CHAPTER 1 INTRODUCTION

1.1 Problem statement

Synthetic biology is the field that designs and builds biological devices and systems for specific purposes. It is a combination of biology and engineering to develop new functional devices of organisms such as DNAs, RNAs and proteins. The functions of RNAs depend on its structure. For examples, small RNAs have various functions including regulation of gene and protein expression, and ribozymes or catalytic RNAs can digest other RNA molecules [1].

In the past, many researchers attempted to develop RNA molecules for controlling gene expression. Before synthesizing RNA molecules, they must design the sequences of RNAs. Currently, RNAs have been designed by programs in computers for more convenient and efficient. The favorite computational tools include RNAsoft, RNAstructure, Vienna RNA package and so on. These tools, however, are able to design single RNA molecules. But in the real world, RNAs form secondary structure and interact with other RNAs to function within cells. Thus, there is a need for development of a new computational tool that helps design multiple RNA molecules that can form and interact to one another following target designs.

Recently, a group of researchers has developed a tool based on the domain-based design (DD) program [2]. The DD program is a tool that can design a set of DNAs for interaction by random bases (A, T, G and C) in each sequence. Thaiprasit et al (2012) applied this program to design multiple RNA molecules that functions as desire. However, because this domain based RNA design tool only lies on a random design, so it does not guarantee an optimal design. Therefore, an optimization step should be added to the exiting version such that an optimal set of functional RNA-RNA interaction is achieved.

GA is the search technique that is based on the principle of natural selection in biology. It is used to generate or search optimal solutions by optimizing an objective function using the technique of natural evolution, such as selection, crossover and mutation. GA is widely applied to find optimal solutions in many problems in engineering, arithmetic, and genetics.

In this study, we plan to incorporate the Genetic Algorithm into the existing domain-based RNA design tool developed by Thaiprasit et al (2012) in hope for obtaining an optimal set of RNA-RNA interaction. A stability factor based on the Gibbs' free energy of hybridized RNA molecules as well as a similarity factor based on the Hamming Distance will be used as an objective function

1.2 Objective

- To develop an automatic design tool of RNA-RNA interaction using genetic algorithm.

1.3 Scope of work

- Develop an automatic design tool by using C language programming.
- Use the genetic algorithm to optimize the design of RNA sequences.
- Optimize design of RNA sequences by minimizing a fitness function that is based on the Gibbs' free energy of hybridized sequences and the Hamming Distance.

CHAPTER 2 THEORIES AND LITERATURE REVIEWS

2.1 Ribonucleic acid (RNA)

Ribonucleic acid or RNA is an important biological molecule. It is a linear polymer of nucleotides, which are consisted of nucleobases, ribose sugar rings and phosphate groups. Figure 2.1 shows the chemical structure of a molecule of nucleotide. A nucleotide can be divided into two parts. The first part is backbone, which is sugar and phosphate group. The other part is nucleobase or base. There are four types of bases in a RNA molecule: Adenine (A), Cytosine (C), Guanine (G) and Uracil (U). Adenine and Guanine are called purines, which are bases that contain two rings in structure. And the other two bases that contain only one ring are called pyrimidine.

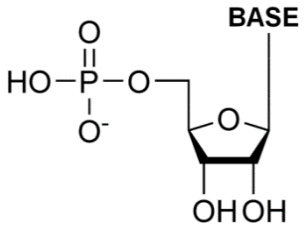

Figure 2.1 A basic structure of nucleotide [3]

For an RNA strand, many nucleotides are linked to form a polymer. The linking occurs between the oxygen atoms on the 5'-phosphate and the 3'-hydroxyl groups of ribose sugar as shown in Figure 2.2, where the letter prime (') is used to distinguish between base and ribose sugar.

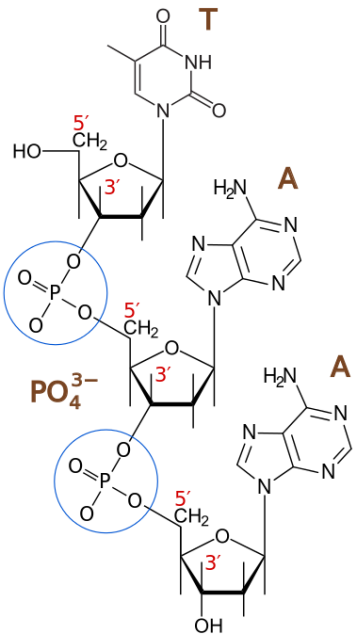

Figure 2.2 A basic structure of RNA strand [4]

The linking is a covalent bond in phosphate linkage form. The polymer of nucleotides can be either short chain (less than fifty nucleotides), called oligonucleotide, or long chain (many thousand nucleotides), called polynucleotide. The length of RNA depends on function and class of an RNA strand [5].

RNA play a prominent role in a variety of cellular function, mainly in the protein synthesis. A protein-encoded gene on DNA is transcribed to messenger RNA or mRNA in nucleus, which is called transcription. Such mRNA will then be translated to produce amino acid sequence and form proteins in ribosome that has ribosomal RNA or rRNA for linking amino acids. The amino acids from nucleus are transferred to ribosome for linking amino acids by transfer RNA or tRNA. It is a carrier that specific with amino acid and has an anticodon (reverse of RNA triplet for each amino acid). The proteins from synthesis can function as catalysts of biological reactions, controllers of gene expression or sensors that respond to cellular signals.

2.1.1 RNA-RNA interaction

Usually, RNA in a cell exists in a single strand form. It is a flexible strand, which can fold itself by forming hydrogen bonding between those bases. An RNA is the sequence of four types of nucleotides that depend on types of bases, which are A, U, G and C. This sequence has the direction from 5’ to 3’ of sugar. In each nucleotide, a base can pair with another base located in another nucleotide by a hydrogen bond. The base pairs of Watson- Crick base pairs consist of the pairs of adenine with uracil (AU) and guanine with cytosine (GC) as well as the wobble base pair of guanine with uracil (GU). This thesis does not consider about wobble base pairs because it is not stable. The hydrogen bonds occur between amine and carbonyl groups on the base. The base complement of AU forms in 2 hydrogen bonds, while GC forms in three hydrogen bonds as shown in Figure 2.3 [6].

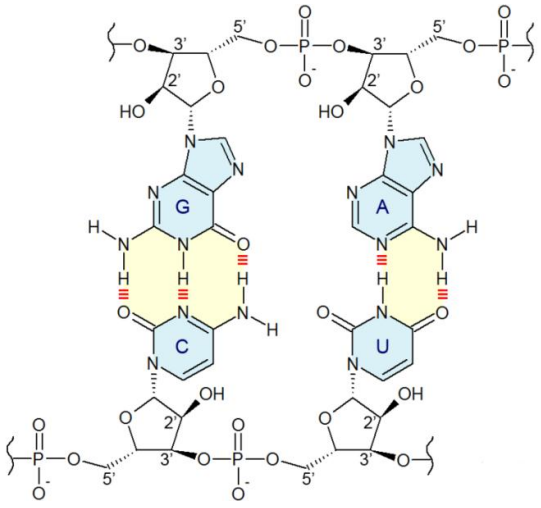

Figure 2.3 The example structures of base paring in double stranded RNA [7]

Thus, the structure of RNA depends on the sequence of RNA that can be divided into three structures. The first is the primary structure. It is written as a string by using the letters A, U, G and C that start from 5' to 3'. For the secondary structure, the RNA strand folds onto itself by forming intra-molecule hydrogen bonds between base pairs. And the last one is the tertiary structure. It is the interaction of secondary structures. The examples of interaction in tertiary structure are hairpins, bulge contracts, pseudoknots and so on as shown in Figure 2.4 [5]. This thesis considers on the tertiary structure.

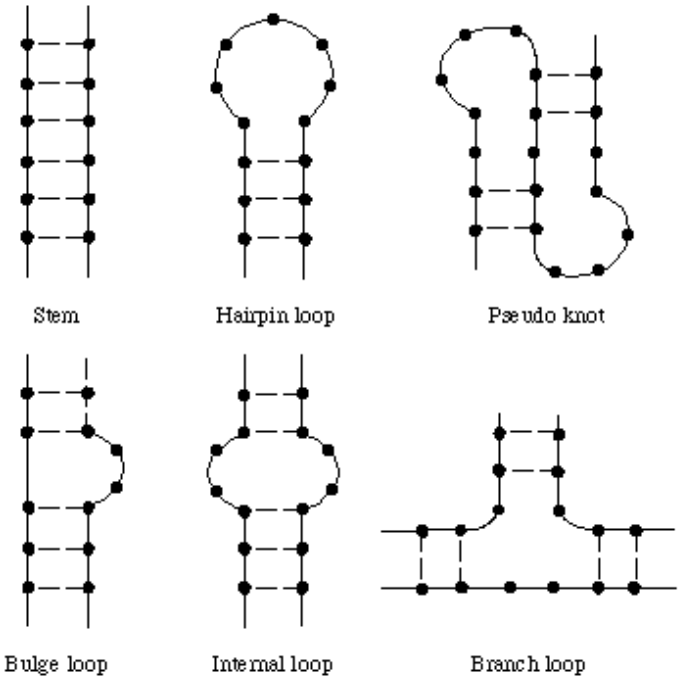

Figure 2.4 Interaction in tertiary structures [8]

RNA-RNA interactions are the basic mechanism of cellular regulation. They can be found in many contexts, which are the bind of small RNAs and a larger RNA target including the regulation of translation and transcription, the targeting of chemical modification or the insertion editing [9].

