Collaborative Learning of Estimation of Distribution Algorithms for RNA secondary structure prediction

Supawadee Srikamdee[1] and **Prabhas Chongstitvatana**[2]

ABSTRACT

Estimation of distribution algorithms (EDAs) are successfully applied in the fields of bioinformatics for tasks such as gene structure analysis, protein structure prediction, and RNA secondary structure prediction. This paper proposes a new method, namely collaborative learning of estimation of distribution algorithms, or Co-EDAs, based on an estimation of distribution algorithm for RNA secondary structure prediction using a single RNA sequence as input. The proposed method consists of two EDAs with minimum free energy objective. The Co-EDAs use both good and poor solutions to improve the algorithm's to search throughout the search space. Using information from poor solutions can indicate which area is unappealing to explore when searching with high-dimensional data. The Co-EDAs method was tested with 750 known RNA structures from RNA STRAND v2.0. That database includes data with more than 14 RNA types. The proposed method was compared to three prediction programs that are based on dynamic programming algorithms called Mfold, RNAfold, and RNAstructure. These programs are available as services on web servers. The results on average show that the Co-EDAs yields approximately 6% better accuracy than those competitors in all metrics.

Keywords: RNA Secondary Structure Prediction, Estimation of Distribution Algorithm, Collaborative Learning

1. INTRODUCTION

RNA is involved in many biological activities of organisms [1], including the process of protein synthesis, regulation of gene expression, transfer of genetic information, and as a catalyst in biomedical reactions. Understanding the functional RNA helps one to understand the various genetic diseases caused by genetic disorders. RNA molecules are directly involved with virus-infected diseases such as viral fever, HIV, influenza, measles, polio, and Ebola. An RNA virus injects malicious genetic information, propagates the diseases in the cell genome, and makes healthy RNA becomes malicious. RNA splicing is useful to cut the RNA viruses at a targeted location to stop propagation and prevent reproducing the infection in other healthy cells. This understanding has led to the production of new drugs for the treatment of genetic diseases.

RNA is a single strand of nucleotides consisting of Adenine (A), Guanine (G), Cytosine (C), and Uracil (U). It can be folded back onto itself to construct a secondary structure. RNA secondary structure is a set of canonical base pairs which include Watson-Crick (A-U, C-G, and vice versa) and GU pairs. Tertiary structure is the full three-dimensional structure. Because the secondary structure provides sufficient information to identify the functional RNA [2], specifying the secondary structure of the RNA sequence is usually the first step taken to understand biological functions for newly discovered RNA sequences. It is also used to identify unknown functional RNAs [3].

Methods of determining the RNA structure can be divided into two main groups: experimental approaches, and computational approaches [4]. Experimental approaches, such as x-ray diffraction or nuclear magnetic resonance, are reliable and provide high accuracy. However, they are expensive, time-consuming, and complicated when dealing with large numbers of cases. Therefore, it is necessary to develop mathematical and computational methods to predict the RNA secondary structure [5]. Until now, there has not been a biological RNA method that can correctly predict a true RNA secondary structure in large quantities. Thus, effective computational prediction algorithms are still needed [6].

Computing approaches for RNA secondary structures can be divided into two classes: comparative analysis, and RNA single strand folding. The comparative method requires a large number of homologous sequences for alignment. However, it provides high accuracy. When the number of homologous sequences is insufficient, predicting the secondary structure using a single sequence is more suitable. The latter method relies on the measurement of the minimum thermodynamic free energy (MFE) [7]. Another important measurement for this method is the maximum expected accuracy (MEA) [8]. Both mea-

Manuscript received on February 21, 2020 ; revised on March 21, 2020.
Final manuscript received on April 3, 2020.
[1,2] The authors are Department of Computer Engineering, Chulalongkorn University, Bangkok, 10330, Thailand., E-mail: Supawadee.Sr@student.chula.ac.th and Prabhas.C@chula.ac.th
DOI: 10.37936/ecti-cit.2020141.239871

surements are widely popular. Many studies support the MEA as an objective function. It is excellent, but not actually optimal in the real world [9]. That is why current RNA secondary structure prediction methods are mainly based on the minimum free energy algorithm [6] and remain essential tools for RNA structural biology [10].

Dynamic programming (DP) was the first computational approach used to predict RNA secondary structure [11, 12]. DP-based methods produce satisfactory results but increase the complexity of both time and space when dealing with long sequences [13, 14]. The time and space complexities of the classical dynamic programming algorithms for solving RNA single strand folding problems are $\Theta(n^3)$ and $\Theta(n^2)$, respectively, where n is the length of RNA sequence [15]. Many well-known RNA secondary structure prediction software applications, such as the Mfold web server [16], RNAfold [17], and RNAstructure [18], are based on DP. Determining the optimal RNA secondary structure is known to be NP-hard [19], so it is reasonable to utilize heuristics to solve the RNA folding prediction problem instead of using exact methods [20]. RNAPredict was invented by Wiese [21] and is based on a genetic algorithm. Simulated annealing-based methods, such as SARNA-Predict [22] and Kai Zhang, et al. [23] were introduced. The ant colony-based method was proposed by [20].

Since the structure prediction problem is difficult, some research has proposed a hybrid algorithm to improve the accuracy of the structure determination. The work of El Fatmi, et al. [13] combines the Greedy Randomized Adaptive Search Procedure (GRASP) with the Genetic Algorithm (GA). DpacoRNA [9] applied a parallel ant colony optimization strategy combined with a bi-directional LSTM recurrent neural network. CDPfold [6] uses a convolutional neural network model combined with a dynamic programming method to improve the accuracy for large-scale RNA sequences. Moreover, some studies use many objective functions. Zhang Kai, and Yulin Lv [14] proposed a multi-objective optimization algorithm including maximum base pair matching and minimum base-pair groups to evaluate the candidate solutions and adapt NSGA-II to find a group of non-dominated solutions. Despite the large body of research already published, developing improved methods for secondary structure prediction is a field of active research [24].

