Multiobjective Evolutionary Optimization of DNA Sequences for Reliable DNA Computing

Soo-Yong Shin, In-Hee Lee, Dongmin Kim, and Byoung-Tak Zhang

***Abstract*—DNA computing relies on biochemical reactions of DNA molecules and may result in incorrect or undesirable computations. Therefore, much work has focused on designing the DNA sequences to make the molecular computation more reliable. Sequence design involves with a number of heterogeneous and conflicting design criteria and traditional optimization methods may face difficulties. In this paper, we formulate the DNA sequence design as a multiobjective optimization problem and solve it using a constrained multiobjective evolutionary algorithm (EA). The method is implemented into the DNA sequence design system, NACST/Seq, with a suite of sequence-analysis tools to help choose the best solutions among many alternatives. The performance of NACST/Seq is compared with other sequence design methods, and analyzed on a traveling salesman problem solved by bio-lab experiments. Our experimental results show that the evolutionary sequence design by NACST/Seq outperforms in its reliability the existing sequence design techniques such as conventional EAs, simulated annealing, and specialized heuristic methods.**

***Index Terms*—DNA computing, DNA sequence design, multiobjective evolutionary algorithm (MOEA), nucleic acid computing simulation toolkit/sequence generator (NACST/Seq).**

I. Introduction

Deoxyribonucleic acid (DNA) computing is a computational model that uses biomolecules as information storage materials and biological laboratory experiments as information processing operators [1]. The ability of DNA computers to perform calculations using specific biochemical reactions between different DNA strands by Watson–Crick complementary basepairing, affords a number of useful properties such as massive parallelism and a huge memory capacity [2], [3]. Recently, several interesting applications have been demonstrated [4], [5].

However, due to the technological difficulty, DNA reactions may result in incorrect or undesirable computation. Sometimes DNA computing fails to generate identical results for the same problem and algorithm. Also, some DNAs can be wasted performing undesirable reactions. To overcome these drawbacks, much work has focused on improving the reliability (correctness) and efficiency (economy) of DNA computing. Most existing approaches focus on the design of DNA sequences that reduce the possibility for illegal reactions [6]. The emphasis on avoiding undesirable reactions should improve both reliability and efficiency of the generated DNA sequences. Existing techniques for DNA sequence design include genetic algorithms, dynamic programming, and heuristic methods [6].

In this paper, we propose a new evolutionary DNA sequence design method which is implemented in the nucleic acid computing simulation toolkit/sequence generator (NACST/Seq) system. Previous evolutionary approaches used a simple genetic algorithm by taking well-known fitness measures and aggregating them into a fitness function of weighted sum. However, as the number of fitness terms increases, normalization of fitness values is required, and also it becomes difficult to find the proper weight value for each criterion. The present work improves on the previous work by formulating and solving the DNA sequence design problem as a multiobjective optimization task. The sequence design problem has a number of heterogeneous and conflicting design criteria (objectives) that must be satisfied simultaneously. Since the multiobjective evolutionary algorithm (MOEA) approach allows a number of heterogenous design criteria to be defined in a consistent way, the MOEA will be a good candidate for DNA sequence optimization. Though there has been some work in the related fields such as microarray probe design [7], [8], no MOEA has been applied to DNA sequence design for DNA computing. The NACST/Seq system demonstrates a practical application of multiobjective evolutionary optimization to real-world biochemical design problems. We verify the utility of NACST/Seq by comparing it with other sequence design tools. We also evaluate the performance of the proposed method on a traveling salesman problem (TSP) solved by biochemical experiments. The results demonstrate NACST/Seq's effectiveness in improving reliability of DNA sequences.

The paper is organized as follows. In Section II, previous work is reviewed and the DNA sequence design problem is defined. Section III gives an overview of the proposed NACST/Seq system. Section IV describes the MOEA and the analysis results of fitness measures. In Section V, the sequence generation results are shown and compared with those of other existing methods. In Section VI, conclusions are drawn and future studies are discussed.

II. DNA Sequence Design

A. Previous Work

Many studies describe the sequence design problem in terms of threshold-based constraints: Sequences are designed for

Manuscript received November 11, 2003; revised October 15, 2004. This work was supported in part by the Molecular Evolutionary Computing (MEC) Project of the Korean Ministry of Commerce, Industry and Energy, in part by the National Research Laboratory (NRL) Program from the Korean Ministry of Science and Technology, and in part by the BK21-IT Program of the Korean Ministry of Education and Human Resources Development.

The authors are with the Biointelligence Laboratory, School of Computer Science and Engineering, Seoul National University, Seoul 151-742 Korea (e-mail: syshin@bi.snu.ac.kr; ihlee@bi.snu.ac.kr; dmkim@bi.snu.ac.kr; btzhang@bi.snu.ac.kr).

Digital Object Identifier 10.1109/TEVC.2005.844166

TABLE I
SUMMARY OF THE PREVIOUS DNA SEQUENCE DESIGN SYSTEMS. THE RELATED WORKS ARE CATEGORIZED BY THEIR SEQUENCE GENERATION ALGORITHM

Approaches	Design systems
Exhaustive Search	Hartemink *et al.* [9]
Random Search	Penchovsky and Ackermann [10]
Template-map Strategy	Frutos *et al.* [11], Arta and Kobayashi [12]
Graph Method	Feldkamp *et al.* [13]
Stochastic Methods	Tanaka *et al.* [14]
Dynamic Programming	Marathe *et al.* [15]
Biological-inspired Methods	Deaton *et al.* [16], [17], Heitsch *et al.* [18]
Evolutionary Algorithms	Deaton *et al.* [19], Zhang and Shin [20], Arita *et al.* [21], Reben *et al.* [22], Shin *et al.* [23]

certain measures of each pair of sequences to exceed given thresholds. More recently, researchers have attempted to design DNA sequences *in vitro*. We briefly review these approaches here. Table I summarizes the previous sequence design systems categorized by their algorithm. Most previous systems are based on threshold values.

The simplest methods are exhaustive search and random search. Hartemink *et al.* [9] implemented the exhaustive search method "SCAN" to generate sequences for the programmed mutagenesis. Though they successfully designed DNA sequences, it took much computational time. Penchovsky and Ackermann designed DNA sequences by a random search algorithm [10]. They encoded binary information in DNA strands and demonstrated twelve-bit DNA library. Frutos *et al.* [11] and Arita and Kobayashi [12] proposed a template-map strategy to select a huge number of dissimilar sequences by using only a significantly smaller number of templates and maps. They build some predesigned template, and then statistically generate a set of sequences ensuring small mismatch probability. The template method has a merit that it can find a reliable sequence in a short time within a given error rate. Feldkamp *et al.* [13] use a directed graph to design DNA sequences. The nodes in the graph represent base strands and a node has four strands that can appear as successors in a longer sequence as its child nodes. Then, by traveling the graph from root to leaf, DNA sequences can be designed. This approach also can find a set of orthogonal DNA sequences within a predefined error rate quickly. Tanaka *et al.* [14] offered some sequence fitness criteria, and generated the sequence using simulated annealing. They also tried to find proper combinations of the proposed fitness functions to find more promising solutions. Marathe *et al.* [15] used a dynamic programming approach to design DNA sequences based on Hamming distance and free energy. They made theoretical investigations on DNA sequence design using classical coding theory.

Unlike previous systems, biological-inspired methods have been offered recently to design DNA sequences. Deaton *et al.* [16] proposed a PCR-based protocol for *in vitro* selection of DNA sequences. This approach is especially interesting in that it uses *in vitro* evolution to find noncross-hybridizing DNA libraries, while most DNA sequence design methods use *in silico* approaches. Though this method is one of the most reliable method to design DNA sequences, it has inherent difficulties, i.e., it cannot distinguish each DNA sequence in the library. To assign information on DNA sequences, one has to know the specific composition of DNA sequences. Other biological-inspired methods take into account the thermodynamics of DNA structures and free energy of DNA sequences. These features can be used to select reliable DNA sequences as alternative fitness measures [17], [18], [24].

Evolutionary algorithms have also been proposed for sequence design. In particular, simple genetic algorithms are often used [19]–[22]. This seems due to their simplicity and efficiency. Deaton *et al.* proposed to use the Hamming distance as a fitness measure [19], and found better sequences than Adleman's original sequences. Zhang and Shin [20] used an iterative genetic search to design DNA sequences in the context of an evolutionary DNA-computing model called molecular programming. Arita *et al.* [21] developed a DNA sequence design system using a genetic algorithm with three fitness criteria. They designed self-complementary sequences for the Whiplash model and compared the results with a random generate-and-test algorithm.

B. Design Criteria

In DNA computing, the hybridization between a DNA sequence and its basepairing complement (also known as Watson–Crick pairing) is the most important factor to retrieve the information stored in DNA sequences and operate the computation processes. For this reason, we desire a set of DNA sequences to form a stable duplex (double stranded DNA) with their complements. We also need to ensure that two sequences which are not complemented each other do not interact. Noninteracting sequences should be prohibitive or relatively unstable, compared with any perfectly matched duplex formed from a DNA sequence and its complement [6]. These can improve the reliability of DNA sequences.

In Table II, we list the objective functions that have been used previously. The best system would be the system that includes all design criteria given in Table II. However, since some criteria overlap with others and the bio-lab methods may vary for different criteria, the criteria should be selected very carefully. These objectives can be classified into four categories: 1) preventing undesired reactions; 2) controlling secondary structures; 3) controlling the chemical characteristics of DNA sequences; and 4) restricting one of the DNA symbols (A, C, G, and T) in DNA sequences.

1) Preventing Undesired Reactions: This criterion forces the set of sequences to form the duplexes between a given DNA sequence and its complement only. This category includes most of the objectives in Table II. "Similarity" is defined as an inverse Hamming distance between two given DNA sequences. "H-measure" tests the possibility of unintended DNA basepairing based on the Hamming distance [28]. Shift means the position shift of DNA sequence to compare the two DNA sequences thoroughly. A more detailed explanation is given in Section IV-A. H-measure on the $3'$-end tests the H-measure with a shift in the $3'$-end. "Reverse complement Hamming distance" checks the Hamming distance using one DNA sequence and the reverse complement of another sequence.

TABLE II
SUMMARY OF THE DESIGN CRITERIA FOR THE PREVIOUS ALGORITHMS. THE DEVELOPED NACST/SEQ COVERS MOST OF THESE DESIGN CRITERIA. THE SYSTEMS ARE LISTED BY SIMILARITY TO NACST/SEQ

Design criteria	Design systems										
	NACST/Seq	[14]	[25]	[26]	[12]	[21]	[27][24]	[20]	[13]	[9]	[16][17]
Similarity (without shift)	O	O	O	O	O	O	O	O	O		
Similarity (with shift)	O	O		O	O			O			
H-measure (without shift)	O	O	O	O	O	O		O			
H-measure (with shift)	O	O	O		O			O			
3'-end H-measure		O				O					
Reverse complement hamming distance						O	O				
Secondary structure	O	O	O	O			O		O	O	
Continuity	O	O	O								
Free energy measure							O			O	O
Melting temperature	O	O		O	O					O	
GC ratio	O	O	O	O	O	O	O		O		
Bio-lab methods (PCR protocol)											O
Constraints on DNA bases	O			O							
Occurrence of specific subsequences			O		O				O		

2) Controlling the Secondary Structures: Secondary structures are usually formed by the interaction of single stranded DNA or RNA. "Secondary structure" includes internal loop, hairpin loop, and bulge loop. To predict the secondary structure, many algorithms have been proposed based on thermodynamic parameters [29], or, simply one can calculate the Hamming distance of given sequences by folding the sequences to hybridize with itself. "Continuity" tests the repeated run of identical bases. If one base is repeated, an unusual secondary structure can be formed. Though the secondary structure of DNA strands is usually prohibited, they could be adopted to perform DNA computing in several ways [30].