The formation of RNA-RNA interaction structures is more complex, which have the variety of RNA classes. The classes of RNA are divided into (i) micro RNA or miRNA (it contains less than 20-22 nucleotides which control the gene expression by hybridization with mRNA.) [10], (ii) small interfering RNA or siRNA (it is a double-stranded RNA molecule, which contain 20-25 base pairs.) [11], (iii) small nuclear ribonucleic acid or snRNA (it is a class of small RNA in nuclear.) [12], (iv) guide RNA or gRNA (it is the RNA that guide to insert or delete in RNA editing.) [13] and (v) small nucleolar RNA or snoRNA (it is a class of small RNA that guide the chemical modification.) [14]. Their roles are the regulation of gene expression. The RNA-base mechanism of these structures needs deeper understanding about thermodynamic and effect of structures for understand them [9].

2.1.2 Existing RNA design tools

Nowadays, the synthetic biology is interested in development of the new functions of organism. It is improvement or mutation of DNAs, RNAs or proteins. It needs to design before but the design is difficult and complex. Thus, the computers are applied to design for convenience. For RNA synthesis, there are many existing software for prediction of RNA structure. But there are a few well-known software for structure prediction, which are Dynalign, MFold, NUPACK, RNAsoft, RNAstructure and Vienna package [15].

For Vienna RNA package, it is the most popular to use for development of other algorithms. It is a program from C code for prediction and comparison of RNA secondary structures by Ivo Hofacker from University of Vienna, Austria. For prediction of RNA secondary structure, this package predicts by consider the minimal energy that contains with three kinds of dynamic programing algorithms. They include (i) the minimum free energy algorithm of Zuker and Stiegler (1981) for optimization of a single structure, (ii) the partition function algorithm of McCaskill (1990) for calculation of base pairs by thermodynamic and (iii) the suboptimal folding algorithm of Wuchty et.al (1999) for formation the structure by energy range of the optimal energy. For comparison of secondary structure, this package consists of many distances (detector of dissimilarity) either string alignment or tree-editing of Shapiro and Zhang (1990) [16].

Furthermore, the algorithm for designing RNA is successively developed, such as the RNA Secondary Structure Designer or RNA-SSD software. It is developed by Andronescu et al (2004). This tool is based on stochastic search for solving hard combinatorial problems. It can predict the structure of biological sequences. From the literature, the RNA-SSD performs better than the Vienna RNA package because it can solve structure consistently, but RNAinverse from the Vienna RNA package cannot find the solutions. Now the RNA-SSD is available under the name of RNA Designer in RNAsoft [1].

Taneda (2010) develop the MODENA (Multi-Objective DEsign of Nucleic Acids) that is a multi-objective genetic algorithm for RNA inverse folding. The RNA inverse folding is a methodology for exploring of the RNA sequence folding by target structure that user defines. The MODENA optimizes the solution by two objective functions, which are structure stability score and structure similarity score. And their results are better than the results from other RNA inverse folding program [17].

Rodrigo et al (2013) develop the full design automation of multi-state RNA devices for small RNA (sRNA). It designs by optimization the solution with an objective functions that are stability of transition and intermolecular state. The objective function relates with measured riboregulatory activity. Thus, this tool is the design of molecular interaction mechanism [18].

2.2 Genetic Algorithm

Genetic algorithm or GA is a search heuristic that mimics the natural evolution. It is a stochastic iteration algorithm for optimization. It can solve the optimal solution by substitution the existing solution in chromosome form. And then, the results in each set (called individual) are improved by methods that involve the evolutionary operation. The evolutionary operation is the stochastic changing, which is crossover and mutation. They are used to find the fitness solution by start with the populations that come from randomization. In each generation, many sets of results will be selected to change randomly by crossover or mutation. This operation will generate the new generation that has the more fitness values. This operation will run until the results that have desired fitness values are found. The genetic algorithm requires two things which are:

i. A genetic representation of the solution.
ii. A fitness function for evaluation of solution.

Generally, a genetic representation uses array of bits method that is a set data structure. But it can use other methods depending on the nature of the problems. For fitness function, genes are used to calculate for finding the quality of those genes. And the quality of gene will be used to find the fitness in next generation. The steps of genetic algorithm are shown in below [19].

i. **Initialization of genetic algorithm**: this is the first step of GA. This step will generate randomly the solution (initial population). The population size depends on the nature of the problem. Generally, it contains hundreds to thousands of possible solutions. And this step can randomize significantly for accuracy, but it also depends on the nature of the problems.
ii. **Selection**: during each generation, genes that have high fitness are selected to generate a new generation for approaching to the solution. This step select genes by consider the fitness-base. The fitness-base is the quality of gene calculated by the fitness-function. The fitness-function is different in each problem.
iii. **Genetic operation**: it consists of two methods, which are crossover and mutation. After solutions that have high fitness are selected from the selection step. Those solutions will be paired to generate a new generation by crossover and mutation. In theory, the average quality of solution in new generation will be better than old generation. Thus, the solutions of new generation from this method are different and in high quality, when compared with old generation.
    For crossover, it is a combination of two parent solutions to generate new children solutions. It is the important algorithm because all solutions have favorable part and unfavorable part. Thus, after stochastic crossover, the new solution can be either better or worse [5].
    For mutation, this step is the random changing of some genes of solution. It is used to avoid premature convergence of genetic and maintains genetic diversity in population [5].

iv. **Termination**: the genetic algorithm runs repeatedly until the termination conditions are met. Generally, termination conditions are;
- A solution is higher than criteria
- The number of generation is reached
- Allocated budget (computation time, money) is reached
- A solution reaches to the highest ranking
- Manual inspection
- Combinations of the above

2.3 Thermodynamics of RNA-RNA interaction

The interaction of RNA molecules constitutes the bonding between bases on RNA strands. In general, chemical bonding involves energy. Thus, for RNA-RNA interaction, the energy associated with the interaction is an important parameter for consideration about suitability, stability and possibility. For the thermodynamics of RNA-RNA interaction, it can be understood as the summation of two energy sources, which are (i) the energy needed for exposure of binding site for an interaction and (ii) the energy that gained from interaction as shown in equation (1) [20].

$$\Delta G = \Delta G_u + \Delta G_h \tag{1}$$

Where ΔG_{u} is the free energy that needed for exposure of binding site in the appropriate conformation.
ΔG_{h} is the energy that gained from hybridization at binding site.

For ΔG_{u}, the probability or $P_u[i,j]$ is a sequence interval $[i, j]$ that unpaired. It is calculated by consideration of all possible conformation in region $[i, j]$ that contain in unpaired state. It is used in equation (2), the energy that needed for exposure of binding site for an interaction.

$$\Delta G_u = (-RT)\ln P_u[i, j] \tag{2}$$

An interval is free of secondary structure if it is the exterior loop (it is not enclose by base pairs) or it is enclose by base pair (p,q). These set of all possible conformation does not involve in any base pairs as shown in Figure 2.5.

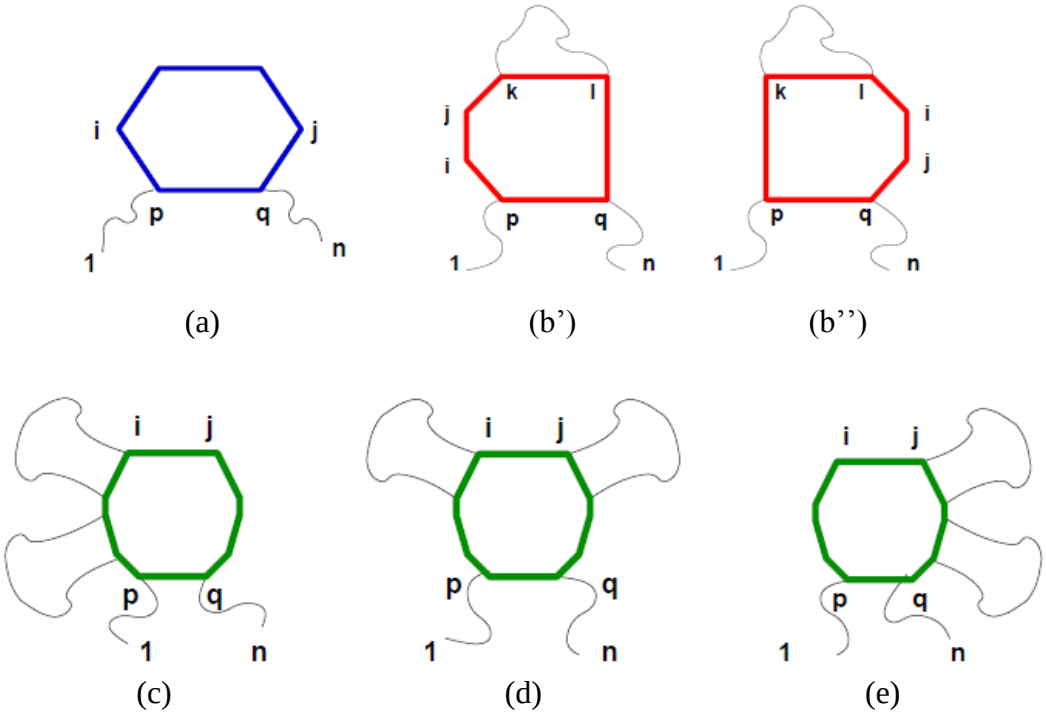

Figure 2.5 A base pair and types of loop [20]

The loop type differences have to be considered. a) A hairpin loop is shown in blue. b) An interior loop that shown in red, two independent contributions are possible. The unstructured region [*i, j*] can be located on either side of the stacked pairs (p, q) or (k, l). c) The region [*i, j*] is contained with multi-loop, we have to account for three different conformations, indicated in the green structures.