This paper proposes a novel collaborative learning algorithm, referred to as Co-EDAs, using two estimation of distribution algorithms for RNA secondary structure prediction. The estimation of distribution algorithms (EDAs) are powerful search approaches. The main idea of the algorithms is to maintain a probability model to represent the distribution of possible solutions and improve the model based on learning from the sampling of those solutions. The probability models can be defined a priori or learned as part of the algorithm process. From literature review, we found that EDA has been applied in the bioinformatics field before 2000. Previous efforts have applied EDA to solve the protein folding problem [25-27]. MARLEDA [28] is an attempt to apply EDA to predict RNA secondary structure. This method uses a Markov random field model and requires study of multiple sequences. They use sets of RNA sequences gathered from multiple species to predict a structure common to the entire set.Compared to MARLEDA [28], which uses comparative analysis that requires the study of multiple sequences, our method belongs to a different group of algorithms which employ one sequence. Multiple sequences methods are more accurate but they require sets of RNA sequences and infer the predicted structure from the similarity of the input sequence with them. So, apart from our preliminary research [29], EDAs have not been applied to RNA secondary structure prediction problems with a single RNA sequence. A comparison of the Co-EDAs method to COIN [29] found that both algorithms have similar results. However, the probability model of COIN is a matrix. The size of the matrix is $n \times n$, where n is the number of all possible helices. The probability model of the Co-EDAs method is a vector. The size of the vector is the number of all possible helices. So, for RNA sequences longer than 1000 nucleotides, the Co-EDAs method is more practical.

Because EDA may lose diversity and tends to prematurely converge after working for a while [30], many studies offer EDA combined with other algorithms such as TSSB-HEDA [31], which uses both the probability model of EDA and genetic operators. Liu Hongcheng, et al. [32] uses EDA together with particle swarm optimization. Tzeng Yeu-Ruey, et al. [33] combines the concept of the ant colony system (ACS) with EDA. The results show that these synergies help improve the efficiency of the proposed algorithms and provide better results.

This research converts the RNA secondary structure prediction problem into a combinatorial optimization problem. Co-EDAs consists of two EDA-based algorithms named EDA-G and EDA-L. Both EDAs run with different search behavior, but they share one probability matrix to exchange their search experiences and co-evolve toward better solutions. EDA-G conducts a global search, while EDA-L conducts a local search using a mutation operator. The motivation for using two EDAs is that in some data, one EDA achieves good prediction but it takes a long time to evolve. This causes severe problems when the sequence is longer or the number of possible helices is growing very fast. We need mutation in order to escape from local minima. We design a second EDA (EDA-L) to help in this situation. Having two EDAs allow us to tune two aspects of the search separately.

Moreover, the Co-EDAs method uses both good

and poor solutions in the evolutionary process to guide the search in the direction that produces better solutions and to avoid exploring unappealing areas. The proposed method was tested on 14 RNA classes from the RNA STRAND v2.0 database [34]. The performance evaluation compared Co-EDAs to Mfold, RNAfold, and RNAstructure. These other methods use a single sequence to perform secondary structure prediction similar to our method. These methods have been widely used and they are available as services on the web. The results show that the average results of the Co-EDAs method yields the best F-measure for almost all classes.

2. RNA SECONDARY STRUCTURE ELEMENTS

An RNA secondary structure for a given RNA sequence that is used in our experiments is defined in this section.

An RNA sequence $a_1, \ldots, a_n$ which has a length of n is defined as a set S of the ordered pairs (i, j) with $1 \leq i < j \leq n$ such that the following conditions are satisfied:

1. Watson-Crick and wobble pairs: if (i, j) is a member of S, then (a_i, a_j) is a member of $\{(A, U),(C, G),(G, C),(G, U),(U, A),(U, G)\}$.

2. No base triples: if (i, j) and (i, k) are members of S, then $j = k$; if (i, j) and (k, j) are members of S, then $i = k$.

3. Nonexistence of pseudoknots: if (i, j) and (k, l) are members of S, then the case $i < k < j < l$ will not exist.

4. The threshold requirement for hairpins: if (i, j) is a member of S, then $j - i > \theta$ for a fixed value $\theta \geq 0$.

5. The threshold requirement for helices: the length of each helix is at least τ for a fixed value $\tau \geq 0$. In this paper, we set $\tau = 2$ when an RNA sequence is less than or equal to 200 nt., and τ=3 for the rest.

A base pair (i, j) is a member of S which is stacked onto another base pair (i', j') if $i^{'} = i + 1$ and $j^{'} = j - 1$. A helix is a maximal sequence $l_0, \ldots, l_k$ of base pairs (l_i, l_j) which is stacked onto a base pair $(l_{(i+1)}, l_{(j-1)})$ $0 \leq i \leq j \leq k$, where the value of k is the length of the helix. This paper defines a secondary structure with a length of n as a set S of base pairs (i, j) where $1 \leq i < j \leq n$ and the previous conditions (1-5) are satisfied. An example of RNA secondary structure is shown in Fig. 1. The bases in areas that are not helices will form different loop types including hairpin, bulge, internal, and multibranch loops.

3. THE CO-EDAS METHOD

This research proposes a new algorithm to predict an RNA secondary structure. We call it Co-EDAs.

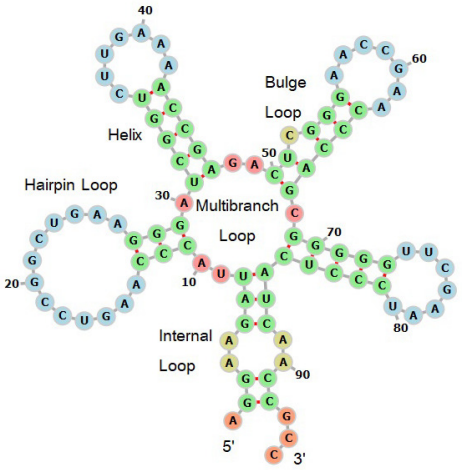

Fig.1: *The example of RNA secondary structure elements.*

The input of the algorithm is an RNA sequence. The output is a set of minimum free energy RNA secondary structures in a dot-bracket notation format. It consists of a set of brackets denoting the position of a pairing base and a set of dots denoting the position of a free base.