3) Controlling the Chemical Characteristics of DNA Sequences: In many cases, it is desirable to control DNA sequences to have similar chemical characteristics. Measures for this criterion include "free energy," "melting temperature," and "GC ratio." "GC ratio" is the percentage of guanine or cytosine in a whole DNA sequence. "Melting temperature" is defined as the temperature at which 50% of the oligonucleotide and its perfect complement are in duplex. "Free energy" is the necessary energy to make a duplex (actually, it is defined as the energy required to break a duplex). Given fixed protocols, one of the most reliable measure for the relative stability of a DNA duplex is its free energy. Melting temperature is also closely related to the free energy of a DNA duplex and a much less accurate measure of the stability is its GC content.

4) Restricting DNA Sequences: This criterion restricts the composition (DNA base or subsequence) of a DNA sequence. In some cases, one of four DNA bases is reserved for special purposes [26]. Also, special DNA subsequences such as the restriction enzyme site should be controlled for proper reactions. "Constraints on DNA bases" and "occurrence of specific subsequences" are related to this criterion.

C. Requirements of DNA Sequence Design Systems

Based on our review, the general requirements of DNA sequence generation systems can be summarized as follows.

- Sequence reliability: Reliability is the most important factor of a DNA sequence design system. All DNA sequence design criteria contribute to improving reliability. Additionally, users may want to choose the necessary design criteria to fit their specific situations.
- User friendliness: A user-friendly interface is necessary for easy use. For example, users should be able to specify the parameters easily through the user interface.
- Analysis capability: To examine the reliability of the designed sequence, users need to analyze the properties of the given sequence.
- Sequence reusability: Users need to save or load the DNA-sequence set to reuse it. Also, a small DNA sequence set can be incorporated to make a larger DNA-sequence set.

The most important feature is the sequence reliability which decides the characteristics of the designed sequences. Additionally, the other requirements are also necessary for the convenience of users.

III. OVERVIEW OF THE NACST/SEQ SYSTEM

As summarized in Section II-C, an ideal DNA sequence designer should offer at least four main features. These were the guidelines for our development of the NACST/Seq system. For sequence reliability, we employ an MOEA by covering as many design criteria as possible from those listed in Table II. For user friendliness, NACST/Seq is implemented using Qt library and C++ language for user interface on Linux platforms. NACST/Seq also adopts a plug-in architecture that makes it possible to develop each fitness plug-in separately and

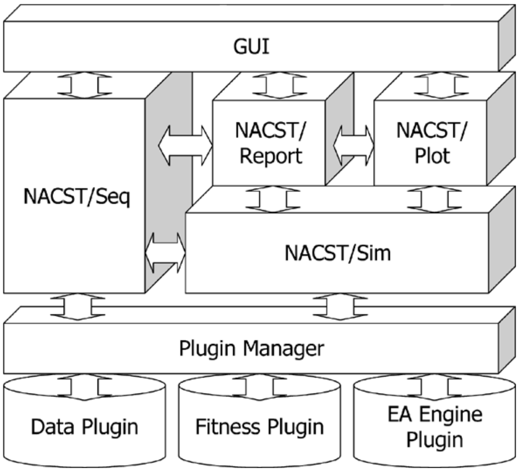


Fig. 1. Structure of nucleic acid computing simulation toolkit (NACST). NACST consists of four main modules: NACST/Seq for sequence generation, NACST/Report, NACST/Plot for analysis and visualization, and NACST/Sim for bio-lab simulation. Plug-in structure can help to add or delete the independent modules.

to assure a future extension. Additional components such as NACST/Report and NACST/Plot to analyze the sequences and to visualize the properties of the sequences are implemented for the analysis capability. Finally, NACST/Seq can save and load the sequences in XML format, generate partial DNA-sequence set, and add a sequence to the set manually to reuse DNA sequences.

The structure of the whole NACST system is shown in Fig. 1. NACST/Seq is responsible for generating DNA sequences (more details in Section IV), NACST/Sim simulates the laboratory experiments, NACST/Report and NACST/Plot analyze the results of the sequence generation or the lab simulation. In this paper, we focus on the NACST/Seq system.

The sequence generation steps in NACST/Seq are shown in Fig. 2. The first step is to select a generation option [Fig. 2(a)]. In this step, a user can generate new sequences and new sets. Otherwise, the user can extend the existing DNA sequence set by generating new sequences, or adding existing sequences to the set manually. The user can decide the sequence structures to prevent the generation of self-complementary sequences or to promote it. Then, the general sequence-option window appears, as shown in Fig. 2(b). In this window, the number of sequences and the length of each sequence can be adjusted. The fitness option window provides the functionality of combining the objectives [Fig. 2(c)]. If the selected fitness needs additional arguments, the user can call up the pop-up windows for tuning the arguments. Finally, the options for the EA are determined [Fig. 2(d)]. These include the number of generations, population size, and crossover and mutation rates. After execution of sequence generation, the main window shows the resulting set of sequences with their melting temperature and GC ratio.

NACST/Report is used to analyze the resulting sequence sets. NACST/Report can load any sequence sets saved in its XML format and examines various aspects of the loaded sequence sets such as the comparison of the fitness values [Fig. 3(a)], the graphical representation of the superiority of fitness value in each sequence between two selected sets [Fig. 3(b)], and the analysis of the sequences in a selected set [Fig. 3(c)].

As shown in Fig. 4, NACST/Plot can be used to visualize the results and the properties of the sequences. The plotted graphs can be saved as a postscript file and users can browse a plotting history. Finally, NACST/Sim is developed to simulate the biological experiments for handling DNA sequences [31]. It is used to verify the quality of designed sequences by estimating the probability of unintended hybridization such as non Watson–Crick DNA basepairing. After testing the designed sequences by NACST/Sim, the most reliable DNA sequence set which has the least unintended DNA basepairing will be recommended.

IV. Multiobjective Evolutionary Algorithms (EAs) for DNA Sequence Design

In this section, we describe the MOEA approach to DNA sequence design adopted in NACST/Seq. First, we describe the fitness criteria for sequence optimization in detail and investigate the characteristics of the fitness landscapes. The experimental results are given in the next section.

A. Fitness Criteria for DNA Sequence Design

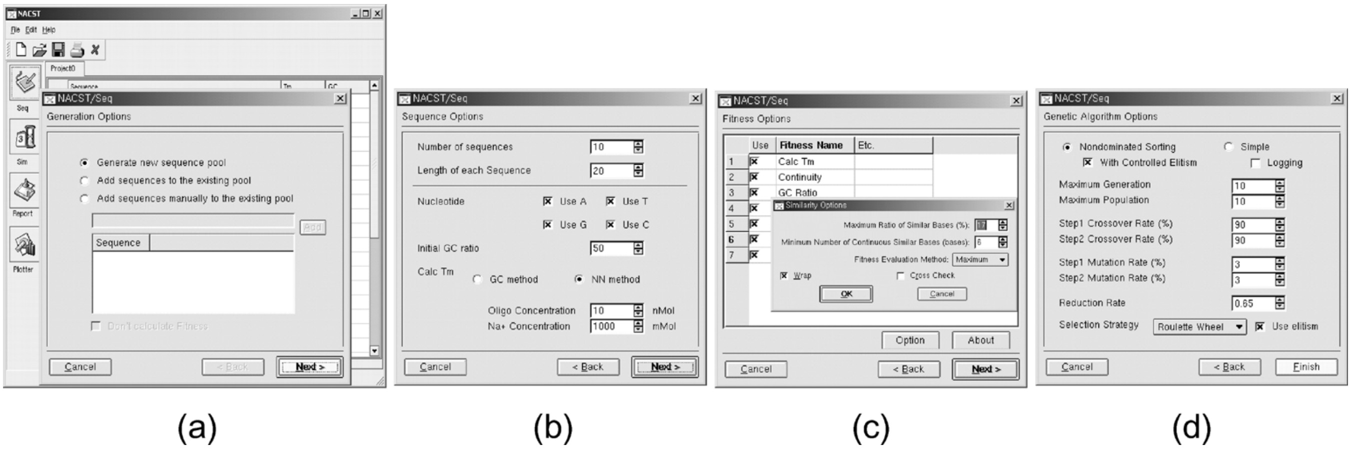
As shown in Table II, the objectives of NACST/Seq is very similar to [14]. That is why we try to accumulate the objectives as many as possible. The differences are constraints on DNA bases and $3'$-end H-measure. The reason why NACST/Seq does not use $3'$-end H-measure is that this factor is already considered in H-measure. Even if $3'$-end H-measure is important for PCR primer design, we do not want to overemphasize the $3'$-end H-measure. In addition, the sequence design algorithm is totally different from each other.

Formally, the DNA sequence design problem can be written as follows:

$$\begin{aligned} \text{minimize} \quad & F(x) = (f_1(x), f_2(x), \ldots, f_n(x)) \\ & f_i(x) \in \{\text{fitness measures in Table II}\}. \end{aligned} \tag{1}$$

Seven measures such as similarity, H-measure, secondary structure, continuity, melting temperature, GC content, and constraints on DNA bases can be considered in NACST/Seq as explained in Table II. Among them, the constraints on DNA bases are necessary for a special purpose design scheme such as stop sequence [26], not one of widely used sequence design objectives. Therefore, we do not consider it in this paper. The explanation of other objectives will be followed.

1) Similarity: The *similarity* measure $f_{\text{Similarity}}(x, y)$ computes the similarity in the same direction of two given sequences to keep each sequence as unique as possible including position shift. For a thorough comparison, we lengthen the sequence by its own sequence, and then calculate the fitness. For example, in the case of Fig. 5, the similarity between $5'$-$ATGCGCATGCATGAT$-$3'$ and $5'$-$TGGCATTGCCATGCA$-$3'$ is calculated. Sequence $5'$-$ATGCGCATGCATGAT$-$3'$ is extended by adding its own sequence to the $3'$-end with gaps (in Fig. 5, the gap is 3).