The probability depends on the exterior loop and enclosed loop, which can be calculated follow equation (3).

$$P_u[i,j] = \frac{Z[1,i-1]\times 1\times Z[j+1,N]}{Z} + \sum_{p,q} P_{pq} \times \frac{Z_{pq}[i,j]}{Z^b[p,q]} \tag{3}$$

The first term is the ratio of the partition function of all structure from 5' to 3' of [*i, j*] and it represents the exterior loop. And the second term represents the enclosed loop. Z_{pq} [*i, j*] is the partition function of all structure in sequences [p, q], while $Z^b[p,q]$ is number of pairs in sequence [*p, q*]. And the P_{pq} is probability fraction of structure [*i, j*] that is left unpaired [20].

For ΔG_h, the energy gained from hybridization is the energy of partition function of all structure that interact between short RNA and target RNA as shown in Figure 2.6.

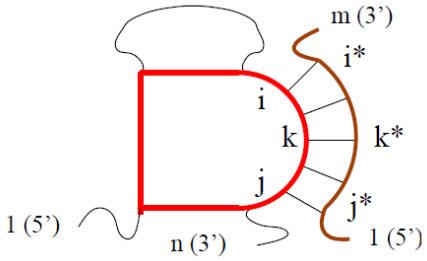

Figure 2.6 Interaction between a short RNA and target RNA [20]

Ulrike et al (2006) shows that $P_u[i,j]$ is an important factor of local target site accessibility for RNA interpolation. And it can be calculated the binding free energy of short RNA to long RNA [20].

2.4 Hamming Distance (HD)

Hamming Distance or HD is a measure index of difference between two strings. For a pair of strings (sequence of characters) that are equal in length, the hamming distance is referred to the total number of positions, at which the characters between the pair are different. Or it measures the minimal number of difference for two strings that compared as shown in below [21].

Target : A U C G A G A U U C

Design : A G C C A G U U U G

Hamming distance : 0 1 0 1 0 0 1 0 0 1 = 4

An algorithm for calculation of hamming distance is divided into three steps. For the first step, the two strings are ensured that the equal length. Because hamming distance can be calculated only two strings that equal length. From the example, the length of strings is equal to 10 characters. Second step, the first character of each strings are compared. If they are the same, "0" is recorded for that character. If they are different, "1" is recorded for that character. And it is repeated for each character in the strings. For this example, the results from recoding are 0101001001. And the last step is summation of the results. The hamming distance is equal to four [22].

2.5 Sorting

For GA, the selection of the best individuals for being parents is importance step to optimize the result. A good selected individual can generate a good new generation. Before selection, data will be sorted for convenient of choosing. It makes program realize that which individual is better than. Therefore sorting before selection is

necessary. The well-known type of sorting has many types that are shown in Table 3.1 (when N = number of data).

Table 2.1 Algorithm run time of 7 sorting types [23]

Name	Number of round in best case	Average number of round	Number of round in worst case	Stability
Binary sort	N	N log N	N log N	Yes
Merge sort	N log N	N log N	N log N	Yes
Heap sort	N log N	N log N	N log N	No
Quicksort	N log N	N log N	N^2	No
Bubble sort	N	N^2	N^2	Yes
Insertion sort	N	N^2	N^2	Yes
Selection sort	N^2	N^2	N^2	No

When consider on average number of round and number of round in worst case, binary sort, merge sort and heap sort are the best but heap sort is not stable. Therefore the best of sorting when they have high number of data is binary sort and merge sort when they have low number of data.

CHAPTER 3 METHODOLOGY

3.1 Overall Methodology

The goal of my thesis is to develop an automatic RNA design tool for designing an RNA-RNA interaction molecule using a genetic algorithm. The developed tool is tested to check its applicability to synthetic biology through case studies. The procedure is divided into 3 main steps as shown in Figure 3.1.

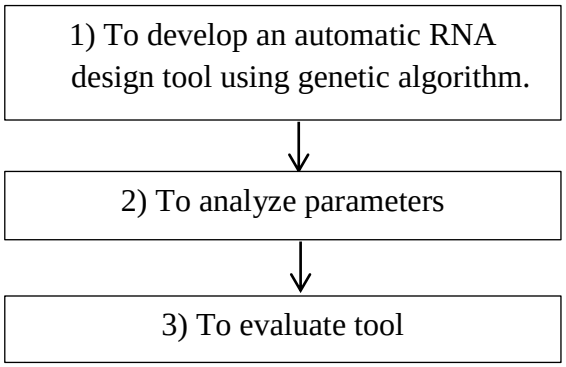

Figure 3.1 The overall methodology for developing an automatic RNA design tool using genetic algorithm

First, the existing RNA design tool is developed for improving the design tool performance using the genetic algorithm (GA). Since the prior design tool and additional tools are C-based program, the GA which is an algorithm for an RNA sequence optimization is also implemented in C programming language. A combination method between Interacting Domain-based Decomposition (*iDoDe*) and DD program [2] in the existing RNA design tool is used to randomly generate initial RNA sequences in an interacting domain (ID) format. These initial RNA sequences are optimized to have properties as the user requirement. The method of the design tool development is described in section 3.2.

Second, the parameters corresponding to an RNA sequence optimization by GA are observed and analyzed their function related to the designed RNAs properties. The optimal parameters that can design the efficient RNA molecules are selected to be used in the developed RNA design tool. The method of parameters analysis is included in section 3.3.

Last, the designing performance of the developing tool is evaluated by RNA test sets. The RNA test sets comprise of artificial RNA sets and synthetic RNA sets [24]. Physical structures and minimal free energy (MFE) of designed RNAs is quantified by a similarity score and a stability score, respectively, that are used to be criteria for evaluation. For the stability score, this thesis performs the MFE of hybridized RNA molecules to represent a stability of RNA-RNA interaction. MFE of designed RNA from developed RNA design tool aims to be less than designed RNA of the prior RNA

design tool [2]. For the similarity score includes the relative distance between the target structures and designed RNA structures. Hamming distance (HD) is used to represent a similarity distance of each designed RNA structures. The evaluation method and calculation is described in section 3.4.

3.2 Development of an automatic RNA design tool using Genetic Algorithm

In this study, I improved the existing RNA design tool [2] by the adding a well-known optimization method, the genetic algorithm (GA) to optimize designed RNA sequences for improving the designed RNA structures and energy. The overview of the developed automatic RNA design tool is presented in Figure 3.2. The processes of the previous tool are depicted in a box with solid-line, while the additional processes of the GA are represented in a box with dash-line.

The tool initially receives the user's requirement and sends a proper information to DD program for generating an initial RNA sequences of ID called "Parent". The generated RNA sequences of IDs are converted to the RNA strands. All initial RNAs of parents including single and hybridized RNA strands are calculated the MFE and predicted the secondary structure by Vienna RNA package [25]. Then, the tool will calculate the HD of the designed structures by comparing to the target structures. Both of MFE and HD are converted to stability score and similarity score, respectively. These two scores are used to calculate a fitness score. This score is used for sorting and selection the parent pairs. These pairs are optimized sequences by genetic operation for generating the new sequences called "Child". They are also calculated MFE and predicted the secondary structure by Vienna RNA package. These scores are used to calculate the fitness score for the evaluation and the reproduction procedures. This tool will run repeatedly until meet termination criteria. Details of each step are described below.

