly selected from the stems whose length is equal to or greater than *IL*.

3.4 The choice of the next stem

In the *i*th iteration process, according to the current stem *u*, an ant *k* chooses the next stem *w* from the allowed stem set $\boldsymbol{allowed_k}$ by Russian roulette and the probability is computed by the following formula:

$$P_{uw}^{k}(i)=\begin{cases}\dfrac{\tau_{uw}^{\alpha}(i-1)\times\eta_{uw}^{\beta}}{\sum\limits_{v\in \boldsymbol{allowed_k}}\tau_{uv}^{\alpha}(i-1)\times\eta_{uv}^{\beta}} & w\in \boldsymbol{allowed_k}\\ 0 & \text{otherwise}\end{cases}, \quad (7)$$

in which α and β are regulatory factors reflecting the relative importance of the heuristic information and the current pheromone, and $\boldsymbol{allowed_k}$ denotes the remaining stem set. The updating mechanism is that *w* will be deleted from $\boldsymbol{allowed_k}$ and added into $\boldsymbol{solution_k}$ when it is selected and the remaining stem in $\boldsymbol{allowed_k}$, which is inconsistent with *w*, will also be obliterated. After these two steps, $\boldsymbol{allowed_k}$ will not contain a stem, which is inconsistent with the stem whichever has been selected.

SelectNextStem($\boldsymbol{allowed_k}$, $\boldsymbol{solution_k}$)
Calculate $sum=\sum\limits_{v\in \boldsymbol{allowed_k}}P_{uv}^{k}(i)$;
Generate a random number $rdm\in[0,1]$;
Let $P_u^k(i)=0$;
For each stem *s* in $\boldsymbol{allowed_k}$
  Recalculate $P_u^k(i)=P_u^k(i)+P_{us}^k(i)/sum$;
  IF $rdm<P_u^k(i)$
    Add stem *s* into the current $\boldsymbol{solution_k}$;
    Update the Set $\boldsymbol{allowed_k}$;
    Break;
  End IF
End For

where the *Logistic* mapping is used to generate a random number and $rdm=4x_0(1-x_0)$, $x_0\in[0,1]$. When updating the set $\boldsymbol{allowed_k}$, the stem in $\boldsymbol{allowed_k}$, which is inconsistent with the *s* th stem, is deleted.

3.5 The strategy of updating pheromone

When $\boldsymbol{allowed_k}$ is empty for any $k\in\{1,2,\ldots,num(ant)\}$, in which *num*(*ant*) denotes the total number of ants, it means that one iteration is accomplished. The pheromone is updated and the strategy obeys the following formula:

$$\tau_{uw}(i)=(1-\rho)\times\tau_{uw}(i-1)+\sum_{k=1}^{num(ant)}\Delta\tau_{uw}^{k}(i), \quad (8)$$

in which

$$\Delta\tau_{uw}^{k}(i)=\begin{cases}\dfrac{E(\boldsymbol{solution_k})}{Q} & u \text{ and } w \text{ in } \boldsymbol{solution_k}\\ 0 & \text{otherwise}\end{cases}, \quad (9)$$

where *Q* is not a constant anymore which is adjusted according to the length of an RNA sequence, ρ is an evaporation constant, which ensures that the pheromone does not go infinitely. When the value of ρ is too large the convergence speed of algorithm is slow. On the contrary, the speed is fast and a suboptimal solution is always obtained. In most situations ρ = 0.2 is selected. Here the new strategy updates the pheromone between any two stems in $\boldsymbol{solution_k}$ which reflects the inherent relationships between two different stems.

UpdatePheromone($\tau_{uw}(i-1)$,$\tau_{uw}(i)$)
Calculate the minimum energy and store in *minenergy*;
Calculate $\tau_{uw}(i)=(1-\rho)\times\tau_{uw}(i-1)$, $\forall\, u, w\in\boldsymbol{\Omega}$;
For each ant *k*
  For the *m*th stem in $\boldsymbol{solution_k}$
    For the *n*th stem in $\boldsymbol{solution_k}$ and *n*>*m*
      $\tau_{mn}(i)=\tau_{mn}(i)+E(\boldsymbol{solution_k})/Q$;
      $\tau_{nm}(i)=\tau_{nm}(i)+E(\boldsymbol{solution_k})/Q$;
    End For
  End For
End For

where the expressions in the 6th and 7th lines denote that the pheromone between the *n*th and *m*th stems in $\boldsymbol{solution_k}$ is updated. It is worth to note that $\tau_{mn}(i)$ denotes the pheromone from the *m*th stem to the *n*th stem while $\tau_{nm}(i)$ is the opposite direction.