(a) NACST/Seq — Generation Options

- ● Generate new sequence pool
- ○ Add sequences to the existing pool
- ○ Add sequences manually to the existing pool
- Sequence
- □ Don't calculate Fitness

(b) NACST/Seq — Sequence Options

- Number of sequences: 10
- Length of each Sequence: 20
- Nucleotide: ☒ Use A ☒ Use T ☒ Use G ☒ Use C
- Initial GC ratio: 50
- Calc Tm: ○ GC method ● NN method
- Oligo Concentration: 10 nMol
- Na+ Concentration: 1000 mMol

(c) NACST/Seq — Fitness Options

	Use	Fitness Name	Etc.
1	☒	Calc Tm	
2	☒	Continuity	
3	☒	GC Ratio	
4	☒		
5	☒		
6	☒		
7	☒		

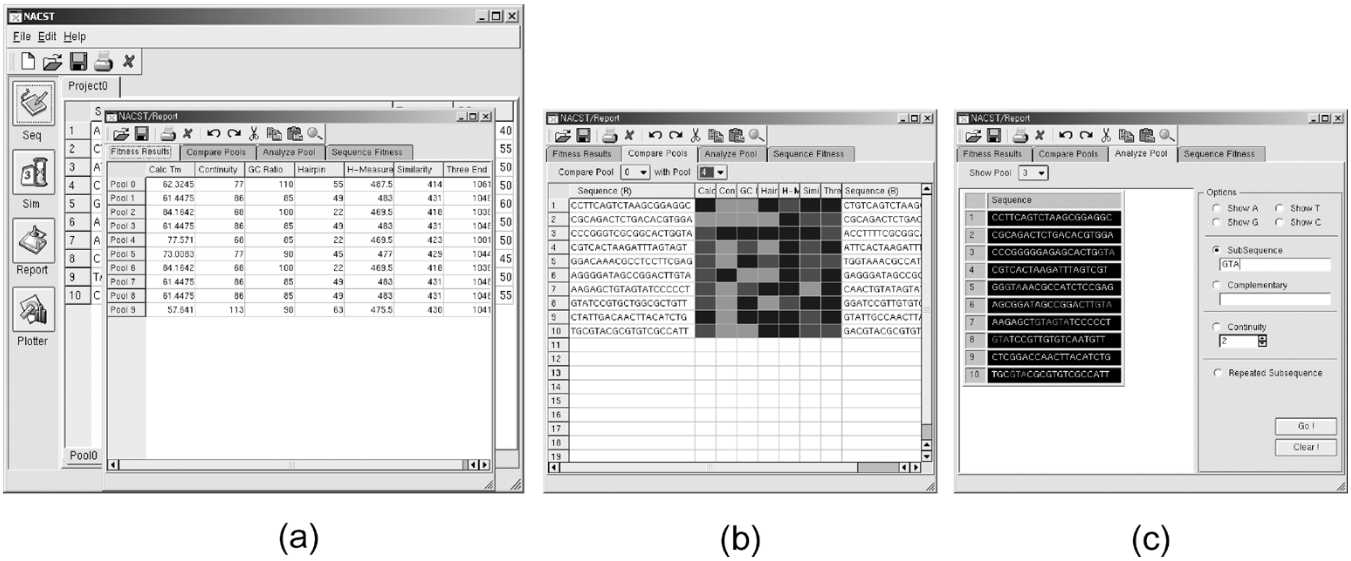
Similarity Options

- Maximum Ratio of Similar Bases (%): 17
- Minimum Number of Continuous Similar Bases (bases): 6
- Fitness Evaluation Method: Maximum
- ☒ Wrap □ Cross Check

(d) NACST/Seq — Genetic Algorithm Options

- ● Nondominated Sorting ○ Simple
- ☒ With Controlled Elitism □ Logging
- Maximum Generation: 10
- Maximum Population: 10
- Step1 Crossover Rate (%): 90
- Step2 Crossover Rate (%): 90
- Step1 Mutation Rate (%): 3
- Step2 Mutation Rate (%): 3
- Reduction Rate: 0.65
- Selection Strategy: Roulette Wheel ☒ Use elitism

Fig. 2. Sequence generation process in NACST/Seq. (a) General options such as the generation of new sequences or the addition of the existing sequences can be selected. (b) Sequence specific information can be set. (c) Options for each fitness function can be decided. (d) Options for EAs can be selected.



(a) NACST/Report — Fitness Results

	Calc Tm	Continuity	GC Ratio	Hairpin	H-Measure	Similarity	Three End
Pool 0	62.3245	77	110	55	487.5	414	1061
Pool 1	61.4475	86	85	49	483	431	1048
Pool 2	84.1642	68	100	22	469.5	418	1038
Pool 3	61.4475	86	85	49	483	431	1048
Pool 4	77.571	68	85	22	469.5	423	1001
Pool 5	73.0083	77	90	45	477	429	1044
Pool 6	84.1642	68	100	22	469.5	418	1038
Pool 7	61.4475	86	85	49	483	431	1048
Pool 8	61.4475	86	85	49	483	431	1048
Pool 9	57.641	113	90	63	475.5	430	1041

(b) NACST/Report — Compare Pools: Compare Pool 0 with Pool 4

	Sequence (R)	Calc	Con	GC	Hair	H-M	Simi	Thre	Sequence (B)
1	CCTTCAGTCTAAGCGGAGGC								CTGTCAGTCTAAG
2	CGCAGACTCTGACACGTGGA								CGCAGACTCTGAC
3	CCCGGGTCGCGGCACTGGTA								ACCTTTTCGCGGC
4	CGTCACTAAGATTTAGTAGT								ATTCACTAAGATT
5	GGACAAACGCCTCCTTCGAG								TGGTAAACGCCAT
6	AGGGGATAGCCGGACTTGTA								GAGGGATAGCCGG
7	AAGAGCTGTAGTATCCCCCT								CAACTGTATAGTA
8	GTATCCGTGCTGGCGCTGTT								GGATCCGTTGTGT
9	CTATTGACAACTTACATCTG								GTATTGCCAACTT
10	TGCGTACGCGTGTCGCCATT								GACGTACGCGTGT

(c) NACST/Report — Analyze Pool: Show Pool 3

	Sequence
1	CCTTCAGTCTAAGCGGAGGC
2	CGCAGACTCTGACACGTGGA
3	CCCGGGGGAGAGCACTGGTA
4	CGTCACTAAGATTTAGTCGT
5	GGGTAAACGCCATCTCCGAG
6	AGCGGATAGCCGGACTTGTA
7	AAGAGCTGTAGTATCCCCCT
8	GTATCCGTTGTGTCAATGTT
9	CTCGGACCAACTTACATCTG
10	TGCGTACGCGTGTCGCCATT

Options

- ○ Show A ○ Show T
- ○ Show G ○ Show C
- ● SubSequence: GTA
- ○ Complementary
- ○ Continuity: 2
- ○ Repeated Subsequence

Fig. 3. Demonstration of NACST/Report. NACST/Report shows the properties of the sequences and compares the sequence sets. (a) All sequence sets can be examined by the comparison of those fitness values measured through each objective used in the optimization procedure. (b) NACST/Report provides a graphical representation of the superiority of fitness value in each sequence between two selected pools. (c) A selected pool can be analyzed. NACST/Report can highlight the position of a specific subsequence in a pool, find all complementary subsequences of user's input sequence, and mark all successive occurrence of the same base running over the threshold.

We distinguish the continuous similarity where the same substrings appear in two sequences from the discontinuous similarity, where the overall trend of two resembles each other. For example, the sequences $5'$-$\underline{A}T\underline{GCAT}G\underline{C}ATGC$-$3'$ and $5'$-$AAGCATCCCGAA$-$3'$ have continuous similarity of 4 ("GCAT") from the third base to the sixth and discontinuous similarity value of 6 (the underlined bases) without a position shift.

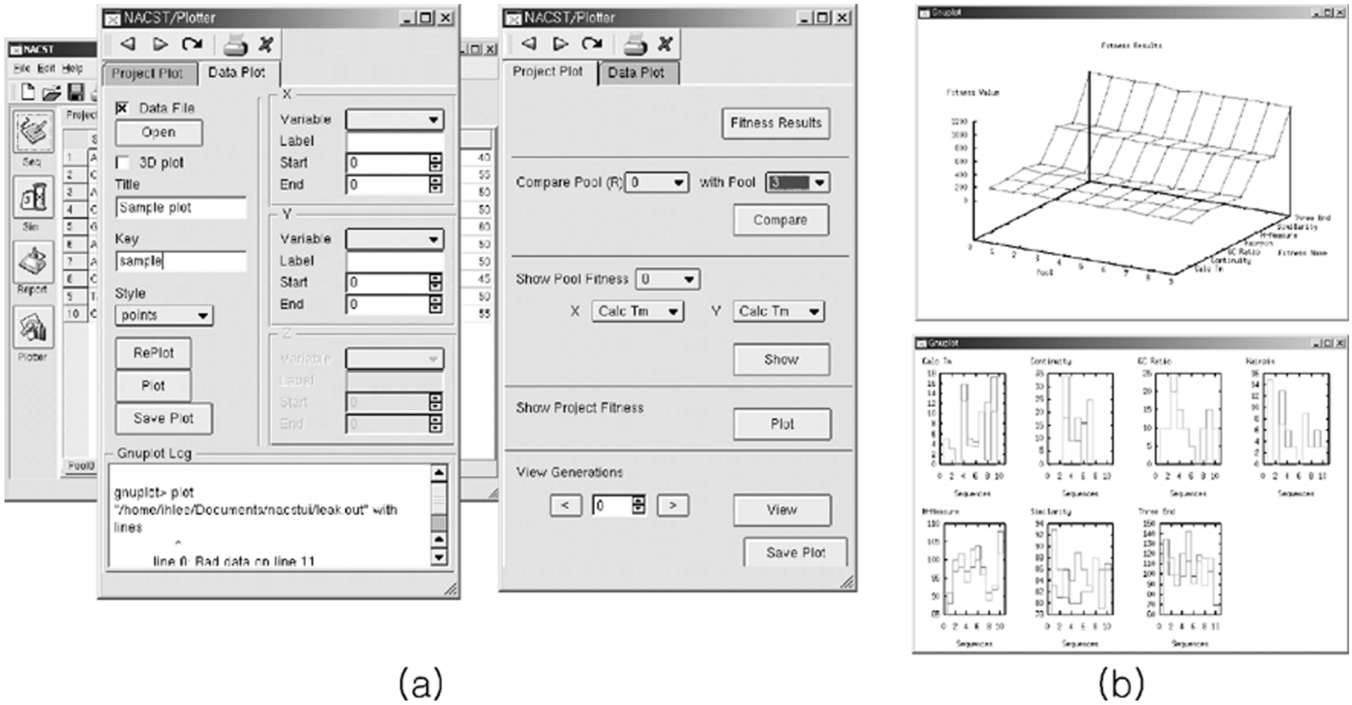
2) H-measure: The H-measure $f_{\text{H-measure}}(x,y)$ considers two sequences as complementary, while similarity regards two sequences as parallel (Fig. 6). That is, H-measure computes how many nucleotides are complementary between the given sequences to prevent cross-hybridization of two sequences. As similarity, H-measure also uses the elongated sequence.

3) Hairpin: This measure calculates the probability to form a secondary structure. We made two variants of hairpin fitness measure, one for preventing the hairpin structure and the other for making planned hairpin structure

$$f_{\text{Hairpin}}(x) = \text{HP}(x) + \text{HP}_{\text{desired}}(x). \quad (2)$$

Even though there are many algorithms that can predict the DNA or RNA secondary structure including hairpin with thermodynamic parameters [32], we calculate the Hamming distance for simplicity by considering the length of hairpin loop and the number of hybridized pairs as illustrated in Fig. 7.

4) Continuity: If same bases occur continuously in a sequence, the sequence can show the unexpected structures. $f_{\text{Continuity}}(x)$ calculates degree of successive occurrence of the same base, as shown in Fig. 8. For more controllable experiments, we can check purine and pyrimidine continuity, as well as adenine, thymine, guanine, and cytosine continuity.



```
gnuplot> plot
"/home/ihlee/Documents/nacstui/leak.out" with
lines
        ^
        line 0: Bad data on line 11
```

Fig. 4. NACST/Plot. (a) Plot of the data file which is saved in XML format of the sequence set. (b) Plot of the fitness results. Also, the comparison results of selected two sets can be plotted.

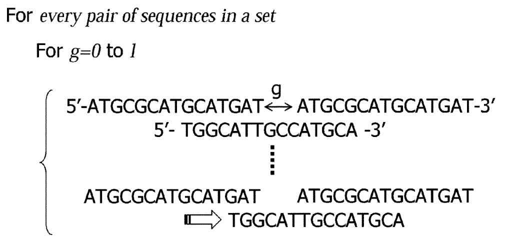

For every pair of sequences in a set
For g=0 to 1
g
5′-ATGCGCATGCATGAT ⟷ ATGCGCATGCATGAT-3′
5′- TGGCATTGCCATGCA -3′
ATGCGCATGCATGAT ATGCGCATGCATGAT
TGGCATTGCCATGCA

Fig. 5. Example of similarity measure. The similarity measure checks how many positions are the same given two strands. Two DNA strands are chosen and one DNA strand is elongated with gap g. Then, the elongated strand and the other strand are compared in each position with position shifts. After all shifts are considered, gap g is increased by one to compare two strands continuously.