The proposed method has two main steps. The first step is the preparation of a set of all possible helices. The second step is the prediction of RNA secondary structure using the proposed Co-EDAs. The objective function is free energy with the nearest neighbor parameters, based on Turner 2004 [35].

3.1 Preparation of all possible helices using a helix generation algorithm

This step is performed as described in [36]. Details are as follows:

1. Calculate base pair probability of an input sequence using the RNAfold program [17].

2. Create an $N \times N$ matrix where N is the length of an input RNA sequence. Each element in the matrix is filled by base pair probability obtained from step 1.

3. Identify the region of helices in which the consecutive elements of the matrix in the diagonal direction are greater than 0. We assign a helix number (id) for reference. The information of each helix is encoded by four variables: *id, start position, end position*, and *length*. Id is a helix number, (i, j) is an initial position of a pairing base of a helix where i is a row number and j is column number in the matrix. An i position is called the start position and an j position is called the end position. A length is calculated by counting the number of consecutive base pairs from the initial position $(i, j),(i + 1, j - 1), \ldots$ to $(i + k, j - k)$, where k is the length of the helix.

An example of encoding details of each helix is displayed in Table 1.

Table 1: *Example of the encoding details of each helix.*

id	i (start position)	j (end position)	length
1	7	19	2
2	9	20	3
3	11	16	3
4	13	16	2

In Table 1, helix_1 consists of two base pairs which are (7, 19) and (8, 18), while helix_2 consists of three base pairs which are (9, 20), (10, 19) and (11, 18). In the next step, the helices are selected using the Co-EDAs method to form an RNA secondary structure. For example, the combination of helix_1 and helix_3 forms the RNA structure shown in Fig. 2.

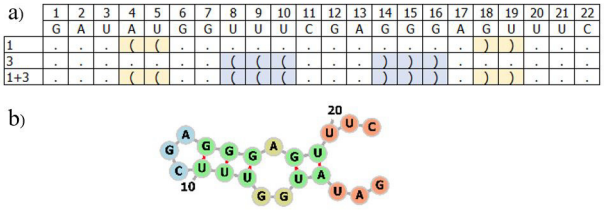

Fig.2: *The structure is created by a combination of* helix_1 *and* helix_3*.*

Different sets of helices form different RNA secondary structures and provide different free energy. Based on this concept, the RNA secondary structure prediction problem is comparable to the subset selection problem. Which helix ids should be selected to form an RNA structure to yield the minimum free energy? This research uses the Co-EDAs to select a set of helices that will be assembled into an RNA secondary structure.

3.2 Prediction of RNA secondary structure using the Co-EDAs method

The Co-EDAs method consists of two EDAs denoted EDA-G and EDA-L. The EDA-G is designed for global search and EDA-L is designed for local search. Both algorithms have the same main workflow according to the EDA framework shown in Fig.3.

Based on an EDA framework, each candidate is represented as a combination of helices. The probability model is used as a generating model. During the evolution process, each element in the probability model updated depends on the success or failure of the generated candidates. An overview of the proposed method is shown in Fig. 4.

To maintain population diversity, the Co-EDAs method applies collaborative learning with EDA-G and EDA-L. In the population generation process, EDA-G samples without replacement the helices from a set of possible helices. This method helps to prevent creating duplicate candidates. In contrast, EDA-L is designed to perform a local search. It generates the

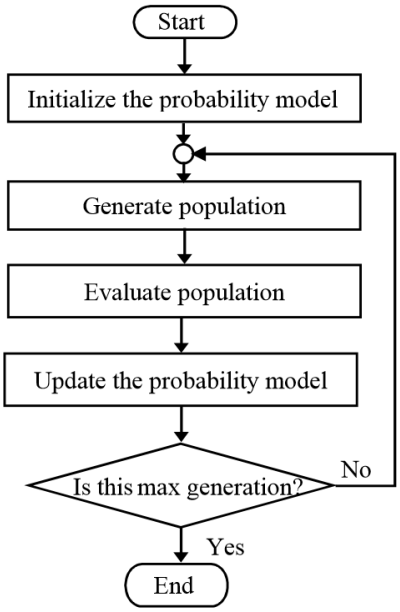

Fig.3: *Flowchart of EDA framework.*

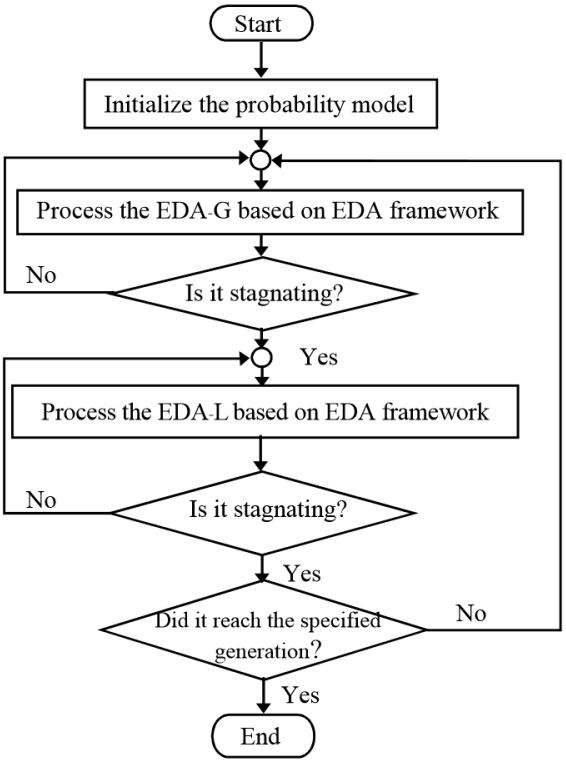

Fig.4: *The proposed Co-EDAs.*

population using a mutation operator by randomly removing some helices from the prototype structure and randomly inserting some compatible helices to produce its child structure. In this way, although the algorithm works for a period of time, the population still contains diversity. Using both good and bad candidates gives information from both sides for consideration to update the probability model.