3.6 The outline of our procedure

The outline of our procedure is shown in Fig. 1. Where the set $\boldsymbol{subsolution}$ will keep the solution whose corresponding energy is no more than 10% in comparison with the current *minenergy*. After some iterations, if the $\boldsymbol{subsolution}$ set keeps no change, the iteration will be terminated. ACRNA2.0 is developed in VC++ environment in Windows XP, and the storage is 1 GB and CPU is Intel 3.6.

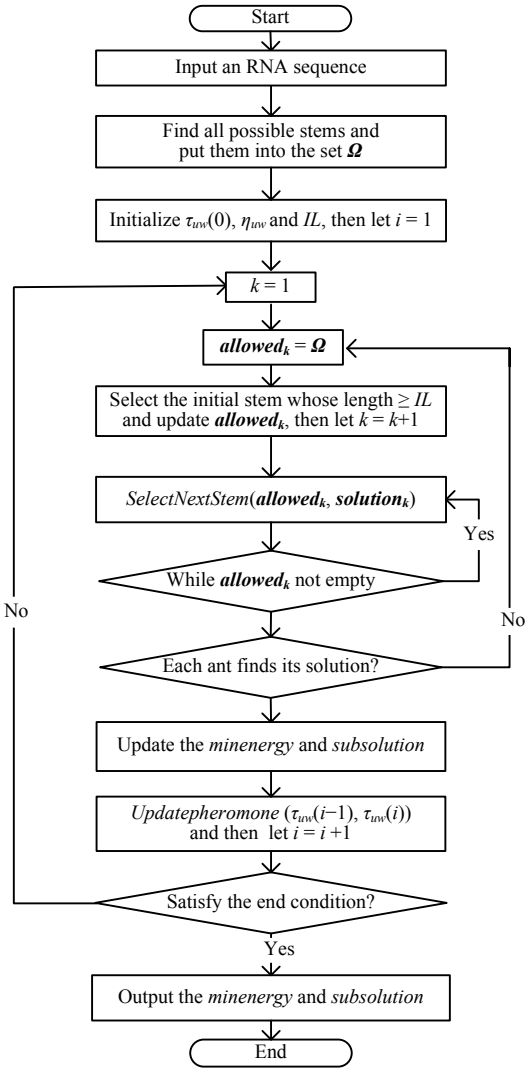

Fig. 1 The flow chart of the procedure ACRNA2.0.

Compared with GA and dynamic programming method, much improvement has been made in ACRNA2.0:

(1) After obtaining all possible stems, the energy of each stem is calculated and stored before iteration. Especially, when a loop closed by the stem u satisfying $u(2) - u(1) - 2u(3) < 8$ is only a hairpin loop, the energy of this stem is the sum of the stem and the corresponding hairpin loop. This new strategy will decrease the computational redundancy in the subsequent process.

(2) Compared with GA, ant colony algorithm embodies the inherent connections between different stems when choosing the next stem. On the contrary, GA is blind in the process of obtaining the initial solutions and crossing two solutions to generate new ones.

(3) The energy of RNA secondary structure decreases with the increase of its length. In order to simulate the folding pathway of RNA sequences with different length and enhance the efficiency of our procedure, the constant parameter Q in traditional ant colony algorithm has been modified to a variable, which varies according to the current RNA sequence.

(4) In our procedure, a more realistic formula has been used to calculate the energy of multi-branch loop compared with the dynamic programming algorithm. So ACRNA2.0 is more accurate than RNAstructure sometimes.

4 Experimental results and discussion

In this section, we will test our procedure ACRNA2.0 and analyze the experimental results to select appropriate value for parameters. Seq1 is from Genomic tRNA Database[22], and Seq2, Seq3 and Seq4 are from Viral RNA Structure Database[23].

4.1 The choice of the parameter λ

When ant colony algorithm is applied to solve many other combinatorial optimization problems, a random node is selected as the first node, whereas the effect of this approach is unfavorable. *IL* is the lower limit length of the initial stem and *Pro* denotes the proportion of the number of possible initial stems whose lengths are equal to or greater than *IL* with all stems. Four RNA sequences with different lengths have been tested and the results are listed in Table 1.