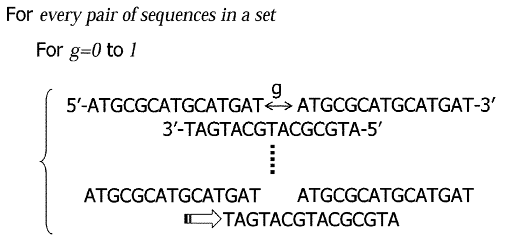

For every pair of sequences in a set
For g=0 to 1
g
5′-ATGCGCATGCATGAT ⟷ ATGCGCATGCATGAT-3′
3′-TAGTACGTACGCGTA-5′
ATGCGCATGCATGAT ATGCGCATGCATGAT
TAGTACGTACGCGTA

Fig. 6. H-measure is similar to the similarity measure. The difference is H-measure compares the given two strands with opposite direction, $5'$-$3'$ and $3'$-$5'$, whereas similarity measure checks them with the same direction. H-measure calculates the unintended DNA base pairing rate.

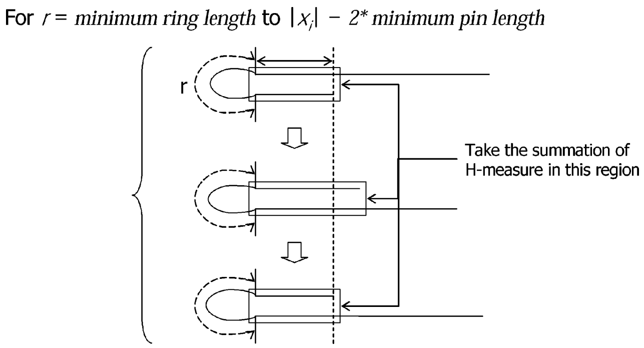

For r = minimum ring length to |x_i| − 2* minimum pin length
r
Take the summation of H-measure in this region

Fig. 7. Hairpin is a self-hybridization by forming the loop. First r (ring length) is initialized, then the number of hybridized pairs is checked. The position of the ring is shifted, as shown in the figure. If the shift ends, the r is increased.

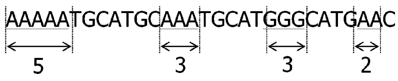


Fig. 8. Continuity measure prohibits the consecutive runs of the same base over the given threshold. If the threshold is four, the first run violates the continuity and others do not.

5) Melting Temperature: Melting temperature is one of the most important features for laboratory experiment. There are many equations to calculate *melting temperature* such as the Wallace 2–4 rule [33], the GC% method [34], and the nearest-neighbor model [35]. We use the GC% method and nearest neighbor model (NN) with SantaLucia's unified NN parameters [35] to calculate melting temperature. f_{Tm} is defined as follows:

$$f_{Tm}(x) = (Tm_{\text{target}}(x) - Tm_{\text{generated}}(x))^2 \tag{3}$$

where Tm_{target} is the target melting temperature, and $Tm_{\text{generated}}$ is the melting temperature of the generated sequence.

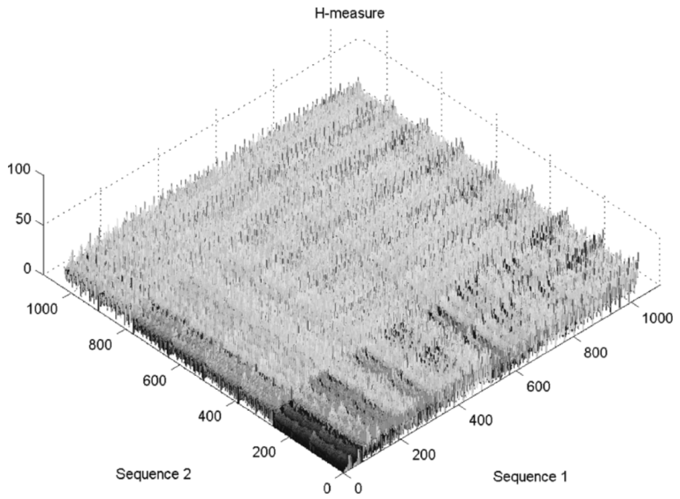


Fig. 9. Plot of H-measure values between 5 bp sequences. The x and y axis show all of the five-mer DNA sequences represented by decimal numbers. For example, z-value of (0,2) is the value of the H-measure between sequences "AAAAA" and "AAAAG."

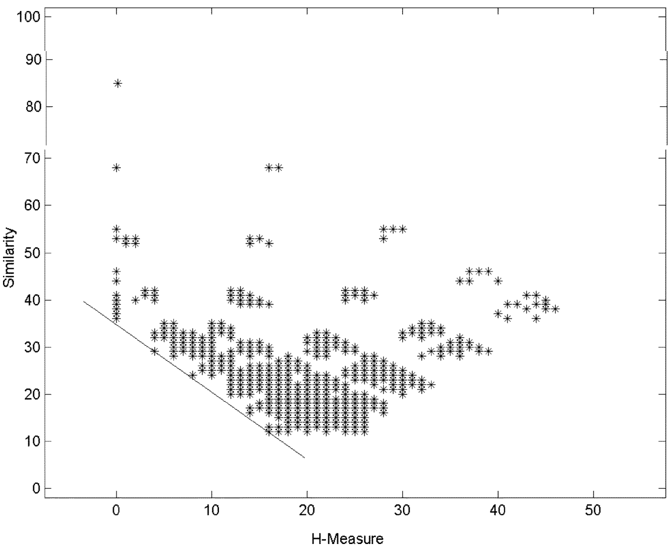


Fig. 10. Objective space of H-measure and similarity for 5 bp DNA sequences. The line depicts the tradeoff surface.

6) GC Content: The *GC content* is the percentage of G and C in a sequence. Since GC content can affect the chemical properties of DNA sequences, it is important criteria

$$f_{\mathrm{GCcontent}}(x) = (\mathrm{GC}_{\mathrm{target}}(x) - \mathrm{GC}_{\mathrm{generated}}(x))^2. \quad (4)$$

More mathematical definitions for these measures are given in the Appendix.

B. Analysis of DNA Sequence Fitness Measures

We investigated fitness objectives to determine the characteristics of DNA sequence optimization. First, we plot the H-measure and similarity for two 5-mer sequences, since H-measure and similarity are the most important objectives. Fig. 9 shows the fitness landscape of H-measure value of two 5-mer DNA sequences. The x and y axis represent the DNA sequences. Setting "A" as 0, "C" as 1, "G" as 2, and "T" as 3, a DNA sequence can be thought as a number of base 4. Then, a sequence can be represented by the decimal number corresponding to this number. For example, a sequence "AAAAA" can be represented by the number $00\,000_{(4)} = 00_{(10)}$ and a sequence "ACGCA" by $01\,210_{(4)} = 100_{(10)}$. The optimal value of H-measure is zero. As shown in Fig. 9, the fitness landscape of H-measure shows many local optima, and it is hard to find any gradient information to the global optimum. Additionally, H-measure is a discrete function. The details are described in the Appendix. The fitness landscape of similarity value is similar to that of the H-measure. Both functions are discrete and have many local optima.

Next, to find out the relationship between H-measure and similarity, we plotted the objective space of H-measure and similarity in Fig. 10. The points on the line in Fig. 10 depicts the best nondominated solutions in the objective space. The objective space of H-measure and similarity shows the traditional relationship of conflict objectives in multiobjective optimization problems. We also analyzed other fitness objectives in this manner. Among them, H-measure and similarity conflicted most with each other. GC ratio and similarity also showed conflict. Other objectives did not conflict with each other, and they had only one optimum point as sphere function. Nevertheless, they were all discontinuous functions which are hard to find the optimum. Therefore, we can conclude that DNA sequence design problem is a hard multiobjective optimization problem.

C. Algorithm of NACST/Seq

As shown in Section IV-B, DNA sequence optimization has many fitness functions which conflict each other. Therefore, MOEA is a natural candidate for DNA sequence optimization. In addition, if we use MOEA, we can easily add or delete fitness objective for sequence optimization. In DNA sequence optimization, selection of fitness objective could be diverse for the purpose of DNA sequences. Therefore, the possibility of easy addition or deletion of objective functions without changing program can be another merit of MOEA for DNA sequence optimization.

Among various MOEAs [36]–[38], we implement NACST/Seq based on a controlled elitist nondominated sorting genetic algorithm (NSGA-II) [39], which is the one of the most widely used MOEAs for the real-world problems. Since DNA sequence optimization has various fitness objectives, as shown in Section IV-A, we need the algorithm to balance convergence and diversity as well as to handle a number of objectives. NSGA can handle a number of objectives through ranking by nondominated sorting procedure. It also showed good convergence and diversity performance among various MOEAs [37]. In addition, the crowding distance measure in NSGA-II overcomes a drawback of deciding the sharing parameter.

In multiobjective optimization, the problem difficulty varies rather interestingly with the number of objectives [40]. Especially, there are a bunch of objectives in DNA sequence optimization compared with other multiobjective problems. Therefore, maintaining the exploration ability is an important issue within DNA sequence optimization. Usually, there are two general approaches to this problem: using a large population size and using a modified algorithm for assigning fitness [41]. Since

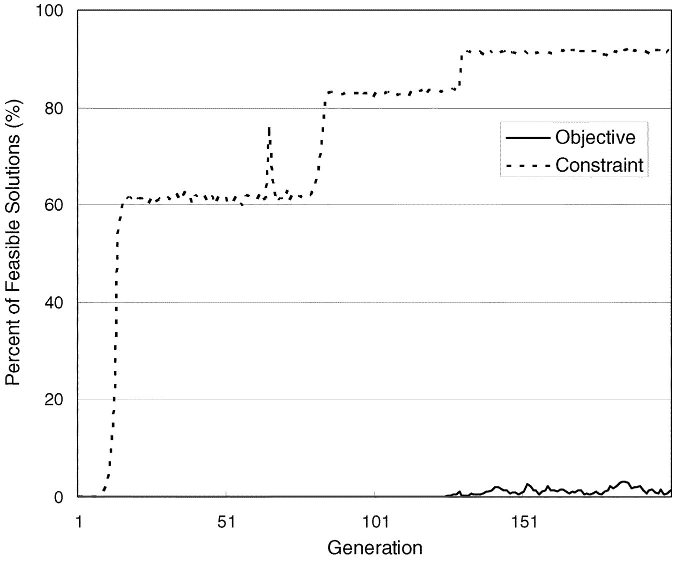


Fig. 11. Percentage of feasible solutions over population through generations for the constrained MOEA and the normal MOEA. The dotted line is the result when the constrained tournament selection is used. The solid line shows that the penalty is treated as an additional objective. The constrained MOEA finds feasible solutions quickly and keeps them in front.

increasing population size requires long computational time and it is a difficult task to modify algorithm to handle a number of objectives, we use the NSGA-II with controlled elitism strategy. Under this approach, the number of individuals in each front is restricted to follow a predefined distribution. This allows to maintain diverse population as well as to utilize the advantage of elitism strategy.

We modified NSGA-II to work within the scope of DNA sequence optimization. NACST/Seq adopts a hierarchical representation including individual level and a sequence level. The crossover and mutation operators are divided into two steps.

Step 1) Individual level operation which is regarded as an exchange of member sequences between two individuals.

Step 2) Belongs to the sequence level which is the same as multipoint crossover in simple genetic algorithms.