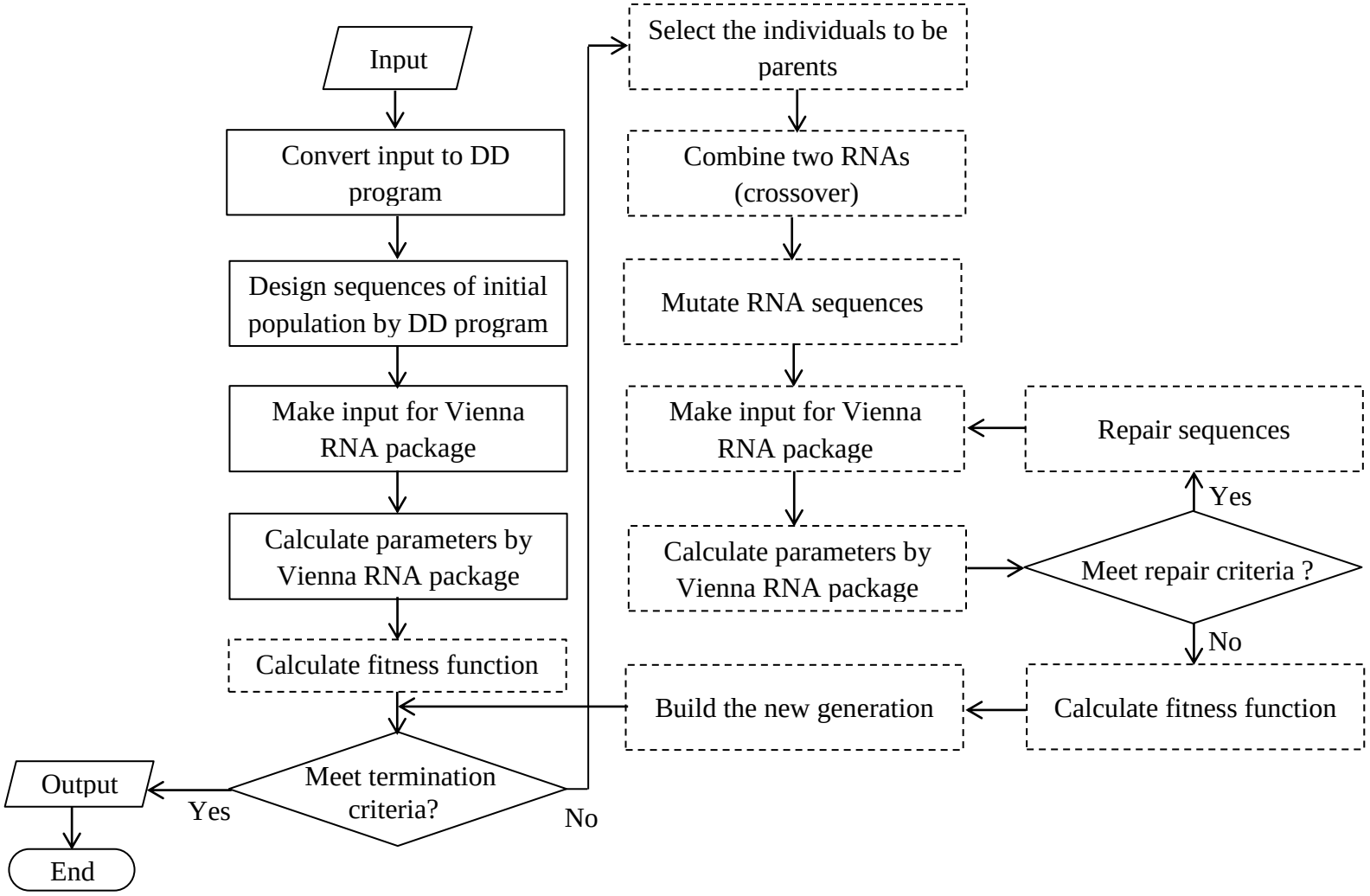

Figure 3.2 Algorithm of a RNA design tool with GA (represents the steps in the previous study and represents the new steps added in this study.

3.2.1 Initialization

In this step, I generated initial RNA sequences by the prior RNA design tool in a random manner. The tool takes the user's requirement consisting of four parts which are (i) specification of RNA-RNA interaction, (ii) sequence constraints, (iii) domain-based decomposition, and (iv) target structures. They are prepared in a format as shown in Figure 3.3 and called 'input file'.

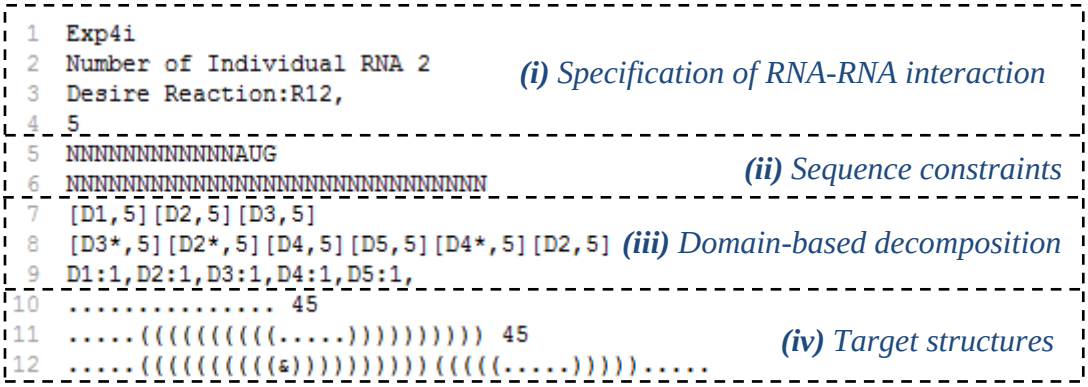

Figure 3.3 The example of input file composition

In a part of specification of RNA-RNA interaction, it consists of experiment name, number of single-stranded RNA, number of reaction and number of domain. For the meaning of domain, it is explained in the third part.

For the second part, Sequence constraints, a user can fix bases and those positions of single stranded RNA by specifying either the upper case letter A, U, G or C to desired locations on the sequences. But the unspecified locations (the bases to be designed by the tool) are written with the letter 'N'. This option provides for specific function assignment e.g. start codon, the stop codon, or the other that affect to function of RNA.

For the next part, interacting domain-based decomposition (*iDoDe*), domain is a set of bases that is a subset of an RNA strand. Each domain is divided for convenience in the design because the program can match the domains whose bases can hybridize with one another. In this part, user can define length of each domain and the reverse complementary domains by putting a sign "*". The reverse complementary domain is the domain that can pair with the same domain number without "*". For the example in Figure 3.3, D2, D3 and D4 have their reverse complementary domains which are equal to 5 bases.

In the last part, the user can define the desired structure in the dot-bracket notation. An unpaired base is represented as dot "." and a base pair is represented as a pair of opening bracket "(" and closing bracket ")" (Figure 3.4).

Figure 3.4 The example of target structures in a) 2D representation and b) dot-bracket notation [26].

The DD program receives the ID input and then randomly designs base sequences of each domain. These random RNA sequences are converted to RNA strands according to an initial position before the iDoDe step, before it is sent to the Vienna RNA package. These results are used as the initial population in GA. This step is run repeatedly until the number of RNA sets is equal to the number of desired RNA sets for using in the GA.

3.2.2 Fitness function calculation

To evaluate the designed RNA properties, the MFE and the secondary structures of designed RNA are used to calculate the fitness function. Both of MFE and secondary structures are taken from Vienna RNA package and subsequently calculated stability score and similarity score, respectively. The fitness function is the equation of combination between stability score and similarity score.

Stability score is calculated from MFE of hybridized strands by equation (4). This equation is MFE proportion of designed hybridized strands (double stranded) to the theory MFE of target RNA.

$$\text{Stability score} = \frac{\text{MFE of designed dsRNA}}{\text{Theoretical MFE of target dsRNA}} \tag{4}$$

The theoretical MFE can be calculated by equation (5). It depends on GC-content. GC-content is a percentage of the number of Guanine and Cytosine in RNA strand. This tool fixes the GC-content at 50% because this value is generally the GC-content of *E. coli*.

$$\text{Theoretical MFE} = (MFE_{GC100} \times \frac{GC-content}{100}) + (MFE_{GC0} \times (1 - \frac{GC-content}{100})) \tag{5}$$

Where MFE_{GC100} = Minimum Gibbs' free energy when GC-content equal to 100.

MFE_{GC0} = Minimum Gibbs' free energy when GC-content equal to 0.

GC-content = Percent of number of Guanine and Cytosine in RNA strand.

MFE_{GC100} and MFE_{GC0} are the minimum Gibbs' free energy calculated by equations (6) and (7), respectively. They depend on the number of base pairs of target structure.

Equations (6) and (7) are linear equations from the experiments by varying the number of base pairs.

$$MFE_{GC100} = -3.30\,(bp) + 7.40 \quad (6)$$

$$MFE_{GC0} = -0.89\,(bp) + 5.02 \quad (7)$$

Where bp = Number of base pairs of target structure.

For similarity score, it is calculated from HD by equation (8). This equation is an average value of the score in each RNA strand, both single strands and hybridized strands.

$$\text{Similarity score} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{L_i - HD_i}{L_i}\right) \quad (8)$$

Where n = Number of RNA strands (both single strands and hybridized strands).

L = Length of each RNA strand (number of base).

HD = Hamming distance.

The maximum scores of the stability score and similarity score are equal to 1. They can be summed to the fitness score by using weight for calculation, using equation (9).

$$\text{Fitness score} = (\text{weight} \times \text{SimScore} + (1 - \text{weight}) \times \text{StaScore}) \quad (9)$$

Fitness score is used as an objective function or fitness function during optimization. The maximum fitness score is equal to 1. The main objective of this tool is base sequence that gives the same RNA structure as the target structure completely (or similarity score equal to 1). Thus, the weight of similarity score is one parameter that must be varied to achieve the objective.

3.2.3 Selection of RNA parents pairs

After calculating the fitness scores, the scores are sorted and then selected to be parents for creating a new generation. To select a proper sorting method, there are two methods i.e. binary sorting and merge sorting that show less time-consuming and high stability. Since a number of input data (Individual number) in this study is small, iteration number for sorting between the binary sorting and merge sorting are closely. However, the strategy of the binary sorting is more complex than the marge sorting. Therefore, I selected the merge sorting for this developed tool.

Merge sorting is a method for sorting data by halve. Program will halve a set of score indefinitely until a set has only one score. And then program will combine backward by comparing and choosing the score in each individual to generate a bigger set again. The example is shown in Figure 3.5.