On the whole, there is no doubt that *E* decreases with the increase of iteration time. When *IL* is added *E* also decreases but rebounds a little with bigger *IL* except Seq3. Apparently, choosing the relatively long stem as the first stem will reduce the total iteration time since the long stem is easily formed in the early stage. However, in the process of disrupting the current local optimal structure and refolding to find the global stable one, a long stem may disappear. In other words, a long stem does not always exist in the optimal structure with the minimum energy. So an appropriate value of parameter λ will accelerate the iteration process and ensure that the stem included in the optimal structure is not excluded simultaneously. The change trend of Seq1, Seq2 and Seq4 is roughly the same. For Seq3 the energy gradually decreases as *IL* increases. But there is no contradiction in fact. The experiment results demonstrate that the effective value of λ is about 0.13. To be on the safe side a value which is slightly greater than 0.13 is selected.

Table 1 Experimental results of four sequences with different initial stem lengths

Name	*Len*	*IL*	*Pro*	*Itr*	*E*	*Itr*	*E*
Seq1	85	3	100%	50	−29.2	100	−31.3
		4	48.3%		−29.4		−32.1
		5	16.2%		−31.2		−32.8
		6	4.7%		−29.9		−30.7
Seq2	147	3	100%	50	−44.4	100	−46.8
		4	43.1%		−47.6		−50.5
		5	13.8%		−50.3		−53.5
		6	3.9%		−47.9		−49.3
Seq3	204	3	100%	100	−76.1	200	−79.6
		4	49.8%		−78.1		−81.0
		5	18.6%		−81.0		−82.7
		6	6.6%		−82.1		−83.4
		7	2.5%		−83.2		−84.1
Seq4	246	3	100%	100	−123.6	200	−127.1
		4	51.1%		−126.5		−131.0
		5	28.2%		−129.8		−132.9
		6	13.1%		−132.2		−135.1
		7	6.8%		−133.3		−134.9
		8	3.9%		−130.7		−132.1

**Len* is the length of an RNA sequence. *Itr* is the iteration time and *E* is the average energy of 100 times.

4.2 The choice of the parameter *Q*

The energy $E(\boldsymbol{solution}_k)$ is a negative number which ranges from scores to hundreds with the sequence's length increasing. So $E(\boldsymbol{solution}_k)/Q$ will vary dramatically and a fixed value of Q is no more suitable for updating the pheromone. A new strategy has been presented to assign Q according to the current energy of the input RNA sequence. The expression is as below:

$$Q = E_{\min} / \gamma, \tag{10}$$

where γ is a parameter and $E_{\min}$ is the minimum energy obtained in the iteration process. In this new formula, the value of $E(\boldsymbol{solution}_k)/Q$ will keep itself steady without suffering from a dramatic change. Its value is close to 1 but never exceeds 1. Four RNA sequences with different lengths have been tested and an appropriate value will be selected for γ when α = 2 and β = 1. The experiment results are shown in Fig. 2.

Taken together, when the iteration time increases, the average energy decreases and the variation of curves is more stable. For Seq1, the lowest point has been reached when γ = 0.4 on the bottom curve, but it also obtains better results on the top three curves when $\gamma = 0.6$. For Seq2, the bottom curve is almost horizontal and the energy is about the same when $\gamma = 0.6$, 0.8, 1.0 and 1.6.

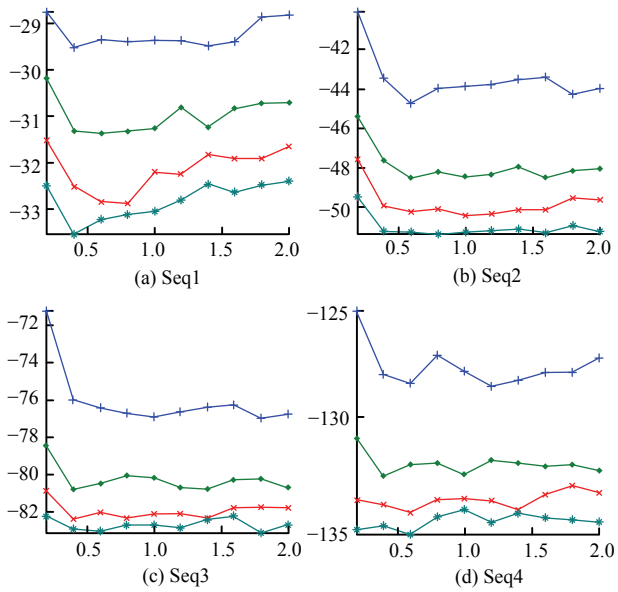

Fig. 2 *X*-axis is γ which changes from 0.2 to 2 and the partition is 0.2. *Y*-axis is the average energy of 100 times. This figure reflects the variation relationship between the energy and γ. There are four lines in each graph and each line has the same iteration time. The iteration times from top to down for Seq1 and Seq2 are 30, 60, 90 and 120 respectively and for Seq3 and Seq4 are 50, 100, 150 and 200 respectively.