The mutation operator changes DNA base at random position. Using both operators showed empirically better results than using one of operators, and tournament selection with tournament size two was used to select the individuals. We used tournament size two to reduce the computational time. Also, empirically there was no significant improvement in using a larger tournament size. To use the controlled NSGA-II, it is important to decide the reduction rate which decides the size of each front. In previous work [41], since a reduction rate 0.65 showed the best performance, we adopt 0.65 in NACST/Seq.

```
evolve($P$)
    $R = P \cup Q$
    evaluate($R$)
    $n$ = constrained-nondominated-sort($R$)
    for each $1 \leq k \leq n$
        if(all solution in front($R$, $k$) is feasible)
            crowding-distance-assignment(front($R, k$))
    $P$ = controlled-parent-selection($R$)
    $Q = \{\}$
    while $|Q| < N$
        $p_1$ = constrained-tournament-selection($P$)
        if (step1 crossover) then
            $p_2$ = constrained-tournament-selection($P$)
            if (step2 crossover) then
                $Q = Q \cup$ {crossover2($p_1$,$p_2$)}
            else
                $Q = Q \cup$ {crossover1($p_1$,$p_2$)}
            if (mutation) then mutate($Q_{p_1}, Q_{p_2}$)
        else
            $Q = Q \cup \{p_1\}$
            if (mutation) then mutate($Q_{p_1}$)
```

Fig. 12. Pseudocode for the main procedure of NACST/Seq. P is the current population and Q is the new population.

Then, constraint handling technology is applied to DNA sequence optimization. Since some objectives such as melting temperature and GC content can be naturally regarded as constraints, not objective functions, the more precise formulation of DNA sequence optimization is a constrained MOEA. Additionally, when using constraint handling technique, we can find more reliable solution in early generation by decreasing the number of objectives. Therefore, (1) can be modified as follows:

$$\begin{aligned} \text{Optimize } & f_i(x), \quad i \in \{\text{continuity, hairpin,} \\ & \qquad\qquad \text{H-measure, similarity}\} \\ \text{subject to } & g_j(x) = 0, \quad j \in \{Tm, \text{GCcontent}\}. \end{aligned} \tag{5}$$

Here, we use constrained tournament selection to handle the penalty functions. There are three cases in the constrained tournament to drive infeasible solutions toward the feasible region [41]: if infeasible solution and feasible solution are selected, feasible solution is selected; if both infeasible solutions are chosen, one with less penalty wins; when both feasible solutions are selected, if any one dominates the other, dominating one is selected, if not, one with larger crowding distance wins.

To demonstrate the effectiveness of constraint handling method in finding feasible solutions, we plot the percentage of the feasible solutions over population for the constrained MOEA and the normal MOEA. Fig. 11 depicts the result of the generation of seven 20-mer DNA sequences. Only GC content was kept at 50%. Except the constraint handling method, the same parameters are used: the population size was 1000, maximum generation was 200, crossover rate was 0.9, mutation rate was 0.01, and reduction rate was 0.5. As shown in Fig. 11, the constrained tournament selection finds feasible solutions quickly and keeps them in front. On the other hand, when the

TABLE III
COMPARISON RESULTS OF THE SEQUENCES IN [43] AND THE SEQUENCES BY NACST/SEQ. THE FITNESS IS EXAMINED BY NACST/REPORT

Sequence ($5' \rightarrow 3'$)	Continuity	Hairpin	H-measure	Similarity	Tm	GC%
Deaton *et al.*						
ATAGAGTGGATAGTTCTGGG	9	3	55	64	52.6522	45
CATTGGCGGCGCGTAGGCTT	0	0	69	51	69.2009	65
CTTGTGACCGCTTCTGGGGA	16	0	60	63	60.8563	60
GAAAAAGGACCAAAAGAGAG	41	0	58	45	52.7111	40
GATGGTGCTTAGAGAAGTGG	0	0	58	54	55.3056	50
TGTATCTCGTTTTAACATCC	16	4	61	50	48.4451	35
TTGTAAGCCTACTGCGTGAC	0	3	75	55	56.7055	50
NACST/Seq						
CTCTTCATCCACCTCTTCTC	0	0	43	58	46.6803	50
CTCTCATCTCTCCGTTCTTC	0	0	37	58	46.9393	50
TATCCTGTGGTGTCCTTCCT	0	0	45	57	49.1066	50
ATTCTGTTCCGTTGCGTGTC	0	0	52	56	51.1380	50
TCTCTTACGTTGGTTGGCTG	0	0	51	53	49.9252	50
GTATTCCAAGCGTCCGTGTT	0	0	55	49	50.7224	50
AAACCTCCACCAACACACCA	9	0	55	43	51.4735	50

GC content is treated as the one of the objectives, infeasible solutions are hardly removed from population. This prevents one from finding the optimal solution.

The entire NACST/Seq procedure is summarized in Fig. 12. First, parent and offspring population are united and evaluated. Then, new parent population is formed by constrained tournament selection. From these new parents, new offsprings are generated by variation operators such as two-step crossover and mutation [42].

V. SEQUENCE DESIGN RESULTS BY NACST/SEQ

A. *Comparison With Other Sequence Design Systems*

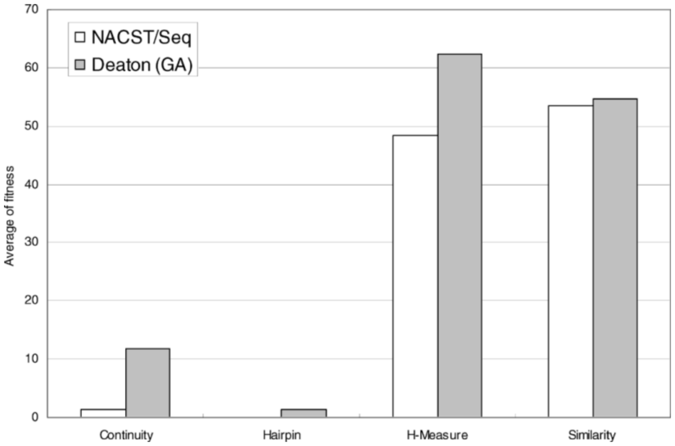


Fig. 13. Average fitness comparison results between Deaton and NACST/Seq. Y axis indicates the average values of each fitness objective in Table III. NACST/Seq shows better results for all objectives.

We compared our algorithm with other approaches to show the performance of NACST/Seq. We analyzed sequences from several publications and those generated by NACST/Seq with comparable restrictions. Sequences were taken from the results of Deaton *et al.* [17], [43], Tanaka *et al.* [44], and Faulhammer *et al.* [26]. The parameters for NACST/Seq were as follows. For H-measure and similarity, we set lower limits for the continuous case equal to six bases and those for the discontinuous case to 17%. For continuity, the threshold value was 2. We assumed that hairpin formation requires at least six basepairings and a six base loop. Though typical minimum settings are stem length 3 and loop length 3, we assumed 6 for both to reduce the computational time. All these values were decided empirically with biochemical background. The melting temperature was decided by the nearest neighbor (NN) method with 1 *M* salt concentration and 10 *nM* DNA concentration. The reduction rate was 0.65 as explained in [41]. The population size was 3000 and the maximum generation was 200 for [26] and [43]. The population size was 5000 and the maximum generation was 300 for [44]. Here, we constrained only the GC ratio, since the range of melting temperature was not given in the references. For [43] and [44], the GC ratio was restricted to 50%. For [26], the range of the GC ratio was 40%∼50%. Among the various DNA sequence set as a result of MOEA, we chose the best one by NACST/Sim which is the biochemical reaction simulator [31].

First, we compared NACST/Seq with [43]. In [43], a genetic algorithm was used to design good sequences for Adleman's graph. The comparison results are shown in Table III and Fig. 13. Fig. 13 shows the average values of all sequences in Table III. The sequences generated by NACST/Seq outperform the sequence in [43]. Our sequences show much lower H-measure and similarity values. This implies the sequences made by NACST/Seq have much higher probability to hybridize with the its correct complementary sequences. The secondary structure is more prohibited due to the very low continuity and hairpin, and the width of the range of melting temperatures and GC ratio are better. In our sequences, the longest length of the

TABLE IV
COMPARISON RESULTS OF THE SEQUENCES IN [44] AND OUR SEQUENCES

Sequence ($5' \rightarrow 3'$)	Continuity	Hairpin	H-measure	Similarity	Tm	GC%
Tanaka's sequence						
CGAGACATCGTGCATATCGT	0	7	143	124	59.6965	50
TATAGCACGAGTGCGCGTAT	0	3	137	130	62.1165	50
GATCTACGATCATGAGAGCG	0	4	135	126	56.1049	50
TCTGTACTGCTGACTCGAGT	0	9	163	124	56.5723	50
CGAGTAGTCACACGATGAGA	0	0	152	132	56.2894	50
AGATGATCAGCAGCGACACT	0	3	133	133	58.9724	50
TGTGCTCGTCTCTGCATACT	0	10	159	130	58.5736	50
AGACGAGTCGTACAGTACAG	0	0	152	134	54.9689	50
ATGTACGTGAGATGCAGCAG	0	0	139	121	57.8232	50
ATCACTACTCGCTCGTCACT	0	3	141	132	58.0122	50
TCAGAGATACTCACGTCACG	0	3	142	123	55.3226	50
GACAGAGCTATCAGCTACTG	0	3	129	124	54.565	50
GCTGACATAGAGTGCGATAC	0	0	130	133	56.5849	50
ACATCGACACTACTACGCAC	0	3	133	144	57.2186	50
NACST/Seq						
GTGACTTGAGGTAGGTAGGA	0	3	129	115	47.2490	50
ATCATACTCCGGAGACTACC	0	3	132	121	47.2304	50
CACGTCCTACTACCTTCAAC	0	0	128	121	47.4589	50
ACACGCGTGCATATAGGCAA	0	3	141	117	52.5401	50
AAGTCTGCACGGATTCCTGA	0	3	132	115	50.5497	50
AGGCCGAAGTTGACGTAAGA	0	0	132	116	51.0482	50
CGACACTTGTAGCACACCTT	0	0	132	123	50.2683	50
TGGCGCTCTACCGTTGAATT	0	0	135	116	52.0565	50
CTAGAAGGATAGGCGATACG	0	0	134	117	46.6253	50
CTTGGTGCGTTCTGTGTACA	0	0	140	116	50.5774	50
TGCCAACGGTCTCAACATGA	0	0	132	121	51.8587	50
TTATCTCCATAGCTCCAGGC	0	0	136	117	48.1017	50
TGAACGAGCATCACCAACTC	0	0	121	121	50.3351	50
CTAGATTAGCGGCCATAACC	0	0	127	119	47.6383	50

repeated base is three which appears only once, whereas it is five that of Deaton *et al.* was applied.

Then, we compared the sequences in [44] generated by simulated annealing. Tanaka *et al.* designed a set of 20 DNA sequences whose length is 20-mer, as shown in Table IV. The evaluation function was a weighted sum of different terms including H-measure, self-complementarity, GC portion, melting temperature, self-complementarity, complete hybridization at the $3'$-end, and continuity. In the generated sequences, the continuous bases did not appear, GC portion of all sequences was 50%, H-measure and similarity values were relatively small. However, NACST/Seq can find better sequences in all measures, as shown in Table IV and Fig. 14, where GC portion and continuity value are the same, 50% and 0%, respectively. Fig. 14 shows comparison results in terms of average of fitness. For hairpin, H-measure, and similarity, the sequences designed by NACST/Seq show much better properties. Even the width of the range of melting temperatures is better.