Fig. 4 shows that the proposed method always starts with EDA-G. After that, each EDA will repeat its process according to the EDA framework until it stagnates. Criteria for detecting stagnation is known using information in the archive. During the evaluation process, a set of the n-best candidates is kept in the archive for checking the progress of evolution. Whenever there is no change of the members in this set after m generations (m is a parameter), the algorithm is considered to be stagnated and is switched to another. The algorithm repeats these steps until the stop condition is met. The details of each procedure are explained next.

3.2.1 Initialize the probability model

The probability model is used as a generating model where each dimension stores the probability that each helix is selected. During the evolution process, the probability model is updated. Increasing, decreasing, or keeping the current value is based on the success or failure of the generated candidates. In addition, traditional EDAs use only good candidates for updating the probability model. The Co-EDAs method uses both good and poor candidates.

The size of the probability model is equal to the number of all possible helices generated with the helix generation algorithm. Each element denotes a probability that the helix appears in a structure. The higher the probability, the higher the chance is of it being selected to be a part of the structure.

The initial value of each element is the average probability of all base pairs in the helix. This is in contrast to a traditional EDA in which the initial value of each element is set to 0.5.

The helix with highest probability is usually chosen first to form a structure. The other helices which conflict with the selected helices will be cut off and only compatible helices will remain. Therefore, one benefit is that the size of the search space decreases every time a helix is selected.

3.2.2 Generate Population

The Co-EDAs algorithm represents the solution of the problem in the form of many sets of compatible helices. They are called a population. The population consists of many candidates which are equal to the *population_size* parameter. In this paper, each candidate represents an RNA secondary structure which is a set of helices selected from a helices pool. Examples of encoded RNA secondary structures are shown in Fig. 5.

1	{1, 22, 42, 46, 89, 106, 124}
2	{1, 19, 22, 42, 46, 94, 103, 111, 116}
…	{7, 22, 42, 46, 91, 101, 118, 124}
N	{1, 19, 22, 41, 45, 48, 94, 103, 111, 116}

Fig.5: *Example of the population with a total of N candidate, where N is population size parameter.*

The two EDAs have different methods to generate a population. EDA-G generates each candidate using random compatible helices to form an RNA structure until there are no more compatible helices or the number of helices in this candidate is equal to a fixed value (*candidate_length*). The chance each helix are selected is based on the values in the probability model. The EDA-L generates each candidate using mutation. A random candidate is chosen from the archive containing the n-best candidates to be a prototype (parent).

Some helices in the prototype will be deleted. The other helices which are compatible with remaining helices in the prototype will be added randomly. In this way, EDA-L provides a new candidate with some information different from the prototype candidate. This is comparable to a local search. Details of the methods outlined in this section are given in Algorithm 1 and Algorithm 2.

3.2.3 Evaluate Population

In the nearest neighbor database [35], there are equations for calculating free energy and a set of parameter values used by these equations. The equations are divided by type of elements such as helix and loop types. These parameter sets can be used for predicting the stability of the nucleic acid secondary structure.

The free energy of an RNA secondary structure can be calculated by calculating the sum of all the energy values associated with the area of helix and loop types. Therefore, each candidate in the population is decoded into the position of base pairs to detect elements which are helices and loops in the structure and its free energy is calculated using nearest neighbor parameter sets. The assumption is that the lower the free energy, the higher the chance that it predicts the structure correctly compared to the known structure.

3.2.4 Probability Model Update

The probability model is used to control direction or the probability that a helix will appear in an RNA structure. So, after the initial step of the probability model, in every generation of Co-EDAs algorithm output the value of the probability model will be updated based on whether helices are successes or failures when being merged into the structure. The he-

lices found in the low free energy structures will be considered good helices. The helices found in the high free energy structures will be considered poor helices. The elements in the probability model corresponding to good helices will be increased but the elements in the probability model corresponding to poor helices will be decreased. After a while, poor helices have fewer possibilities to be chosen to create a structure. In the end, we expect that the probability model will induce the proposed algorithm to yield a set of helices that are assembled and formed with the minimum free energy.

Both EDA-G and EDA-L share the probability model but use a different method for updating the model. The EDA-G updates the probability model by classifying the population into good and poor candidate groups and uses information of both groups to update the model. The EDA-L updates the probability model by comparing the prototype candidate and its child which is produced by the mutation operator, and selects only pairs of child candidates which have lower free energy than their prototypes. EDA-L then uses this information for updating the probability model. Details in this section are described in Algorithm 1 and Algorithm 2.

3.2.5 Archive Update

This research uses an archive which stores candidates with the N-minimum free energy RNA secondary structures which are found throughout the evolution process. The number of structures in the archive is set to 20 candidates. In each generation, the current population will be compared to candidates kept in the archive. If some of the current candidates are better, they replace the worst candidates in the archive.

Archive update information is used as an indicator of improvement in each EDA. Whenever the information in the archive is not updated for m consecutive generations (m is a parameter), the master algorithm assumes that the running algorithm (EDA-G or EDA-L) is stagnating. The master algorithm will switch to another algorithm. It will change from EDA-G to EDA-L, or switch from EDA-L to EDA-G. The process repeats until a specified number of generations is reached.

In summary, EDA-G and EDA-L have different methods for generating a population and updating the probability model steps. Apart from that, both algorithms are the same. The details of the EDA-G and EDA-L algorithms are shown in Algorithm 1 and Algorithm 2 respectively.