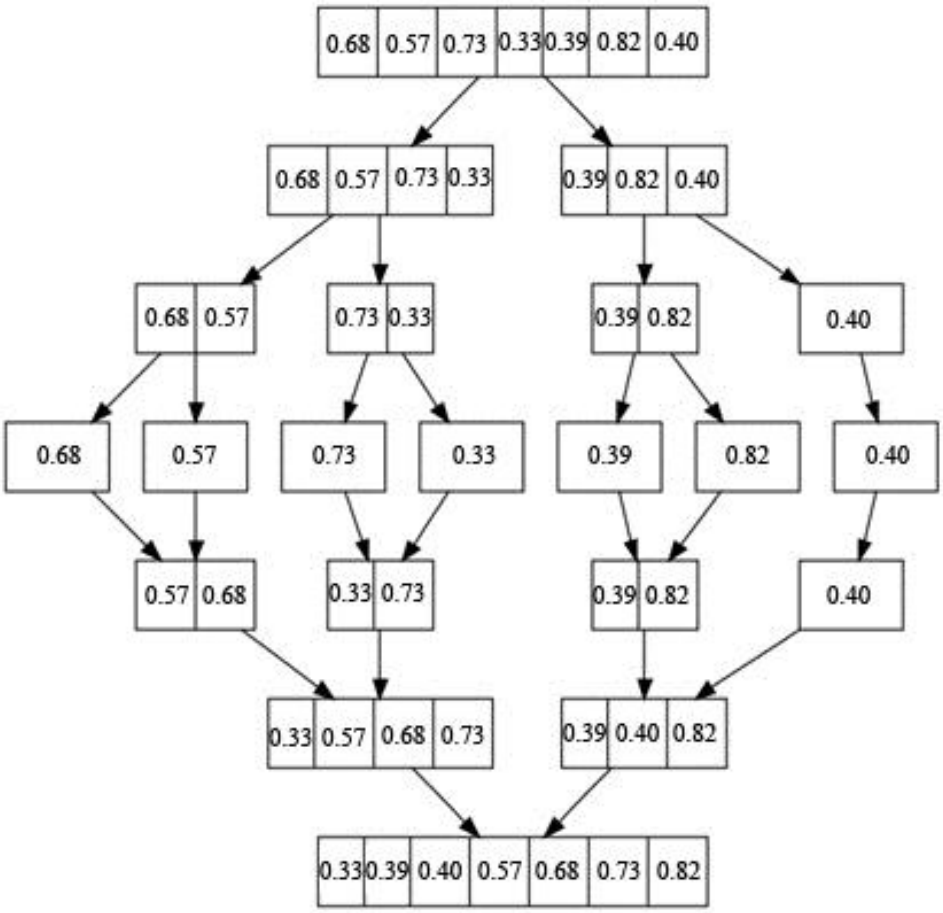

Figure 3.5 Example of merge sort [27]

After scores are sorted, they will be converted to proportion for pairing of parents, using equation (7). The summation of proportion is equal to 1. Therefore, it can be used to criteria for selection of parents by accumulation of proportion. Program will random a number from zero to one. And then, it will be checked for its value. If it is in which range of accumulated proportion, that individual will be used to parents.

$$P_i = \frac{F_i}{\sum_{j=1}^{n} F_j} \tag{7}$$

Where P = Proportion

F = Fitness score

n = Number of individual

For this selection method, individuals are selected by probability. Since, high score has high proportion or wide range. Therefore, an individual that has high fitness score has opportunity to be selected more than those individuals that have low fitness scores. And each individual can be selected repeatedly but cannot be paired with itself.

3.2.4 Crossover

Each RNA set called individual which are selected to be parents for optimizing RNA sequence by genetic operation. To optimize RNA sequence, each ID is performed crossover with same ID of its parent pair. The crossover is the reciprocal recombination between two RNA sequences or individuals will be performed to promote diversity in the population by crossover rate. The crossover rate is the predefined value for the probability of a domain / an individual to be selected for crossover, for each sequence, a random number is generated and compared with the crossover rate to determine whether this sequence is selected for crossover; if the random number is lower than the crossover rate, the sequence is selected. The example of crossover is shown below. This tool randomizes the position for crossover. Therefore, the same pair of parents is not necessary to get the same children. For the algorithm of this step, it is shown in appendix B.

	Domain 1	Domain 2	Domain 3
Parent1	AUGAGAUGG	CCGAUA	UGCA
Parent2	GUACCAUAG	AUCGUA	UAGC
Child1	AUGACAUAG	CCGAUA	UGCC
Child2	GUACGAUGG	AUCGUA	UAGA

Since the crossover is the genetic operation in a fragment of RNA, the change of RNA sequence is thus awide exploration for this tool. The parameter of this step should be analyzed combined with the mutation step.

3.2.5 Mutation

The mutation step is a step that generates a new sequence by changing some bases in a sequence to promote diversity again with probability or mutation rate. Generally, the mutation rate is lower than the crossover rate because this step wants to change a few bases (crossover step changes a range of bases). The mutation is lower diversity than the crossover. If the random number of each is lower than the mutation rate, that base will mutate, which is the same as in the crossover step. The example of the mutation is shown in below.

	Domain 1	Domain 2	Domain 3
Before	GUACGAUGG	AUCGUA	UAGA
After	GUACGAAGU	AUGGUA	UCGA

The mutation step is the exploitation. It is used for increasing the efficiency of tool. For complicated and large problems, exploration and exploitation should be kept balance.

Hence, the crossover rate and the mutation rate must be varied for finding their appropriate values.

3.2.6 Repair

This step is a special function for changing the bases in positions that affect to the wrong structures. It will be used in some generations, when the program cannot generate a new sequence that has a higher score than existing solutions.

This function starts by finding the position that affects to the wrong structure by using the Hamming distance. Those positions are positions that should not be paired in the target structure, but they pair. Therefore this program will find them and their base pair and change only one base to another base that cannot pair with the old base pair.

3.3 Parameters analysis

Since there are many parameters in this tool which can affect the speed and efficiency of RNA design procedure. The parameters change may or may not affect the efficiency. Therefore, parameter analysis is an important step for finding optimal parameters of RNA design. Trends of the results can explain the effect of each parameter to desired results and can help choose the appropriate values of each parameter.

Aforementioned, the crossover rate, the mutation rate, and the weight of similarity score are important parameters. The crossover rate and the mutation rate are exploration and exploitation, respectively, of sequence optimization. The tool performance causes the relation of them. Therefore, these two rates are analyzed together. For the weight of similarity score, it is a determinant of optimization. Because the objective function or the fitness function depends on it. The appropriate value of the weight will get the results that meet with the objective. Furthermore, after this tool has the appropriate values of parameters, the number of iteration can be defined to appropriate value.

3.4 Evaluation of RNA design tool performance

After the parameter analysis step, this tool is tested and compared the results with the prior tool [26]. The fitness sore, the similarity score, the stability score and the number of iteration are used to evaluate the performance of this tool.

The RNA test sets consist of 2 groups: 1) artificial RNA test sets and 2) an existing synthetic RNA set. The artificial RNA test sets comprise four RNA models which are two single strands and one hybridize strand (Figure 3.6).

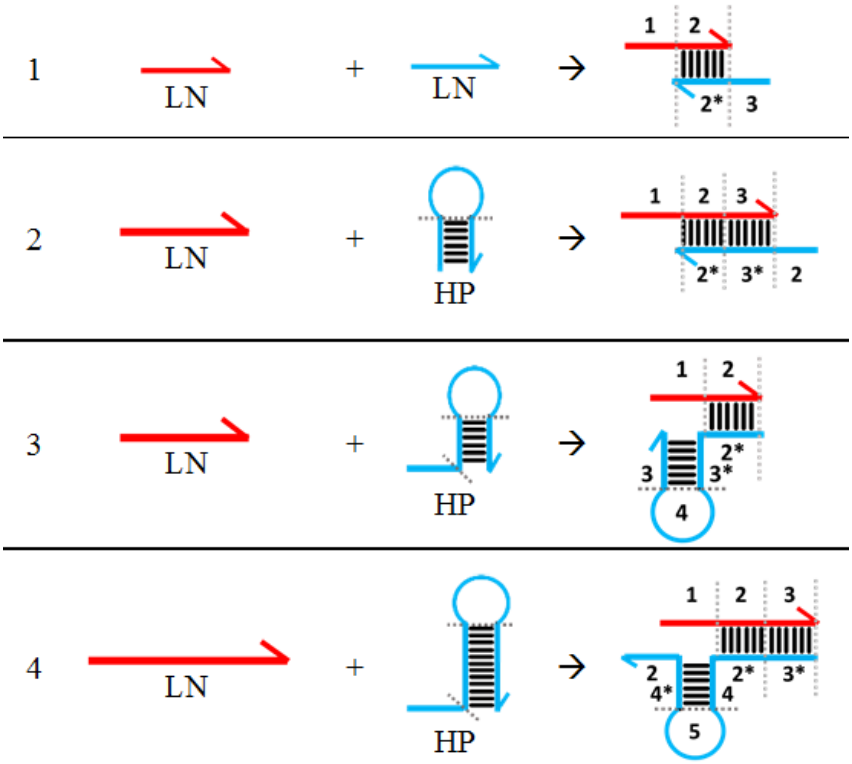

Figure 3.6 Structures of artificial RNA test sets [26]

The first model is an interaction between two linear strands which is a basic structure of double-stranded RNA. For the second model, there are hybridization between a linear strand and hairpin-loop without hanging region. This model is used to test ability of the complete hairpin-loop destruction. For models 3 and 4, they are an interaction a linear strand and hairpin-loop with hanging region. But model 3 represents the incomplete hairpin-loop breaking down, model 4 represents the partial hairpin-loop breaking down.

A more complex model of Isaacs et al (2004) is used as an existing synthetic RNA to check whether the tool can be used to design a more complex RNA structure.