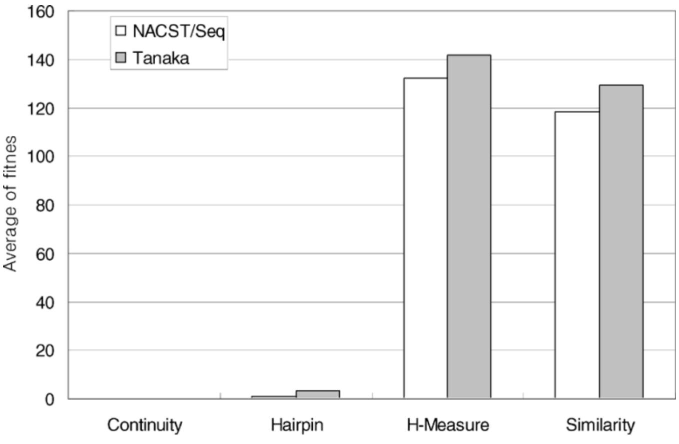


Fig. 14. Comparison results between average values of Tanaka and NACST/Seq.

Table V presents the sequences for the chess knight movement problem in [26]. Referring to [26], these sequences were

TABLE V
COMPARISON RESULTS OF THE SEQUENCES IN [26] AND NACST/SEQ

Sequence ($5' \rightarrow 3'$)	Continuity	Hairpin	H-measure	Similarity	Tm	GC%
Faulhammer's sequence						
CTCTTACTCAATTCT	0	0	114	162	36.1609	33.33
CATATCAACATCTTA	0	0	130	175	35.9744	26.67
ATCCTCCACTTCACA	0	0	107	187	41.6455	46.67
TTAAAATCTTCCCTC	25	0	121	159	35.4845	33.33
CTATTTATCCACACC	9	0	109	175	36.6125	40
GCTTCAAACAATTCC	9	0	117	154	41.2682	40
AACTCTCAAATTCAA	9	0	132	158	38.1608	26.67
CTAACCTTTACTTCA	9	0	120	178	35.2144	33.33
CATTCCTTATCCCAC	9	0	100	178	38.4742	46.67
CACCCTTTCTCCTCT	18	0	88	159	41.0117	53.33
TCCTCACATTACTTA	0	0	119	172	35.9641	33.33
ACTTCCTTTATATCC	9	0	116	168	33.0108	33.33
TTATAACAAACATCC	9	0	131	157	35.813	26.67
ACATAACCCTCTTCA	9	0	116	179	39.5534	40
ACCTTACTTTCCATA	9	0	118	174	34.8949	33.33
GTACATTCTCCCTAC	9	0	114	155	38.5983	46.67
CATAATCTTATATTC	0	0	131	175	30.7646	20
ATAATCACATACTTC	0	0	125	172	34.713	26.67
TCCACCAACTACCTA	0	0	104	159	41.5263	46.67
TTTTAAATTTCACAA	34	0	137	166	31.1984	13.33
NACST/Seq						
AAGAAAGGCGAAGAA	9	0	116	124	37.1454	40
CAACAAGAGCACATA	0	0	109	150	35.2693	40
CAACAGGACAAACGA	9	0	104	151	38.8994	46.67
AACCACCACTTCCTA	0	0	107	153	37.6383	46.67
TCTCTCACACATCTC	0	0	107	140	35.8936	46.67
AACAGCCTAACCGTA	0	0	115	146	38.7104	46.67
TGCATCCTTTCCTCT	9	0	125	121	38.3245	46.67
GGCATAACCACTCTT	0	0	124	143	37.2994	46.67
GAAGGCAGTCACTTA	0	0	136	123	37.0216	46.67
AAAAAGCACAGCTAC	25	0	108	147	36.4861	40
CCAAACAAACCGAGA	18	0	96	154	38.7244	46.67
AACGACAACGAACAA	0	0	97	160	38.8840	40
CACAACCTAACACCA	0	0	85	162	37.6240	46.67
CAATCCTTCTCGTTC	0	0	129	129	36.2532	46.67
CAACAAACAGGCTAC	9	0	105	159	37.2122	46.67
CCACTACATCTCTAA	0	0	117	157	32.0238	40
GGTTATCTATCTCCA	0	0	135	133	31.4527	40
CATCCACCTCAATTC	0	0	107	140	35.7989	46.67
AACTACGGACCTATT	0	0	123	131	34.4633	40
ACACCATAACAACAC	0	0	85	161	35.4422	40

verified by real-laboratory experiments. Faulhammer *et al.* [26] generated 20 15-mer DNA sequences using PERMUTE that repaired sequences until they met the criteria such as Hamming distance, melting temperature, and so on. As in the previous case, our sequences are more dissimilar, prohibit unintended DNA basepairing, and have the smaller range of melting temperature and GC ratio (refer to Fig. 15). Both sequences do not have hairpin formation. If we relax the constraint of the GC ratio, we can get better sequences for H-measure and similarity.

Finally, we compared the sequences in [17]. Unlike the previous methods, the minimum free energy for duplex formation between two given sequences was calculated. They showed 40

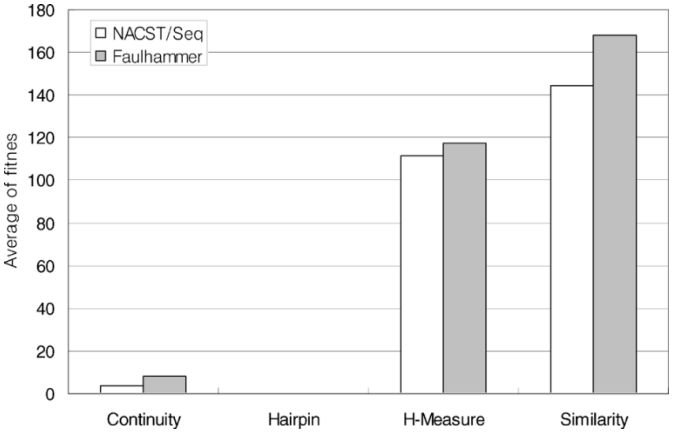


Fig. 15. Comparison results between Faulhammer and NACST/Seq. NACST/Seq outperforms Tanaka *et al.*'s algorithm.

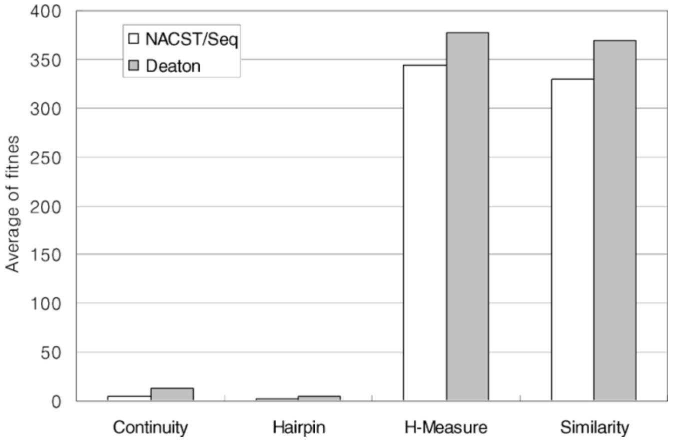


Fig. 16. Comparison results of Deaton (new method) and NACST/Seq. NACST/Seq shows better results for all objectives.

DNA sequences among generated 3595 sequences. However, NACST/Seq can also find more promising DNA sequence set. Fig. 16 shows the comparison results. While they used free energy calculation to optimize DNA sequences, we applied free energy term to select best candidate set among various set of MOEA run by NACST/Sim. As we chose the final sequence using NACST/Sim, our sequences show better performance in H-measure and similarity, even if Deaton *et al.* [17] used free energy calculation term directly.

B. Application to the Traveling Salesman Problem (TSP)

At the first trial, we used the conventional EA with multiple-point crossover, single-point mutation, and RW selection. The fitness function was the weighted sum of required fitness measures

$$\text{fitness} = \sum_i w_i f_i,$$
$$f_i \in \{f_{\text{similarity}}, f_{\text{H-measure}}, f_{\text{Hairpin}}, f_{\text{continuity}}, f_{Tm}, f_{\text{GCcontent}}\}. \qquad (6)$$

For simplicity, we set each weight to one. The best fitness value is zero, therefore, we have to minimize (6).

TABLE VI

GENERATED VERTEX SEQUENCES FOR SEVEN-TSP. BOTH SEQUENCES BY THE CONVENTIONAL EVOLUTIONARY ALGORITHM AND BY THE PROPOSED ALGORITHM ARE SHOWN

No.	conventional evolutionary algorithm	Multi-objective evolutionary algorithm
	Sequence ($5' \rightarrow 3'$)	Sequence ($5' \rightarrow 3'$)
0	AGGCGAGTATGGGGTATATC	AATAGGAGCAGGAGACAACG
1	CCTGTCAACATTGACGCTCA	CTCTCATCTCTCCGTTCTTC
2	TTATGATTCCACTGGCGCTC	TATCCTGTGGTGTCCTTCCT
3	ATCGTACTCATGGTCCCTAC	ATTCTGTTCCGTTGCGTGTC
4	CGCTCCATCCTTGATCGTTT	TCTCTTACGTTGGTTGGCTG
5	CTTCGCTGCTGATAACCTCA	TAGTTCCAAGCGTCCGTGTT
6	GAGTTAGATGTCACGTCACG	TATCCACACCAACACACCAC

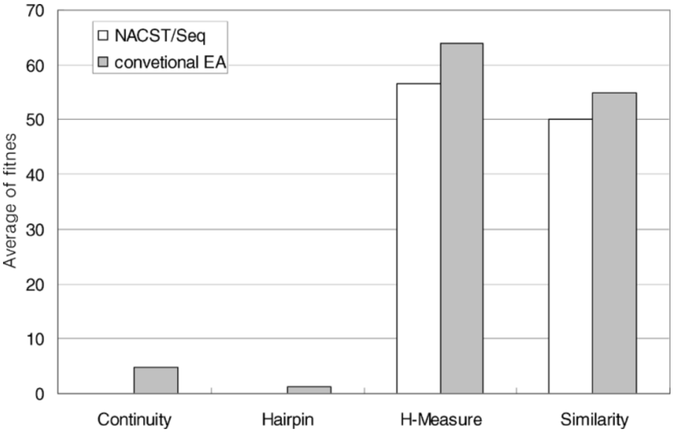


Fig. 17. Comparison results between conventional EA and NACST/Seq.

We designed DNA sequences for the TSP using conventional EA. The target TSP was a seven-node incomplete graph, seven vertex sequences, five weight sequences, and 24 edge sequences were generated to solve this problem. The population size was 1000, maximum generation was 1000, crossover rate was 0.9, and mutation rate was 0.05. The parameters for DNA sequences were the same as the setting of the previous section. For more detailed description, see [45]. The top seven DNA sequence for seven-TSP designed by EA in Table VI were verified by the biological experiments [46].

As explained earlier, the properties of DNA sequence optimization are suitable for MOEA, but not conventional EA. We generated the vertex sequence for the TSP by the proposed constrained MOEA to compare with the result of EA. The population size was 3000, the generation size was 200, and the reduction rate was 0.65. Although the population size is bigger than EA, the number of iteration is much smaller. The sequences by MOEA are compared to those by EA in Table VI.