Algorithm 1: EDA-G

Procedure Generate_Population
while the size of population is less than *POPULATION_SIZE*
Generate a candidate by randomly selecting a helix from the helices pool
end while

Procedure Update_Probability_Mode
1. Sort the population according to each member's free energy in ascending order
2. Classify the candidates into two groups: good candidates and poor candidates
   2.1 The good candidates are selected from the top c% of the population
   2.2 The poor candidates are selected from the bottom c% of the population
3. Collect all helix_ids found in the good candidates, count their frequency, and add those which are found more than one time to the Good set
4. Collect all helix_ids found in the poor candidates, count their frequency, and add those which are found more than one time to the Poor set
5. Increase the probability of every helix_id in the *Good* set
6. Decrease the probability of every helix_id in the *Poor* set

Algorithm 2: EDA-L

Procedure Generate_Population
while the size of population is less than *POPULATION_SIZE*
1. Randomly choose a candidate from the archive and use it as a prototype (parent)
2. Perform mutation to produce a new candidate (child) by randomly deleting some elements in the parent. The number of deleted helices is determined by the *Per_Remove* parameter. The other helices which are compatible with remaining helices will be added to the child randomly.

end while

Procedure Update_Probability_Model
1. Select the pairs of parent and child (P, C) which have a child with lower free energy
2. Collect all helix_id values which are removed from parents (P), count their frequency, and add those which are found more than one time to the *Poor* set
3. Collect all helix_id values which are inserted into offspring (C), count their frequency, and add those which are found more than one time to the *Good* set
4. Increase the probability of every helix_id in the *Good* set
5. Decrease the probability of every helix_id in the *Poor* set

3.3 Computational Complexity

The Co-EDAs's computational complexity is dominated by generating the population and updating the probability model procedures. A single iteration of the generating population procedure provides a computational complexity of $O(mn)$, where (i)n is chromosome length which varies with the length of RNA sequence, and (ii)m is population size.

Updating the probability model is done by sorting and then traversing the population. A single iteration of the sorting step provides a computational complexity of $O(m \log m)$, where m is the population size.

The calculation of the probability model is $O((g + p)n)$, where (i) n is chromosome length which varies with the length of RNA sequence, (ii) g is the number of good candidates, and (iii) p is the number of poor candidates.

4. RESULTS AND DISCUSSION

This section presents details of the data set used in testing the proposed algorithm, and how various parameters related to the Co-EDAs algorithm were determined. It considers other predictive methods for comparison, explains accuracy measures used to evaluate the accuracy of the compared algorithms, gives the structure prediction results, and ends with a discussion.

4.1 Data set

The proposed method was tested on 750 RNA sequences from the RNA STRAND v2.0 [34]. The test data consists of 14 RNA types. For each type, the RNA sequences with lengths in the range 100-2000 nt. and consisting of only nucleotides {'A', 'C', 'G', 'U'} were selected. They were sorted by length and RNA strand ID in ascending order. If multiple sequences had the same length, the first one was selected.

In this study, we tested the effectiveness of the proposed method on a diverse set of RNA sequences both in terms of length of sequence and type of RNA. We collected all RNA classes that were available in this database. Short RNA sequences have quite similar structure and there are only a few helices. We tested only RNAs with lengths of more than 100 nucleotides. Among all the RNA classes, only four classes have some RNA sequences longer than 2000 nucleotides. To save time in the experiment, we specified the maximum length of the tested RNA to be no longer than 2000 nucleotides.

4.2 Parameter setting

In the experiment, the population size was set to 50 candidates whose sequences had lengths less than or equal to 500 nt., and 100 candidates for the rest. The number of generations was 100 for sequences with lengths less than or equal to 500 nt., and 200 for the rest. The chromosome length is a length of an input sequence divided by 15. This parameter is set to limit the algorithm and prevent creating a structure that contains too many base pairs.

The parameters which had the largest impact on the performance of the algorithm were population size and the number of iterations in the evolution. These parameters were discovered by performing varying parameters in many experiments and these parameter settings gave the best result.

The percentage of good and poor candidates was 20. This created an appropriate selection pressure. If it is set too high (fewer percent), then the selected candidates will be too few to affect the next generation population. The percentage of the number of removed helices in process of generating the population of EDA-L was 50. The number of candidates kept in the archive was 20 candidates. The number of generations for checking for stagnation was 5. The *learning_rate* for increasing and decreasing the probability of good and poor helices was 0.01. These parameters were used for all tested RNA sequences. Each test ran 30 times and reported results of the best F-measure.

4.3 Comparison method

The three prediction methods which were compared to the proposed method were:

1. Mfold [16] which is accessible from http://unafold.rna.albany.edu/?q=mfold
2. RNAfold [17] which is accessible from http://rna.tbi.univie.ac.at/cgi-bin/RNAWebSuite/RNAfold.cgi
3. RNAstructure [18] which is accessible from http://rna.urmc.rochester.edu/RNAstructureWeb/Servers/Predict1/Predict1.html

All comparative software was run using the default parameters and the minimum free energy structure was the representative answer of the algorithm.

4.4 Accuracy Measurement

The accuracy of RNA secondary structure prediction is found by comparing to a known structure [24]. Performance of all compared algorithms was measured using these metrics:

F-measure is the harmonic mean of sensitivity and specificity which is calculated with Equation (1).

$$\text{F-measure} = (2 \times specificity \times sensitivity)/(specificity + sensitivity) \quad (1)$$

The sensitivity is the proportion of the number of predicted base pairs which are correct to a number of all base pairs in the known structure which is calculated with Equation (2).

$$\text{Sensitivity} = TP/(TP + FN) \quad (2)$$

The specificity is the proportion of the number of predicted base pairs which are correct to the number of all base pairs in the predicted structure which is calculated with Equation (3).

$$\text{Speccificity} = TP/(TP + FP) \quad (3)$$

The true positive (TP) is the number of the predicted base pairs found in the known structure.

The false positive (FP) is the number of predicted base pairs not found in the known structure.

The false negative (FN) is the number of base pairs found in the known structure but which were not predicted.