CHAPTER 4 RESULTS AND DISCUSSION

4.1 Parameters analysis

To find a good setting parameter values that work well on a RNA design problem, different values of the three parameters: the crossover rate, the mutation rate and the weight of similarity score were used to design the target RNA model. I used the synthetic RNA structure of Isaacs et al (2004) as a target model of design. The model of Isaacs et al. (2004) composes of two single stranded RNAs (R1 and R2) and one hybridized RNA (R12) as shown in Figure 4.1.

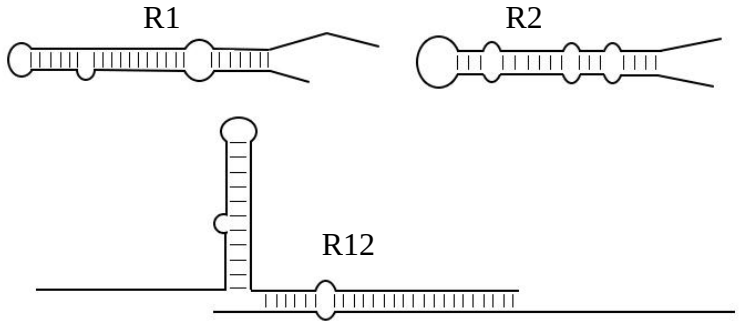

Figure 4.1 The structures of synthetic RNA model for GA parameters analysis (Modified from [24])

The effect of the GA parameters on the RNA design tool performances was determined using three scores of the designed RNA: fitness score, similarity score and stability score. Also, the data of designed RNAs were collected 100 generations in all analysis. The results of different parameters analysis is described in the following sections.

4.1.1 Weight of similarity score analysis

Since the goal of the developed tool is to design RNA sequences that can adopt totally the target structure. In this study, the fitness score which is a function of the similarity score and stability score was used to be an objective function of the RNA design. In general, the objective function is a critical determinant for a best individual selection that is involved in the optimization process. To obtain the appropriate objective function or the fitness function related to the goal of study, the weight of similarity score on the fitness function was tested at different values for observing their effects on the optimization.

As the goal of the developed tool considers the structure similarity of RNA more than the structure stability, the weight of similarity score was set from 0.5 to 1.0. When the crossover rate and the mutation were equal to 0.75 and 0.25, respectively. Also, the number of generation and the number of individual were the same as the above analysis that is 100 generations and 10 individuals, respectively. The median lines of all three scores are represented in Figure 4.2.

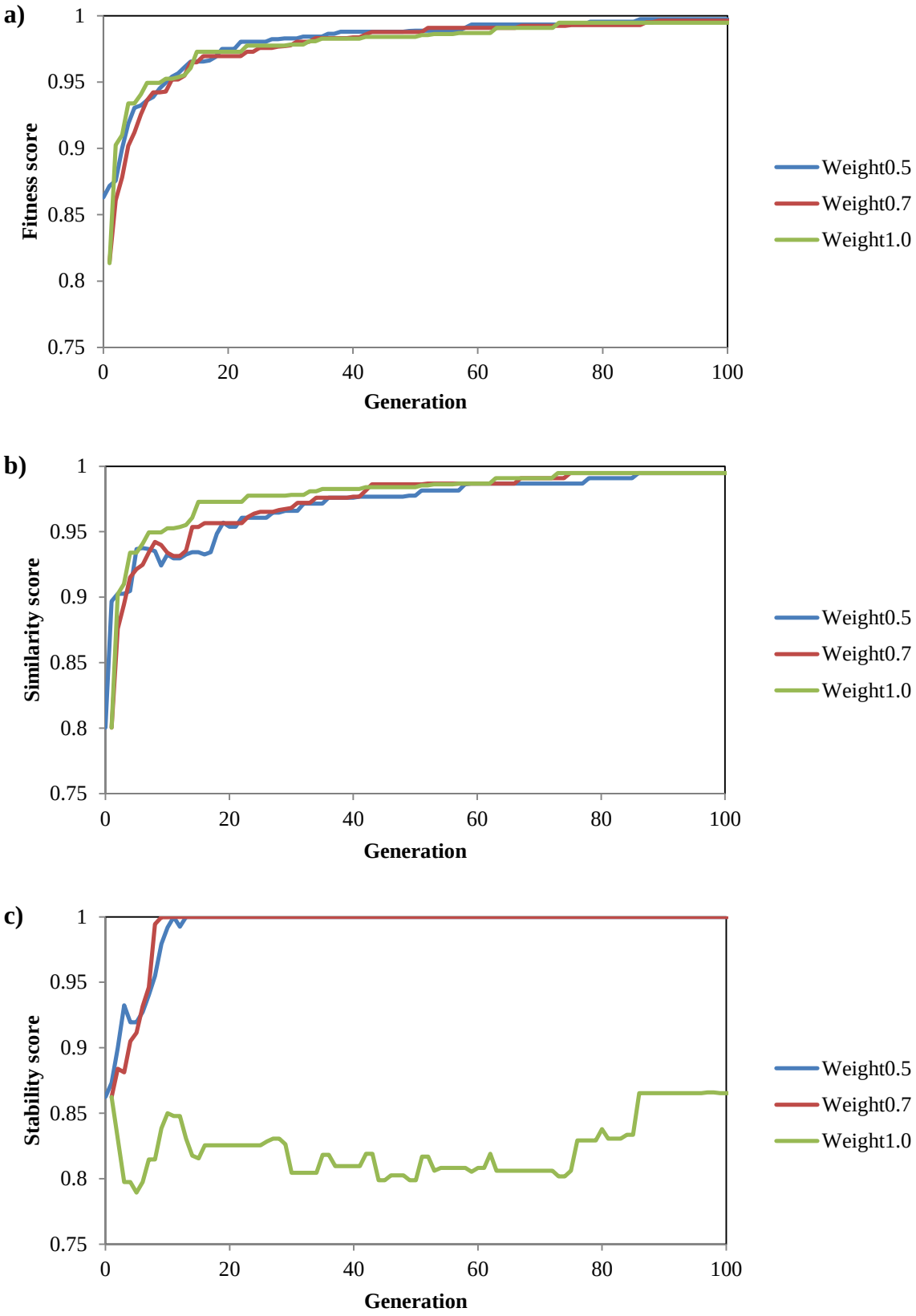

Figure 4.2 The median of 3 scores obtained from 100 generations of the RNA design at the different weights of similarity score; a) fitness score, b) similarity score and c) stability score.

Figure 4.2 (a) indicates that the increase of this parameter retarded the highest score achievement. During the simulation, I found that the similarity score is always lower than the stability score. With this reason, the fitness score with the high weight showed gradually decreasing the score improvement. However, only the result of the fitness score is not adequate for selecting the appropriate weight of the similarity score. Thus, the behaviour of the similarity score and the stability score at the different parameter values is also analyzed.

From Figure 4.2 (c), , the trend of the stability scores at the weight equal to1 do not increase. Because, at the weight equal to 1, the stability score is not calculated and determined for optimization. Therefore that value is not inceased. Since the stability of structure is one objective of this tool. Hence this value is not appropriate to be used in this tool.

When considering Figures 4.2 (b) and (c), it was found that when the similarity score increases gradually, the stability score will increases steeply. Therefore, when considering on the weight that are equal to 0.5 and 0.7, the appropriate value of the weight equal to 0.7. Because, at this value, the similarity score and the stability score are balanced.

4.1.2 Crossover rate analysis

The crossover rate is a parameter that controls the frequency of crossover operator. Therefore different values of this parameter can affect the RNA sequence improvement of new generation. If it is too high, high-performance individuals are discarded faster than selection can produce improvements. For this trial, the crossover rate was tested at 0.1, 0.3, 0.5, 0.7, and 0.9. For the other parameters, the mutation rate and the weight of similarity score were fixed at 0.25 and 0.7, respectively. The developed tool was run for 100 generations (100 iterations) and each generation was composed of 10 individuals (set of sequence). Three scores obtained during 100 generations were represented by a line of the median that is shown in Figure 4.3.

Figure 4.3 shows that the crossover rate at 0.9 gave the highest value among all 3 scores. It indicates that the developed tool can be achieved to a steady state quickly with the highest score by the high crossover rate. In contrast, a median line of the crossover rate at 0.1 was reached a steady state slowly. Meanwhile, the crossover rate from 0.3 to 0.7 produced a random pattern, and the late state of all scores laid on the same value. However, at the crossover rate equal to 0.3 to 0.7, the trends of the results are close, possibly because the crossover rate is only a criterion of probability. Therefore the results from each trial also depend on the value of random number that checked with the criterion. Each time, the random numbers are different. Moreover, the positions for the crossover are also different in each time.