Table VII and Fig. 17 show the comparison results of conventional EA and MOEA. The sequences designed by MOEA are better than those by EA for several reasons. In MOEA results, there are no repeated nucleotides (continuity) and no hairpin structures. H-measure and similarity measures in MOEA are also much more suitable for a laboratory experiment. The GC ratio is the same. Only the width of range of melting temperature of EA is comparable to that of MOEA. Therefore, we can conclude that sequences generated by MOEA can successfully

TABLE VII
FITNESS OF THE GENERATED VERTEX SEQUENCES FOR TSP

No.	GC%	Tm	Continuity	Hairpin	H-measure	Similarity
Multi-objective evolutionary algorithm						
0	50	49.0774	0	0	67	41
1	50	46.9393	0	0	44	52
2	50	49.1066	0	0	57	54
3	50	51.3380	0	0	55	54
4	50	49.9252	0	0	57	52
5	50	51.6399	0	0	55	53
6	50	49.4220	0	0	58	44
conventional evolutionary algorithm with sum of fitness values						
0	50	47.607	16	0	66	48
1	50	47.8464	9	0	64	54
2	50	50.6204	0	3	66	57
3	50	50.4628	9	0	62	58
4	50	49.8103	0	3	68	54
5	50	48.3995	0	3	67	51
6	50	50.1205	0	0	61	58

solve the same TSP done by the sequence of EA, since the result of MOEA outperforms the result of EA.

C. Other Problems

Using the MOEA described in this paper, we also designed the sequences for a resolution refutation [47] and version space learning with a lab-on-a-chip technology [48]. These sequences have been also verified by laboratory experiments.

VI. Conclusion

We presented a multiobjective evolutionary algorithm (MOEA) to solve a real-world DNA sequence design problem. The DNA sequence design problem is formulated as a multiobjective optimization task and solved by a controlled elitist MOEA with constrained tournament selection. This method is implemented as a DNA sequence design software called NACST/Seq. The rationale for this approach is based on our analysis of the problem. The landscape of DNA sequence optimization shows many local optima and little gradient information. It depicts the conflicting relationship between fitness objectives. This supports that MOEAs are actually a good candidate for DNA sequence optimization. The controlled elitist MOEA proved also useful since we have a large number of fitness objectives in this problem domain. The constrained tournament method turns out to be effective since some fitness objectives are more suitable to constraints, not to objectives.

The DNA sequences designed by NACST/Seq were compared with those designed by other existing sequence design systems. The results show that NACST/Seq can generate better or comparative sequences in all objectives than other systems. The generality of the NACST/Seq system is also proven by its successful application to a wide range of laboratory experiments, including the TSP, molecular theorem proving, and molecular version-space learning. Experimental results support that NACST/Seq is a useful tool for sequence design in DNA computing. From the evolutionary computational point of view, the DNA sequence design problem exemplifies a new practical application of MOEA that demonstrates its usefulness.

It should be mentioned that there is still room for further improvement of the NACST/Seq system with respect to its speed. The MOEA approach generally takes significant CPU time to find optimal sequence sets, which is caused by the explorative nature of evolutionary computation combined with the difficulty of the design problem we addressed. Introduction of a heuristic such as using a template map and generating a nonrandom initial population would accelerate the convergence speed of MOEA. Also, we try to incorporate more fast MOEA such as ϵ-MOEA and reformulate objective functions for speed-up.

Finally, as mentioned in Section III, NACST/Seq is a subpart of NACST, which is *in silico* platform for DNA computing. Using the NACST system, users can design DNA sequences (NACST/Seq), simulate laboratory experiments (NACST/Sim), and analyze the results (NACST/Report and NACST/Plot). This integrated system can accelerate the development of DNA computing algorithms.

Appendix

Basic Definitions

We define an alphabet to consist of each single nucleotide and gab as $\Lambda = \{A, C, G, T, -\}$, where "$-$" denotes a gab. Also, an alphabet consists of only single nucleotide can be defined as $\Lambda_{nb} = \{A, C, G, T\}$. Then, the set of all DNA sequences is denoted as Λ^*_{nb}. Let $a, b \in \Lambda$ and $x, y \in \Lambda^*$. The length of sequence x is denoted as $|x|$, and x_i $(1 \leq i \leq |x|)$ means ith nucleotide from $5'$-end of sequence x. A set of n sequences with the same length l is denoted by Σ, where ith member of Σ

is denoted as Σ_i. $\bar{a}$ is the complementary base of a, and other basic definitions are as follows:

$$bp(a,b) = \begin{cases} 1, & a = \bar{b} \\ 0, & \text{otherwise} \end{cases}$$

$$eq(a,b) = \begin{cases} 1, & a = b \\ 0, & \text{otherwise} \end{cases}$$

$$T(i,j) = \begin{cases} i, & i > j \\ 0, & \text{otherwise} \end{cases}.$$

For a given sequence $x \in \Lambda^*$, the number of nonblank nucleotides is defined as

$$\text{length}_{nb}(x) = \sum_{i=1}^{|x|} nb(x_i)$$

where

$$nb(a) = \begin{cases} 1, & a \in \Lambda_{nb} \\ 0, & \text{otherwise} \end{cases}$$

and a shift of a sequence x by i bases is denoted as follows:

$$\text{shift}(x,i) = \begin{cases} (-)^i x_1 \cdots x_{l-i}, & i \geq 0 \\ x_{i+1} \cdots x_l (-)^i, & i < 0 \end{cases}.$$

A. H-Measure

H-measure for the given set of sequences Σ is defined as follows:

$$f_{\text{H-measure}}(\Sigma) = \sum_{i=1}^{n} \sum_{j=1}^{n} \text{H-measure}(\Sigma_i, \Sigma_j)$$

where Σ_i and Σ_j are anti-parallel to each other. H-measure(x,y) is divided by two terms. One term is for the overall complementarity and the other is the penalty term for the continuous complementary region. Formally, H-measure(x,y) is defined as follows:

$$\text{H-measure}(x,y) = \text{Max}_{g,i}\big(h_{\text{dis}}(x, \text{shift}(y(-)^g y, i)) + h_{\text{con}}(x, \text{shift}(y(-)^g y, i))\big)$$

where

$$0 \leq g \leq l-3, \qquad |i| \leq l-1$$

$$h_{\text{dis}}(x,y) = T\left(\sum_{i=1}^{l} bp(x_i, y_i), \text{H}_{\text{dis}} \times \text{length}_{nb}(y)\right)$$

$$h_{\text{con}}(x,y) = \sum_{i=1}^{l} T(cbp(x,y,i), \text{H}_{con}).$$

H_{dis} is a real-value between 0 and 1, and H_{con} is an integer between 1 and l. Both values are set by user, and $cbp(x,y,i)$ means the length of continuous basepairing starting from ith base of sequence x, as shown in the first equation at the bottom of the page.

B. Similarity

Similarity is calculated as

$$f_{\text{Similarity}}(\Sigma) = \sum_{i=1}^{n} \sum_{j=1}^{n} \text{Similarity}(\Sigma_i, \Sigma_j)$$

where Σ_i and Σ_j are parallel each other. Similarity(x,y) is also divided into two terms. One term is for the overall discrete similarity and the other is the penalty term for the continuous common subsequence

$$\text{Similarity}(x,y) = \text{Max}_{g,i}\big(s_{\text{dis}}(x, \text{shift}(y(-)^g y, i)) + s_{\text{con}}(x, \text{shift}(y(-)^g y, i))\big)$$

where

$$0 \leq g \leq l-3, \qquad |i| \leq l-1$$

$$s_{\text{dis}}(x,y) = T\left(\sum_{i=1}^{l} eq(x_i, y_i), S_{\text{dis}} \times \text{length}_{nb}(y)\right)$$

$$s_{\text{con}}(x,y) = \sum_{i=1}^{l} T(ceq(x,y,i), S_{\text{con}}).$$

S_{dis} is a real value between 0 and 1, and S_{con} is an integer between 1 and l. Both values are given by user, and $ceq(x,y,i)$ means the length of the continuous common subsequence starting from ith base of sequence x, as shown in the second equation at the bottom of the page.

C. Hairpin

We assume that a hairpin has at least $R_{\min}$ bases as a loop and at least $P_{\min}$ base pairs as a stem. In this fitness function, we calculate the penalty for formation of hairpin with various size at every position in the sequence. We consider a hairpin with r-base loop and p-base pairs stem to be formed at position i in the sequence x, if more than half of bases in the subsequence $x_{i-p} \cdots x_i$ hybridize to the subsequence $x_{i+r} \cdots x_{i+r+p}$. We define the number of matches in these subsequences as the penalty for this hairpin. The formal description of fitness is shown in the first equation at the top of the next page, where $pinlen(p,r,i) = \min(p+i, l-r-i-p)$ denotes maximum number of possible basepairs when a hairpin is formed at center $p+i+r/2$, $\min(i,j)$ takes the smaller one.

$$cbp(x,y,i) = \begin{cases} c, & \text{if } {}^{\exists}c, \text{ s.t. } bp(x_i,y_i) = 0,\, bp(x_{i+j}, y_{i+j}) = 1 \text{ for } 1 \leq j \leq c,\, bp(x_{i+c+1}, y_{i+c+1}) = 0 \\ 0, & \text{otherwise} \end{cases}$$

$$ceq(x,y,i) = \begin{cases} c, & \text{if } {}^{\exists}c, \text{ s.t. } eq(x_i,y_i) = 0,\, eq(x_{i+j}, y_{i+j}) = 1 \text{ for } 1 \leq j \leq c,\, eq(x_{i+c+1}, y_{i+c+1}) = 0 \\ 0, & \text{otherwise} \end{cases}$$

$$f_{\text{Hairpin}}(\Sigma) = \sum_{i=1}^{n} \text{Hairpin}(\Sigma_i)$$

$$\text{Hairpin}(x) = \sum_{p=P_{\min}}^{(l-R_{\min})/2} \sum_{r=R_{\min}}^{l-2p} \sum_{i=1}^{l-2p-r} T\left(\sum_{j=1}^{pinlen(p,r,i)} bp(x_{p+i+j}, x_{p+i+r+j}), \frac{pinlen(p,r,i)}{2} \right)$$

$$\text{Continuity}(x) = \sum_{i=1}^{l-t+1} \sum_{a \in > \Lambda_{nb}} T(c_a(x,i),t)^2$$

$$c_a(x,i) = \begin{cases} o, & \text{if } {}^{\exists}o, \text{ s.t. } x_i \neq a, x_{i+j} = a \text{ for } 1 \leq j \leq o, x_{i+o+1} \neq a \\ 0, & \text{otherwise} \end{cases}$$

$$Tm(x) = \begin{cases} \frac{\Delta H^o}{\Delta S^o + R\ln(\frac{|C_T|}{4})}, & \text{if NN method,} \\ 81.5 + 16.6 \times \log_{10}\left(\frac{[\text{salt}]}{(1.0+0.7\times[\text{salt}])}\right) + 41 \times \text{GC}(x) - \frac{500}{|x|}, & \text{if GC ratio method} \end{cases}$$

D. Continuity

Continuity for a set of sequences Σ is defined as follows:

$$f_{\text{Continuity}}(\Sigma) = \sum_{i=1}^{n} \text{Continuity}(\Sigma_i)$$

where $\text{Continuity}(x)$ is shown in the second equation at the top of the page.

E. Melting Temperature

Let $\text{GC}(x)$ denotes the percentage of bases G and C in sequence x. Then, melting temperature ($Tm(x)$), is like the third equation at the top of the page, where x is DNA sequence, [salt] is salt concentration, R is gas constant, and $|C_T|$ is total sequence concentration.

ACKNOWLEDGMENT

The authors would like to thank ICT at Seoul National University who provided research facilities for this study.

REFERENCES

[1] L. M. Adleman, "Molecular computation of solutions to combinatorial problems," *Science*, vol. 266, pp. 1021–1024, 1994.

[2] C. C. Maley, "DNA computation: Theory, practice, and prospects," *Evol. Comput.*, vol. 6, no. 3, pp. 201–229, 1998.