4.5 Results and Discussion

We evaluated the effectiveness of comparison methods on each RNA type. The results are shown in

Table 2. In Table 2, the first column shows the RNA types. The second column shows the comparative method. We use the best result for comparison. The 3rd – 5th columns show the average results of sensitivity, specificity and F-measure respectively, which are provided by each method for each RNA type. The highlighted areas in the table show the best method for each RNA type when considered using each metric.

Table 2: *Prediction accuracy of the comparative methods on 750 RNA sequences in term of RNA type.*

RNA type	Method	sensitivity	specificity	F-measure
5S Ribosomal RNA	RNAfold	61.45	61.19	61.22
	Mfold	59.48	62.06	60.62
	RNAstructure	56.64	55.71	56.12
	Co-EDAs	67.12	70.65	68.71
Group I Intron	RNAfold	54.83	40.75	45.04
	Mfold	51.02	39.22	42.67
	RNAstructure	49.65	38.94	41.99
	Co-EDAs	62.16	47.98	52.02
Hammerhead Ribozyme	RNAfold	51.19	20.05	28.77
	Mfold	57.14	22.75	32.52
	RNAstructure	57.14	22.31	32.06
	Co-EDAs	60.71	28.60	38.85
Other Ribosomal RNA	RNAfold	43.47	47.54	45.19
	Mfold	42.53	46.93	44.35
	RNAstructure	40.80	45.13	42.62
	Co-EDAs	51.63	57.96	54.30
Other Ribozyme	RNAfold	58.08	66.11	61.50
	Mfold	53.98	61.22	57.14
	RNAstructure	59.87	68.13	63.55
	Co-EDAs	62.32	69.97	65.77
Group II Intron	RNAfold	45.66	24.84	31.36
	Mfold	39.93	22.09	27.72
	RNAstructure	45.43	25.56	31.90
	Co-EDAs	47.02	27.47	33.74
Cis-regulatory element	RNAfold	85.42	87.97	86.61
	Mfold	80.21	83.64	81.85
	RNAstructure	80.21	81.68	80.88
	Co-EDAs	83.34	85.88	84.45
Transfer Messenger RNA	RNAfold	44.89	37.43	40.27
	Mfold	42.08	35.16	37.79
	RNAstructure	43.40	35.98	38.77
	Co-EDAs	53.90	46.49	49.26
16S Ribosomal RNA	RNAfold	34.28	31.61	32.81
	Mfold	35.04	33.16	33.99
	RNAstructure	34.44	31.73	32.94
	Co-EDAs	36.54	36.13	36.21
Transfer RNA	RNAfold	32.39	26.46	29.07
	Mfold	33.24	27.69	30.14
	RNAstructure	26.09	22.48	24.10
	Co-EDAs	45.83	39.07	41.96
Ribonuclease P RNA	RNAfold	54.98	52.92	53.68
	Mfold	52.85	52.04	52.20
	RNAstructure	51.69	52.37	51.74
	Co-EDAs	62.08	63.85	62.59
Synthetic RNA	RNAfold	42.01	44.62	42.91
	Mfold	39.99	42.50	40.84
	RNAstructure	45.19	48.44	46.38
	Co-EDAs	46.76	51.48	48.63
Signal Recognition Particle RNA	RNAfold	60.52	56.70	58.41
	Mfold	63.13	60.87	61.83
	RNAstructure	62.29	58.60	60.25
	Co-EDAs	68.22	67.90	67.85
23S Ribosomal RNA	RNAfold	24.24	18.77	20.95
	Mfold	21.78	17.09	19.02
	RNAstructure	27.31	20.35	23.13
	Co-EDAs	32.08	25.12	27.98

Considering each RNA type, we found that the Co-EDAs method yields the best prediction accuracy for 13 of the 14 RNA types. Only for Cis-regulatory element types were the results of our method lower than RNAfold, and even then they were only about 2% lower. However, in this RNA type, every method yielded good results and provided average F-measures of more than 80%.

The types which are considered to present a moderate difficulty problem, since every method provides average F-measure less than 70%, include:

1) Ribonuclease P RNA type, where the other methods yield an average F-measure of about 50% while the Co-EDAs yields a result of about 63%. Our method is better than the other methods by about 9 – 11%.

2) Signal Recognition Particle and 5S Ribosomal RNA types, where the other methods yield an average F-measure about 60% while the Co-EDAs method yields results of about 68% and 69% respectively. Moreover, our proposed method yields an average F-measure better than other methods by approximately 6 – 9% for Signal Recognition Particle, and about 7 – 13% for 5S Ribosomal type.

3) Other Ribozyme RNA type, where Mfold yields an average F-measure of less than 60%, RNAfold is 61.5%, RNAstructure is 63.55%, and Co-EDAs is 65.77%. In these tests, Co-EDAs is the best method and yields results better than the other methods by about 9 – 12%.

4) For Group I Intron type, other methods yield an average F-measure less than 50%, while the Co-EDAs yields a results of 52.02 and it is better than the other methods by about 7-10%.

The types which are considered to be quite difficult provide an average F-measure of less than 50

1) 16S Ribosomal RNA type, where the other methods yield an average F-measure of about 32 – 33%, while the Co-EDAs method yields a result of 36.13%. Our method is better than the other methods by about 2 – 3%.

2) Transfer RNA, where RNAfold and Mfold yield an average F-measure of about 30%, RNAstructure is about 24%, and Co-EDAs is about 42%. Our method is better than the other methods by about 12 – 18%.

3) Transfer Messenger RNA type, where the other methods yield an average F-measure of about 40%, while the Co-EDAs method yields a result of 49.26. Our method is better than the other methods by about 10%.

4) 23S Ribosomal RNA type, where RNAfold and RNAstructure yield an average F-measure of about 20%, Mfold is less than 20%, and the Co-EDAs is 27.98%. Our method is better than the other methods by about 5 – 9%.

5) Hammerhead Ribozyme type, where the other methods yield an average F-measure of about 29 – 33%, while the Co-EDAs yields a result of about 39%. Our method is better than the other methods by about 6 – 10%.