For Seq3, the lowest point has been achieved when γ = 0.6 and 1.8 simultaneously. It is obviously that $\gamma = 0.6$ is a good choice for Seq4. According to the analysis, the optimal value of parameter γ is 0.6.

4.3 The efficiency of our program

In the process of secondary structure prediction, it is important to predict as many correct base pairs as possible while avoiding false positive base pairs. The traditional evaluation measures for secondary structure prediction, which we describe below, are suitable for this purpose. To measure the result accuracy, the widely used indexes of sensitivity (*Sen*), specificity (*Spe*) and Matthew's Correlation Coefficient (*MCC*) are adopted:

$$Sen = \frac{TP}{TP + FN}, \tag{11}$$

$$Spe = \frac{TP}{TP + FP}, \tag{12}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}, \tag{13}$$

in which *TP* is the number of correctly predicted base pairs, *TN* is the number of base pairs which is correctly predicted as non-matching, *FN* is the number of base

pairs in the correct structure which are not predicted and *FP* is the number of incorrectly predicted base pairs. When *Sen* is equal to 1, the predicted structure contains all of the correct base pairs and when the *Spe* is equal to 1, the predicted structure contains only the correct base pairs. Therefore, a trade-off relation exists between *Sen* and *Spe*, and perfect prediction is achieved if both *Sen* and *Spe* are equal to 1. On the other hand, *MCC* is an evaluation measure related to the balance between *Sen* and *Spe*. Note that the only correctly predicted base pairs in the predicted secondary structure can contribute to the improvement of the measures. The experimental results are shown in Table 2.

On the whole, our procedure ACRNA2.0 is faster and more accurate than RNAstructure since a more realistic formula has been adopted to calculate the energy of multi-branch loop while RNAstructure predicts a set of minimal free energy RNA secondary structures, which is time consuming. In fact, our procedure can also give suboptimal and optimal foldings of an RNA sequence, which can be stored and updated after each iteration without time increasing. The mutation operator in GA does not include the relationship information between two different stems when generating a new solution compared with ant colony algorithm. Our procedure presents a new strategy to calculate the energy of each stem after obtained, which avoids the computational redundancy in the following iteration process. Although GA and ACRNA2.0 has the same accuracy, our procedure is more effective. For Seq1, RNAstructure predicts one less base pair and one more base pair compared with the real structure. For Seq2, ACRNA2.0 predicts one less correct base pair and one less incorrect base pair compared with RNAstructure. For Seq3, ACRNA2.0 predicts one less incorrect base pair compared with RNAstructure. Three procedures have the same result for Seq4. With the addition of the length of RNA sequence, the total number of stems increases significantly. Therefore the calculating time of GA also increases obviously for Seq3 and Seq4.

The big merit of RNAstructure, which is based on dynamic programming algorithm, is that it can ensure to obtain the optimal folding every time while ACRNA2.0 may achieve the suboptimal folding occasionally. However, it is important to note that ACRNA2.0 is not influenced by the formula, which is used to calculate the energy of secondary structure.

5 Conclusion

In this paper we modify the traditional ant colony algorithm to make it more adoptable for simulating the folding process of RNA secondary structure. The energy of each stem is calculated before the following iteration process and a new approach is presented to select the initial stem whose length satisfies the requirement. Also the pheromone between any two stems in the solution is updated and the parameter Q varies with the input RNA sequence. These strategies enhance the efficiency of our procedure. Compared with RNAstructure, a more realistic formula is used to compute the energy of multi-branch loop. Furthermore, our procedure can predict a set of minimal free energy RNA secondary structures without adding computation complexity. The final experiment result demonstrates that our procedure is faster than GA and more accurate than RNAstructure.

Especially, the problem to predict RNA secondary structure including arbitrary pseudoknots is proved to be NP-Hard. Some articles have focused on this area with the restricted type of pseudoknots and different expressions have been used to calculate the energy of pseudoknots. Since the structure of pseudoknots is too intricate to estimate its energy accurately. To present an effective formula and include pseudoknots in our procedure is the next orientation.