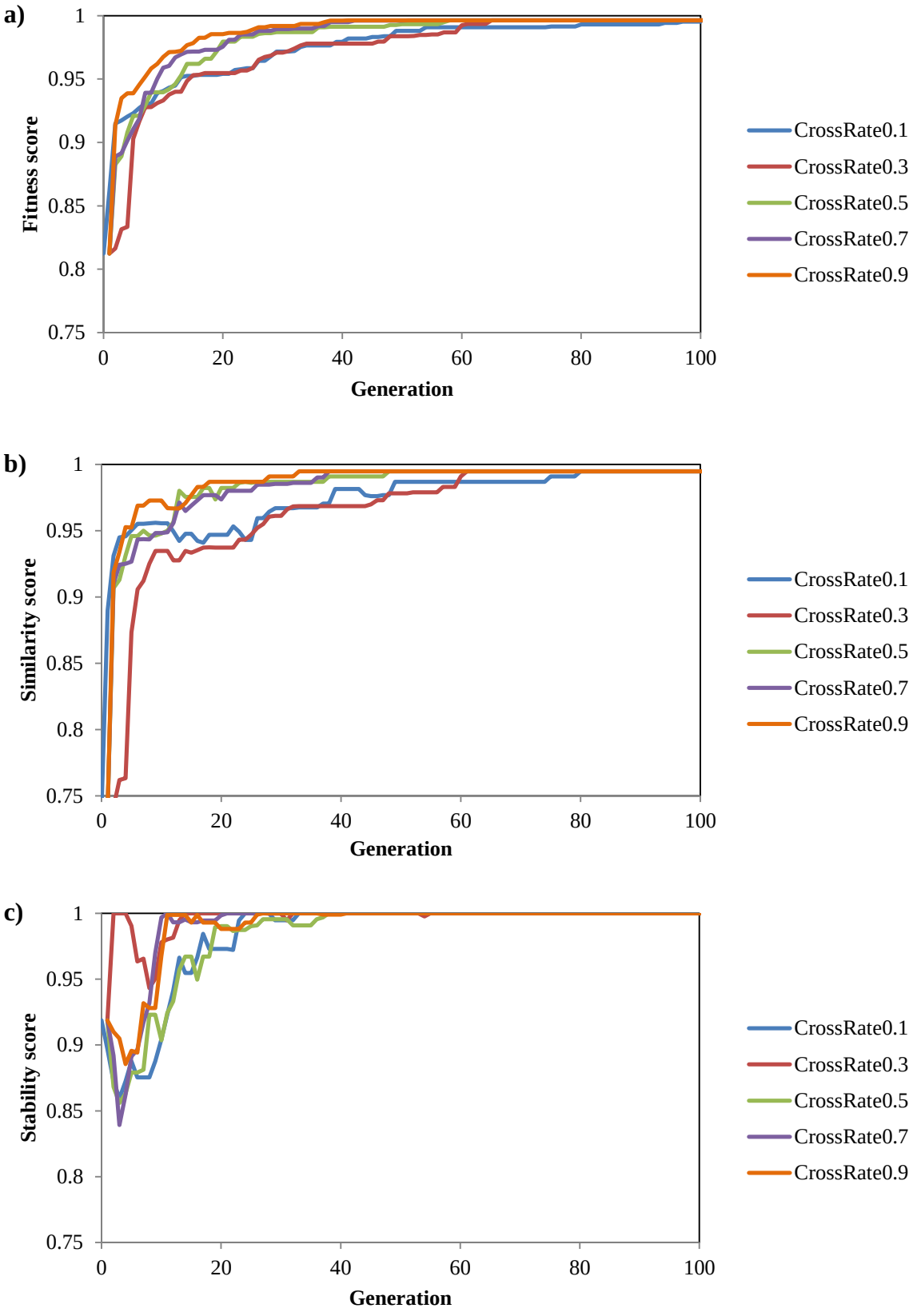

Figure 4.3 The median of 3 scores obtained from 100 generations of the RNA design at the different crossover rates; a) fitness score, b) similarity score and c) stability score.

As described above, these results depend on many factors, not only the crossover rate, but also the random number in each time and the position for the crossover. Although it is difficult to find what value is clearly proper to be used in GA, it seems that the highest crossover rate at 0.9 achieved the steady state faster than the other crossover rates. For this analysis, the crossover rate of 0.9 was selected as an appropriate value for the further optimization.

4.1.3 Mutation rate analysis

Mutation is another operator for improving a new individual by changing arbitrary base of each domain. The cause of too high mutation rate will lead to a number of changed bases. Therefore, the mutation of a designed RNA should not be too high because it will become a random search of RNA sequence. To find an appropriate value for RNA design problem, the effect of the mutation rate was observed at various values on the same three scores.

As described above, the three scores of designed RNAs were investigated at various mutation rates; 0.1, 0.2, 0.3, 0.4 and 0.5. Whereas, the crossover rate and the weight of similarity score ware set to 0.9 and 0.7, respectively. The designed RNA by different mutation rates were collected at 100 generations and 10 individuals per generation. The median values of all three scores were plotted and shown in Figure 4.4.

Figure 4.4 shows that the increase in the mutation rate affected to the fitness scores that slowly achieved to a steady state. It seems that a high mutation rate reduces the optimization property of GA since a number of bases were more changed leading to a random search. Therefore, I selected the mutation rate of 0.1 to be the optimal value for the RNA design because it gave the highest scores within early generation in all three scores.

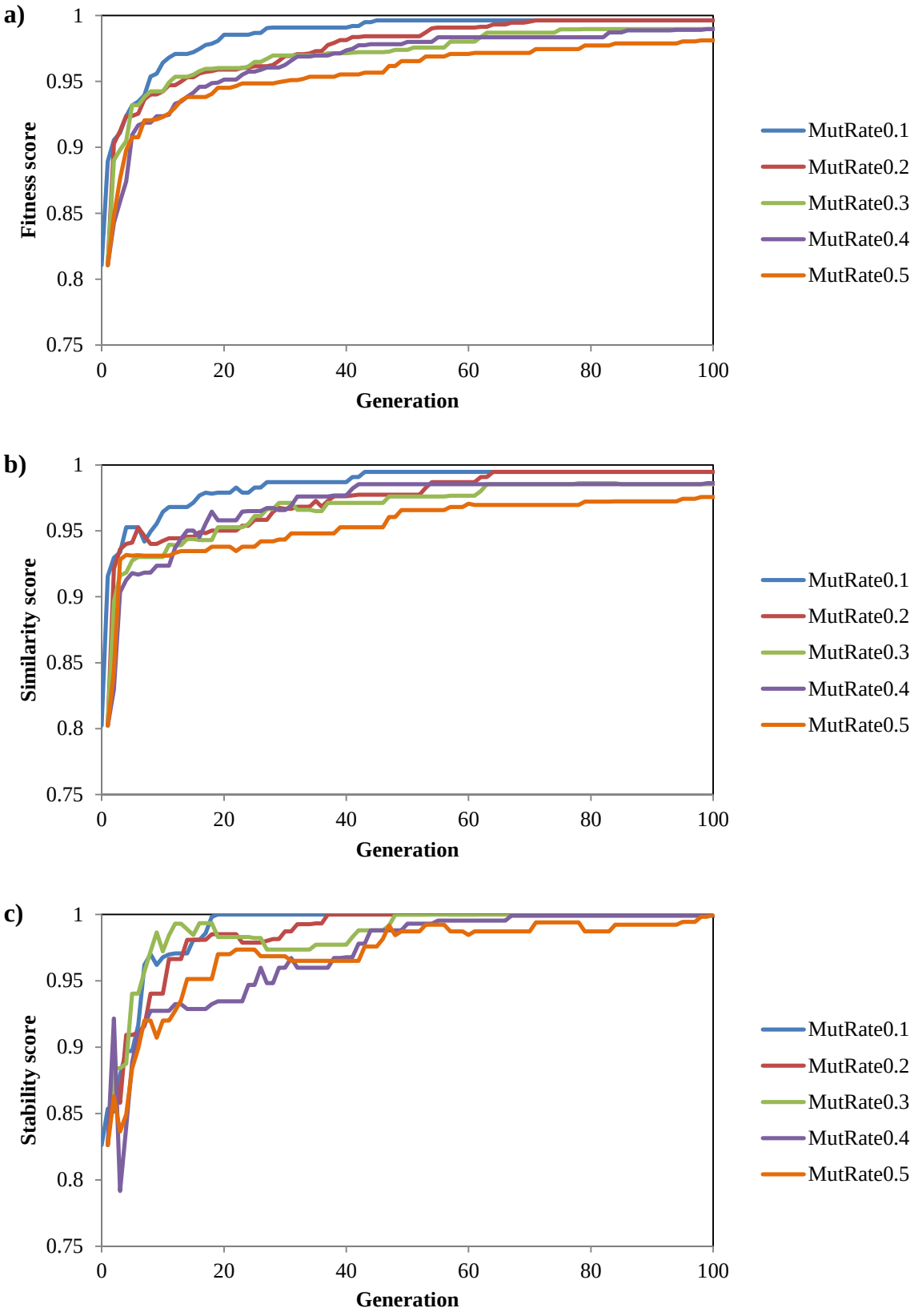

Figure 4.4 The median of 3 scores obtained from 100 generations of the RNA design at the different mutation rates; a) fitness score, b) similarity score and c) stability score.

4.1.4 Population size analysis

In addition to those three variables, there is another variable that affects the speed of the tool. It is a population size or the number of individual (solution) in each generation. The number of individual affects the results in a next generation. If it is high, the diversity of solutions in a next generation will be high also. But the high number of individual affects the computer work. Hence it needs to be tested for observing its effect to the computer work.