6) Group II Intron type, where the other methods yield an average F-measure of about 28 – 32%, while the Co-EDAs yields a result of about 39%. Our method is better than the other methods by about 2 – 6%.

From the evaluation of all results, we found that for some RNA types, various methods yield good prediction accuracy. However, the prediction accuracy in some types is low and should be improved. This is caused by the longer RNA sequence affecting the number of possible helices. As the RNA sequence becomes longer, the solution space grows larger. The size of the population and the number of iterations should be increased.

The strong points of our algorithm are:

1) EDA is flexible. It can handle many different applications which have complex representation.

2) The probability model allows us to pre-define some constraint (using the domain knowledge) in the beginning of the evolution. Therefore, scientists can add their special knowledge into the search procedure to gain better results.

3) Most work in this field employs only good candidates. This algorithm uses both good and poor candidates. Using both encourages the search to spend more effort in the region of good candidates and avoid the region of poor candidates.

4) Combining two EDAs achieves good results and allow us to tune two aspects of the search separately.

However, the weak points of the algorithm are:

1) Using two EDAs makes it more complicated.

2) There are many parameters hence making it difficult to adjust.

5. CONCLUSIONS

This paper applied the estimation of distribution algorithm to predict RNA secondary structure, which is an important problem gaining attention in current biotechnology research. The proposed method employs two EDAs working together. Each EDA is designed to have different strengths. EDA-G is responsible for global search, while EDA-L is responsible for a local search. Combining two EDAs allow us to tune aspects of the two searches separately. As shown in the results of the experiments, combining Global and Local EDA achieves very good results. On average, the Co-EDAs can predict the RNA secondary structure more accurately than comparative methods by at least 6% in all metrics. Moreover, the proposed method is flexible. The probability model can be predefined by a specialist to gain even better results.

The proposed algorithm is a viable alternative method to discover the RNA secondary structure. The method is competitive compared to dynamic programming. It supports prediction of RNA secondary structures that are diverse. This was confirmed by sample testing of RNA sequences from all databases that appear on the RNA STRAND v2.0. The proposed method integrates both global and local search and uses both positive and negative solutions to guide the search to find better solutions. These properties help the algorithm solve these problems effectively.

In future work, Co-EDAs still needs to be improved for long sequences. The prediction of RNA structure with pseudoknot should be fulfilled. Moreover, behavior or convergence of the collaborative effort of two EDAs when applying them to various problems might be analysed to contribute some benefits in the evolutionary computing field.

References

[1] A. Tripathi, K.K. Mishra, S. Tiwari, and P.C. Vashist, "Nature inspired optimization algorithm for prediction of "minimum free energy" RNA secondary structure," *Journal of Reliable Intelligent Environments*, 5(4), pp.241-257, 2019.

[2] E.P. Nawrocki, and S.R. Eddy, "Infernal 1.1: 100-fold faster RNA homology searches," *Bioinformatics*, 29(22),pp.2933-2935,2013.

[3] J. Zuber, B.J. Cabral, I. McFadyen, D.M. Mauger, and D.H. Mathews, "Analysis of RNA nearest neighbor parameters reveals interdependencies and quantifies the uncertainty in RNA secondary structure prediction," *Rna*, 24(11), pp.1568-1582, 2018.

[4] I.K. Oluoch, A. Akalin, Y. Vural, and Y. Canbay, "A review on RNA secondary structure prediction algorithms," *In 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT)*, pp. 18-23,2018.

[5] L. Wang, Y. Liu, X. Zhong, H. Liu, C. Lu, C. Li, and H. Zhang, "DMFold: A novel method to predict RNA secondary structure with pseudoknots based on deep learning and improved base pair maximization principle," *Frontiers in genetics*, vol.10, p.143, 2019.

[6] H. Zhang, C. Zhang, Z. Li, C. Li, X. Wei, B. Zhang, and Y. Liu, "A new method of RNA secondary structure prediction based on convolutional neural network and dynamic programming," *Frontiers in genetics*, vol.10, 2019.

[7] D.H. Mathews, and D.H. Turner, "Prediction of RNA secondary structure by free energy minimization," *Current opinion in structural biology*, 16(3), pp.270-278, 2006.

[8] Z.J. Lu, J.W. Gloor, and D.H. Mathews, "Improved RNA secondary structure prediction by maximizing expected pair accuracy," *Rna*, 15(10), pp.1805-1813, 2009.

[9] L. Quan, L. Cai, Y. Chen, J. Mei, X. Sun, and Q. Lyu, "Developing parallel ant colonies filtered by deep learned constrains for predicting RNA secondary structure with pseudo-knots," *Neurocomputing*, 2019.

[10] S. Poznanovic, F. Barrera-Cruz, A. Kirkpatrick, M. Ielusic, and C. Heitsch, "The challenge of RNA branch-ing prediction: a parametric analysis of multiloop initiation under thermodynamic optimization," *bioRxiv*, 2020.

[11] R. Nussinov, G. Pieczenik, J.R. Griggs, and D.J. Kleitman, "Algorithms for loop matchings," *SIAM Journal on Applied mathematics*, 35(1), pp.68-82, 1978.

[12] M. Zuker, and P. Stiegler, "Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information," *Nucleic acids research*, 9(1), pp.133-148, 1981.

[13] A. El Fatmi, A. Chentoufi, M.A. Bekri, S.

Benhlima, and M. Sabbane, "A heuristic algorithm for RNA secondary structure based on genetic algorithm," in*2017 Intelligent Systems and Computer Vision (ISCV)*, Fez, pp. 1-7, 2017.

[14] K. Zhang, and Y. Lv, "A Multiobjective RNA Secondary Structure Prediction Algorithm Based on NSGAII," *In 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation*, Guangzhou, pp. 1450-1454, 2018.

[15] R. Backofen, D. Tsur, S. Zakov, and M. Ziv-Ukelson, "Sparse RNA folding: Time and space efficient algorithms," *Journal of Discrete Algorithms*, vol.9, issue 1, pp.12-31, 2011.