Table 2 Experimental results of ACRNA2.0 compared with GA and RNAstructure

Name	*Len*	RNAstructure				GA				ACRNA2.0			
		Sen	*Spe*	*MCC*	*T* (s)	*Sen*	*Spe*	*MCC*	*T* (s)	*Sen*	*Spe*	*MCC*	*T* (s)
Seq1	85	95.7%	95.7%	93.0%	0.23	100%	95.8%	96.6%	0.13	100%	95.8%	96.6%	0.06
Seq2	147	88.6%	64.6%	59.3%	0.95	85.7%	65.2%	58.2%	0.77	85.7%	68.2%	61.9%	0.53
Seq3	204	95.8%	64.8%	64.0%	2.13	95.8%	65.7%	65.2%	2.89	95.8%	65.7%	65.2%	1.18
Seq4	246	91.5%	85.2%	65.4%	4.05	91.5%	85.2%	65.4%	4.71	91.5%	85.2%	65.4%	3.24

* *T*(s) denotes the time the procedure takes and its unit is second.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant No.60971089) and the Specialized Research Foundation for the Doctoral Program of Higher Education of China (Grant No.20070183057).

References

[1] Mathews D H, Turner D H. Dynalign: An algorithm for finding the secondary structure common to two RNA sequences. *Journal of Molecular Biology*, 2002, **317**, 191–203.

[2] Michiaki H, Kengo S, Hisanori K, Toutai M, Kiyoshi A. Predictions of RNA secondary structure by combining homologous sequence information. *Bioinformatics*, 2009, **25**, 330–338.

[3] Waterman M S, Smith T F. RNA secondary structure: A complete mathematical analysis. *Mathematical Biosciences*, 1978, **42**, 257–266.

[4] Zuker M, Sankoff D. RNA secondary structure and their prediction. *Bulletin of Mathematical Biology*, 1984, **46**, 591–621.

[5] Mathews D H. Using an RNA secondary structure partition function to determine confidence in base pairs predicted by free energy minimization. *RNA*, 2004, **10**, 1178–1190.

[6] Hofacker I L. Vienna RNA secondary structure server. *Nucleic Acids Research*, 2003, **31**, 3429–3431.

[7] Reuter J S, Mathews D H. RNAstructure: software for RNA secondary structure prediction and analysis. *BMC Bioinformatics*, 2010, **11**, 129.

[8] Akutsu T. Dynamic programming algorithm for RNA secondary structure prediction with pseudoknots. *Discrete Applied Mathematics*, 2000, **104**, 45–62.

[9] Rivas E, Eddy S R. A dynamic programming algorithm for RNA structure prediction including pseudoknots. *Journal of Molecular Biology*, 1999, **285**, 2053–2068.

[10] Robert M D, Niles A P. A partition function algorithm for nucleic acid secondary structure including pseudoknots. *Journal of Computational Chemistry*, 2003, **24**, 1664–1677.

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

[12] Wuju L, Jiajin W. Prediction of RNA secondary structure based on helical regions distribution. *Bioinformatics*, 1998, **14**, 700–706.

[13] Jih H C, Shuyun L, Jacob V M. Prediction of common secondary structures of RNAs: A genetic algorithm approach. *Nucleic Acids Research*, 2000, **28**, 991–999.

[14] Wiese K C, Deschenes A A, Hendriks A G. RnaPredict-An evolutionary algorithm for RNA secondary structure prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2008, **5**, 25–41.

[15] Zuker M. On finding all suboptimal foldings of an RNA molecule. *Science*, 1998, **244**, 48–52.

[16] Mathews D H. Predicting a set of minimal free energy RNA secondary structures common to two sequences. *Bioinformatics*, 2005, **21**, 2246–2253.

[17] Dorigo M, Gambardella L M. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1997, **1**, 53–66.

[18] Demirel N C, Toksari M D. Optimization of the quadratic assignment problem using an ant colony algorithm. *Applied Mathematics and Computation*, 2006, **183**, 427–435.

[19] Costa D, Hertz A. Ants can colour graphs. *Journal of the Operational Research Society*, 1997, **48**, 295–305.

[20] Mathews D H, Sabina J, Zuker M, Turner D H. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *Journal of Molecular Biology*, 1999, **288**, 911–940.

[21] Turner D H, Mathews D H. NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. *Nucleic Acids Research*, 2010, **38**, D280–D282.

[22] Lowe T M, Eddy S R. tRNAscan-SE: A program for improved detection of transfer RNA genes in genomic sequence. *Nucleic Acids Research*, 1997, **25**, 955–964.

[23] Thurner C, Witwer C, Hofacker I L, Stadler P F. Conserved RNA secondary structures in flaviviridae genomes. *Journal of General Virology*, 2004, **85**, 1113–1124.