For this trial, the number of individual was tested at 8, 10, 20, 50 and 100 per generation. The other parameters i.e., the crossover rate, the mutation rate and the weight of similarity score were fixed at 0.9, 0.1 and 0.7, respectively. The program was run until it met the first solution that had a maximum fitness score. The results are shown in Table 4.1

Table 4.1 The speed of the tool in each population size

Population size	Number of Generation	Number of total individuals	Total time (s)
8	42	336	50
10	23	230	36
20	24	480	75
50	12	600	107
100	8	800	158

They are the data of the first generation that obtains a maximum fitness score. The increasing of the population size affected the first generation that met a maximum fitness score. The large population size made this tool run with in a few iterations. But when considering the total time, at the large population size, this tool spent more time. Because, in each generation, this tool spent the time for generating as many solutions as a the population size. Or the total time depended on a number of total individuals that is shown in Table 4.1. Therefore the best population size for this tool is 10 because it spent a minimal time.

4.1.5 Repair analysis

From the above three analysis of the synthetic model, it seems that, this tool still does not achieve the fitness score equal to 1. Their structure had some positions that are different from the target structure. Therefore the repair operation was added to solve this problem. The repair operation was used in every generation for comparing the effect of repair function. It operated only the individual that had the fitness score more than 0.95 due to the performance reason. For this trial, the weight, the crossover rate, the mutation rate and the population size were equal to 0.7, 0.9, 0.1 and 10, respectively. The results of this trial are shown in Figure 4.5.

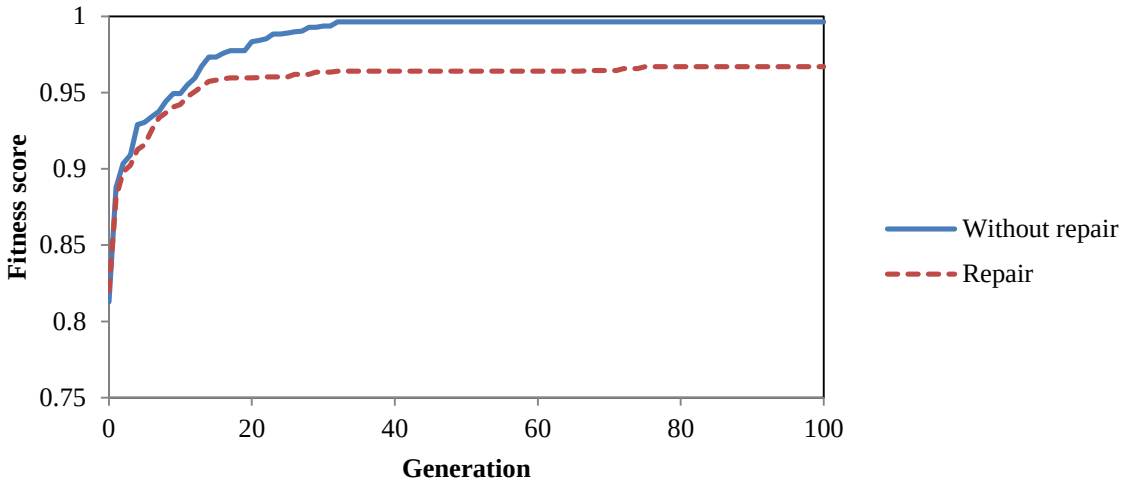

Figure 4.5 The median of fitness score of tool with and without repair function

The increasing of fitness score in each generation of both conditions were quite the same in the beginning. It seems that the repair function could not increase the speed of this tool. And it increases the total time for solving, because it must recheck every individual in each generation. Therefore this function must be used only when necessary.

Moreover, the last fitness score of both conditions were different because the final structures of the design sequence are different. Since when the repair function edited the position that was wrong, the structure predicted from this tool was changed to a new structure that had more wrong positions. But, when the sequence structure from Isaacs et al (2004) was predicted by this tool, that structure was the same as the structure from the repair function (that structure was different from the structure mentioned in the paper). It seems that the tool that is used for predicting the structure was the cause. Therefore, in the part of prediction of the structure, the tool must be validated with laboratory Data before further used.

4.2 Performance evaluation of RNA design tool

To evaluate the performance of the developed tool, 5 artificial RNA models, which are four simple models and one complex model, were used for testing. The simple models used in this step were the same as the artificial RNA test set in the previous study [26]. The illustrations of the simple RNA model are shown in Figure 3.6. The complex model was a synthetic RNA structure of Isaacs et al (2004) shown in Figure 4.1.

For the prior tool, it is a combination method between Interacting Domain-based Decomposition (iDoDe) and DD program [23], which is only a random design without the optimization process. Therefore, the comparison of the results from these two tools

was observed. For the developed tool with GA, the crossover rate, the mutation rate and the weight were set equal to 0.9, 0.1 and 0.7, respectively.

For this trial, the current developed tool (iDoDe + GA) was run first. It was run until this tool met the solution that had a maximum fitness score as same as shown in part 4.1.4. The total run time was recorded and used as the termination criteria in iDoDe. Therefore both tools used the same computational time. The results are shown in Table 4.2.

Table 4.2 The optimal fitness scores and the success rates of iDoDe and the current developed tool

Model	RNA length	Total Time (s)	iDoDe		iDoDe + GA	
			Optimal fitness score	Success rate	Optimal fitness score	Success rate
1	10/10	3	0.947	0/11	1	3/80
2	15/15	2	0.969	0/13	1	3/30
3	10/20	5	0.874	0/19	1	1/130
4	15/30	4	0.936	0/12	1	1/50
5	55/71	37	0.925	0/73	0.996	4/240

For the complexity of structure, it is shown in RNA length form. They are the length of each single RNA strand. When considering the optimal fitness score of both tools, the fitness scores of the current developed tool are higher than those of the prior tool. The current tool can achieve the maximum fitness scores for models 1, 2, 3 and 4 but the prior tool cannot. For the success rate of both tools, the prior tool does not have results that are success, but the current developed tool has (when success is result that has the fitness score more than 0.99). When considering the total number of solution, this tool can generate the higher number of solution than the prior tool, given the same computational time. It appears that this tool can generate the solution faster than the prior tool. Moreover, even when iDoDe was run for 1,000 trials, the maximum fitness score obtained for each model is shown in Table 4.3.

Table 4.3 The optimal fitness score of iDoDe when run in 1,000 trials

Model	Optimal fitness score
1	1.000
2	1.000
3	0.922
4	0.968
5	0.967

Unlike iDoDe plus GA, it seems that iDoDe cannot give the sequence that has maximum fitness score for models 3 and 4 even it was run for 1,000 trials. In other words, iDoDe plus GA can give the solution of structure that iDoDe cannot give. Therefore it can be concluded that iDoDe plus GA can design the better solution than iDoDe.

In conclusion, the tool with GA as the optimizer can generate the RNA sequences that have the maximum fitness score. It can be concluded that the current developed tool can successfully design all four simple models of RNAs. And for the complex model, the current developed tool can generate the RNA sequences that have higher fitness score than the prior tool.

CHAPTER 5 CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

An automatic RNA design tool using genetic algorithm is successfully developed. The tool is necessary to be tested for finding an appropriate value of each parameter and observing its performance. It is divided into two parts, which are parameters analysis and evaluation of RNA design tool performance.

For parameters analysis, the parameters of interest are the weight of similarity score, the crossover rate, the mutation rate and the population size. They affect the accuracy and the speed of the tool. They were tested by using the synthetic RNA model of Isaacs et al (2004) as a test set. These trials were operated on 100 iterations. The weight of similarity score was tested at 0.5, 0.7 and 1 by fixing the crossover rate, the mutation rate and the population size at 0.75, 0.25 and 10, respectively. It was concluded that the weight of similarity score at 0.7 was the most appropriate value. For the crossover rate, it is a parameter that affects to the RNA sequence improvement of new generation. It was tested at 0.1, 0.3, 0.5, 0.7 and 0.9. For these trials, the mutation rate, the weight of similarity score and the population size were fixed at 0.25, 0.7 and 10, respectively. From the results, the crossover rate of 0.9 was selected to be an appropriate value because it reaches the maximum score fastest. For the mutation rate, it was tested at 0.1, 0.2, 0.3, 0.4 and 0.5 by fixing the weight, the crossover rate and the population size at 0.7, 0.9 and 10, respectively. It was concluded that 0.1 was an appropriate value for the mutation rate. And for the last parameter, the population size or the number of individuals, it was tested at 8, 10, 20, 50 and 100, while the other parameters were fixed at the appropriate values; the weight, the crossover rate and the mutation rate were 0.7, 0.9 and 0.1, respectively. The time when the program met the first maximum score was lowest at the population size equal to 10. Hence it was an appropriate value for the population size.

For the performance evaluation of the RNA design tool, this trial was a comparison of the results from this tool with the prior tool (iDoDe) by running with the RNA test sets. The RNA test sets consisted of four artificial models of Thaiprasit, J., et al. (2014) [26] and one synthetic model of Isaacs et al (2004) [24]. It was concluded that the time when the program met the first maximum fitness score of this tool was lower than the prior tool. In addition, this tool can generate more candidate RNA sets with higher success rate than iDoDe considering the same computational time. Thus, we conclude that our developed tool, which incorporates genetic algorithm, has a much better performance than iDoDe.

5.2 Recommendations

- This current version of the tool developed in this work is restricted to the design of 2 single RNA strands with one hybridized strand. Further expansion of the tool should be done such that the tool can design more than two single RNA strands.

- The current tool still fails to give a perfect design (fitness score <1) for complex RNA structures. Thus, it needs further improvement by adding a function that fixes those mismatched bases as well as repeated bases occurred in a given optimal design solution. This should help improve the accuracy of the tool in designing complex RNA structures such as the Issacs model.