[16] M. Zuker, "Mfold web server for nucleic acid folding and hybridization prediction," *Nucleic acids research*, 31(13), pp.3406-3415, 2003.

[17] R. Lorenz, S.H. Bernhart, C.H. Zu Siederdissen, H. Tafer, C. Flamm, P.F. Stadler, and I.L. Hofacker, "ViennaRNA Package 2.0," *Algorithms for molecular biology*, 6(1), p.26, 2011.

[18] S. Bellaousov, J.S. Reuter, M.G. Seetin, and D.H. Mathews, "RNAstructure: web servers for RNA secondary structure prediction and analysis," *Nucleic acids research*, 41(W1), pp. W471-W474, 2013.

[19] T. Akutsu, "Dynamic programming algorithms for RNA secondary structure prediction with pseudo-knots," *Discrete Applied Mathematics*, 104(1-3), pp.45-62, 2000.

[20] S. Takitou, and A. Taneda, "Ant colony optimization for predicting RNA folding pathways," *Computational biology and chemistry*, vol.83, p.107118, 2019.

[21] K. Wiese, A. Deschenes, and A. Hendriks, "RnaPredict—an evolutionary algorithm for RNA secondary structure prediction," in*IEEE/ACM Transac-tions on Computational Biology and Bioinformatics*, vol.5, no.1, pp.25-41, 2008.

[22] P. Grypma, and H.H. Tsang, "SARNA-Predict: Using adaptive annealing schedule and inversion mutation operator for RNA secondary structure prediction," in*2014 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM)*,Orlando, pp. 150-156, 2014.

[23] Z. Kai, W. Yuting, L. Yulin, L. Jun, and H. Juanjuan, "An efficient simulated annealing algorithm for the RNA secondary structure prediction with Pseudo-knots," *BMC genomics*, vol.20, Sup.13, pp.1-13, 2019.

[24] D.H. Mathews, "How to benchmark RNA secondary structure prediction accuracy," *Methods*, 2019.

[25] R. Santana, P. Larranaga, and J.A. Lozano, "Protein folding in 2-dimensional lattices with estimation of distribution algorithms," *In International Symposium on Biological and Medical Data Analysis*, Springer, Berlin, Heidelberg, vol.3337, pp. 388-398, 2004.

[26] R. Santana, P. Larrañaga, and J.A. Lozano, "Protein folding in simplified models with estimation of distribution algorithms," *IEEE transactions on Evolutionary Computation*, vol.12, no.4, pp.418-438, 2008.

[27] R. Santana, P. Larrañaga, and J.A. Lozano, "Combining variable neighborhood search and estimation of distribution algorithms in the protein side chain placement problem," *Journal of Heuristics*, vol.14, no.5, pp.519-547, 2008.

[28] M.E. Alden, "MARLEDA: effective distribution estimation through Markov random fields," (Doctoral dissertation), 2007.

[29] S. Srikamdee, W. Wattanapornprom, and P. Chongstitvatana, "RNA secondary structure prediction with coincidence algorithm," in *2016 16th International Symposium on Communications and Information Technologies (ISCIT)*, Qingdao, pp. 686-690, 2016.

[30] S.H. Chen, M.C. Chen, P.C. Chang, Q. Zhang, and Y.M. Chen, "Guidelines for developing effective estimation of distribution algorithms in solving single machine scheduling problems," *Expert Systems with Applications*, vol.37.issue 9, pp.6441-6451, 2010.

[31] K. Wang, S.H. Choi, and H. Lu, "A hybrid estimation of distribution algorithm for simulation-based scheduling in a stochastic permutation flowshop," *Computers & Industrial Engineering*, vol.90, pp.186-196, 2015.

[32] H. Liu, L. Gao, and Q. Pan, "A hybrid particle swarm optimization with estimation of distribution algorithm for solving permutation flowshop scheduling problem,"*Expert Systems with Applications*, vol.38, issue 4, pp.4348-4360, 2011.

[33] Y.R.Tzeng, C.L. Chen, and C.L. Chen, "A hybrid EDA with ACS for solving permutation flow shop scheduling,"*The international journal of advanced manufacturing technology*, 60(9-12), pp.1139-1147, 2012.

[34] M. Andronescu, V. Bereg, H.H. Hoos, and A. Condon,"RNA STRAND: the RNA secondary structure and statistical analysis database," *BMC bioinformatics*, 9(1), p.340, 2008.

[35] D.H. Turner, and D.H. Mathews, "NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure," *Nucleic acids research*, 38(suppl_1), pp. D280-D282, 2010.

[36] S. Montaseri, M. Ganjtabesh, and F. Zare-Mirakabad, "Evolutionary algorithm for RNA secondary structure prediction based on simulated SHAPE data," *PloS one*, vol.11, 11, e0166965, 2016.

Supawadee Srikamdee earned her B.Sc.in Computer Science from Burapha University, Thailand, in 2010 and her M.S. in Information Technology in 2012. Currently, she is a Ph.D. candidate in the department of Computer Engineering, Chulalongkorn University, Thailand. Her research involves evolutionary computation, machine learning, and bioinformatics.

Prabhas Chongstitvatana is a professor in the department of Computer Engineering, Chulalongkorn University, Thailand.He earned his B.Eng. in Electrical Engineering from Kasetsart University, Thailand, in 1980 and a Ph.D. from the department of artificial intelligence, Edinburgh University, U.K., in 1992. His research involves robotics, evolutionary computation, computer architecture, and bioinformatics. He is a lifetime member of the Thailand Engineering Institute, a senior member of the Thai Academy of Science and Technology, senior adviser of the Thai Robotics Society, and a founding member of the Thai Embedded System Association and IEEE Robotics and Automation Society. He was the president of ECTI Association of Thailand from 2012 to 2013. He was awarded "National Distinguished Researcher" by the National Research Council of Thailand in 2009. His current interest is in building a Quantum computer.