[3] M. H. Garzon and R. J. Deaton, "Biomolecular computing and programming," *IEEE Trans. Evol. Comput.*, vol. 3, pp. 236–250, Sep. 1999.

[4] J. H. Reif, "The emergence of the discipline of biomolecular computation in the US," *New Generation Comput.*, vol. 20, no. 3, pp. 217–236, 2002.

[5] M. Amos, G. Paun, G. Rozenberg, and A. Salomaa, "Topics in the theory of DNA computing," *Theor. Comput. Sci.*, vol. 287, pp. 3–38, 2002.

[6] A. Brenneman and A. Condon, "Strand design for biomolecular computation," *Theor. Comput. Sci.*, vol. 287, pp. 39–58, 2002.

[7] J. Liu and H. Iba, "Selecting informative genes using a multiobjective evolutionary algorithm," *Proc. 2002 Congr. Evol. Comput.*, pp. 297–302, 2002.

[8] A. R. Rebby and K. Deb, "Identification of multiple gene subsets using multiobjective evolutionary algorithms," *Lecture Notes in Computer Science*, vol. 2632, pp. 623–637, 2003.

[9] A. J. Hartemink, D. K. Gifford, and J. Khodor, "Automated constraint-based nucleotide sequence selection for DNA computation," in *Proc. 4th DIMACS Workshop DNA Based Comput.*, 1998, pp. 227–235.

[10] R. Penchovsky and J. Ackermann, "DNA library design for molecular computation," *J. Comput. Bio.*, vol. 10, no. 2, pp. 215–229, 2003.

[11] A. G. Frutos, A. J. Thiel, A. E. Condon, L. M. Smith, and R. M. Corn, "DNA computing at surfaces: Four base mismatch word design," in *Proc. 3rd DIMACS Workshop DNA Based Comput.*, 1997, p. 238.

[12] M. Arita and S. Kobayashi, "DNA sequence design using templates," *New Generation Comput.*, vol. 20, pp. 263–277, 2002.

[13] U. Feldkamp, S. Saghafi, W. Banzhaf, and H. Rauhe, "DNA sequence generator–A program for the construction of DNA sequences," in *Proc. 7th Int. Workshop DNA Based Comput.*, 2001, pp. 179–188.

[14] F. Tanaka, M. Nakatsugawa, M. Yamamoto, T. Shiba, and A. Ohuchi, "Developing support system for sequence design in DNA computing," in *Proc. 7th Int. Workshop DNA Based Comput.*, 2001, pp. 340–349.

[15] A. Marathe, A. E. Condon, and R. M. Corn, "On combinatorial DNA word design," in *Proc. 5th DIMACS Workshop DNA Based Comput.*, 1999, pp. 75–89.

[16] R. Deaton, J. Chen, H. Bi, M. Garzon, H. Rubin, and D. H. Wood, "A PCR-based protocol for *in vitro* selection of noncrosshybridizing olgionucleotides," in *Proc. 8th Int. Workshop DNA Based Comput.*, 2002, pp. 196–204.

[17] R. Deaton, J. Chen, H. Bi, and J. A. Rose, "A software tool for generating noncrosshybridization libraries of DNA oligonucleotides," in *Proc. 8th Int. Workshop DNA Based Comput.*, 2002, pp. 252–261.

[18] C. E. Heitsch, A. E. Condon, and H. H. Hoos, "From RNA secondary structure to coding theory: A combinatorial approach," in *Proc. 8th Int. Workshop DNA Based Comput.*, 2002, pp. 215–228.

[19] R. Deaton, M. Garzon, R. C. Murphy, J. A. Rose, D. R. Franceschetti, and S. E. Stevens, Jr., "Reliability and efficiency of a DNA-based computation," *Phy. Rev. Lett.*, vol. 80, no. 2, pp. 417–420, 1998.

[20] B.-T Zhang and S.-Y Shin, "Molecular algorithms for efficient and reliable DNA computing," in *Proc. Genetic Program. (GP)*, 1998, pp. 735–742.

[21] M. Arita, A. Nishikawa, M. Hagiya, K. Komiya, H. Gouzu, and K. Sakamoto, "Improving sequence design for DNA computing," in *Proc. Genetic Evol. Comput. Conf. (GECCO)*, 2000, pp. 875–882.

[22] A. J. Ruben, S. J. Freeland, and L. Landweber, "PUNCH: An evolutionary algorithm for optimizing bit set selection," in *Proc. 7th Int. Workshop DNA Based Comput.*, 2001, pp. 260–270.

[23] S.-Y Shin, D.-M Kim, I.-H. Lee, and B.-T Zhang, "Evolutionary sequence generation for reliable DNA computing," in *Proc. Congr. Evol. Comput. (CEC)*, 2002, pp. 79–84.

[24] M. Andronescu, D. Dees, L. Slaybaugh, Y. Zhao, A. Condon, B. Cohen, and S. Skiena, "Algorithms for testing that DNA word designs avoid unwanted secondary structure," in *Proc. 8th Int. Workshop DNA Based Comput.*, 2002, pp. 182–195.

[25] A. G. Frutos, Q. Liu, A. J. Thiel, A. M. W. Sanner, A. E. Condon, L. M. Simith, and R. M. Corn, "Demonstration of a word design strategy for DNA computing on surfaces," *Nucleic Acids Res.*, vol. 25, no. 23, pp. 4748–4757, 1997.

[26] D. Faulhammer, A. R. Cukras, R. J. Lipton, and L. F. Landweber, "Molecular computation: RNA solutions to chess problems," in *Proc. Natl. Acad. Sci. U.S.A.*, vol. 97, 2000, pp. 1385–1389.

[27] D. C. Tuplan, H. Hoose, and A. Condon, "Stochastic local search algorithms for DNA word design," in *Proc. 8th Int. Workshop DNA Based Comput.*, 2002, pp. 229–241.

[28] M. Garzon, P. Neathery, R. Deaton, R. C. Murphy, D. R. Franceschetti, and S. E. Stevens, Jr., "A new metric for DNA computing," in *Proc. Genetic Program. (GP)*, 1997, pp. 472–478.

[29] D. H. Mathews, J. Sabina, M. Zuker, and D. H. Turner, "Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure," *J. Mol. Bio.*, vol. 288, pp. 911–940, 1999.

[30] I.-H. Lee, J.-Y. Park, H.-M. Jang, Y.-G. Chai, and B.-T. Zhang, "DNA implementation of theorem proving with resolution refutation in propositional logic," in *Proc. 8th Int. Workshop DNA Based Comput.*, 2002, pp. 156–167.

[31] S.-Y Shin, H.-Y Jang, and B.-T Zhang, *Thermodynamic Prediction of DNA/DNA Bimolecular Two-State Hybridization Chain Reaction in Solution*. Seoul, Korea: Biointell. Lab., School Comput. Sci. Eng., Seoul Nat. Univ., 2004.

[32] J. Santa Lucia, Jr. and D. Hicks, "The thermodynamics of DNA structural motifs," *Annu. Rev. Biophy. Biomolecular Structure*, vol. 33, pp. 415–440, 2004.

[33] R. B. Wallace, J. Shaffer, R. F. Murphy, J. Bonner, T. Hirose, and K. Itakura, "Hybridization of synthetic oligodeoxyribonucleotides to phi chi 174 DNA: The effect of single base pair mismatch," *Nucleic Acids Res.*, vol. 6, no. 11, pp. 3543–3557, 1979.

[34] J. G. Wetmur, "DNA probes: Applications of the principles of nucleic acid hybridization," *Critical Rev. Biochem. Molecular Bio.*, vol. 26, pp. 227–259, 1991.

[35] J. Santa Lucia, Jr., "A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics," in *Proc. Nat. Acad. Sci. U.S.A.*, vol. 95, 1998, pp. 1460–1465.

[36] D. A. Van Veldhuizen and G. B. Lamont, "Multiobjective evolutionary algorithms: Analyzing the state-of-the-art," *Evol. Comput.*, vol. 8, no. 2, pp. 125–147, 2000.

[37] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2000.

[38] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, Apr. 2003.

[39] K. Deb, A. Pratap, and T. Meyarivan, "Controlled elitist nondominated sorting genetic algorithms for better convergence," in *Proc. 1st Int. Conf. Evol. Multicriterion Opt. (EMO)*, 2001, pp. 284–298.

[40] V. Khare, X. Yao, and K. Deb, "Performance scaling of multiobjective evolutionary algorithms," in *Proc. 2nd Int. Conf. Evol. Multicriterion Opt.*, 2003, pp. 376–390.

[41] K. Deb, *Multiobjective Optimization using Evolutionary Algorithms*. New York: Wiley, 2001.

[42] I.-H. Lee, S.-Y Shin, and B.-T. B.-T. Zhang, "DNA sequence optimization using contrained multiobjective evolutionary algorithm," in *Proc. Congr. Evol. Comput. (CEC)*, 2003, pp. 2270–2276.

[43] R. Deaton, R. C. Murphy, M. Garzon, D. R. Franceschetti, and S. E. Stevens, Jr., "Good encodings for DNA-based solutions to combinatorial problems," in *Proc. 2nd Annu. Meeting DNA Based Comput.*, 1996, pp. 247–258.

[44] F. Tanaka, M. Nakatsugawa, M. Yamamoto, T. Shiba, and A. Ohuchi, "Toward a general-purpose sequence design system in DNA computing," in *Proc. Congr. Evol. Comput. (CEC)*, 2002, pp. 73–78.

[45] J. Y. Lee, S.-Y. Shin, S. J. Augh, T. H. Park, and B.-T Zhang, "Temperature gradient-based DNA computing for graph problems with weighted edges," in *Proc. 8th Int. Workshop DNA Based Comput.*, 2002, pp. 73–84.

[46] J. Y. Lee, S.-Y. Shin, T. H. Park, and B.-T. Zhang, "Solving traveling salesman problems with DNA molecules encoding numerical values," *BioSystems*, vol. 78, pp. 39–47, 2004.

[47] I.-H. Lee, J.-Y. Park, Y.-G. Chai, and B.-T. B.-T. Zhang, "RCA-based detection methods for resolution refutation," in *Proc. 9th Int. Workshop DNA Based Comput.*, 2004, pp. 32–36.

[48] H.-W. Lim, H.-M. Jang, S.-M. Ha, Y.-G. Chai, S.-I. Yoo, and B.-T. Zhang, "A lab-on-a-chip module for bead separation in DNA-based concept learning," in *Proc. 9th Int. Workshop DNA Based Comput.*, 2004, pp. 1–10.

Soo-Yong Shin received the B.S. and M.S. degrees in computer engineering from Seoul National University (SNU), Seoul, Korea, in 1998 and 2000, respectively. He is currently working towards the Ph.D. degree at the School of Computer Science and Engineering, SNU.

He has been a visiting student at the Computer Science and Artificial Intelligence Laboratory (SCAIL), Massachusetts Institute of Technology (MIT), Cambridge, from March 2004 to August 2004. His research interests include evolutionary computation, probabilistic graphical models, DNA computing, and molecular evolutionary computation.

In-Hee Lee received the B.S. degree in computer engineering from Seoul National University (SNU), Seoul, Korea, in 2001. She is currently working towards the Ph.D. degree at the School of Computer Science and Engineering, SNU.

Her research interests include multiobjective evolutionary computation, DNA computing, and molecular theorem proving methods.

Dongmin Kim received the B.S. degree in computer science from Seoul National University (SNU), Seoul, Korea, in 2002.

He is a Researcher in the Biointelligence Laboratory, School of Computer Science and Engineering, SNU. His research interests include numerical optimization, evolutionary computation, and probabilistic graphical models.

Byoung-Tak Zhang received the B.S. and M.S. degrees in computer science and engineering from Seoul National University (SNU), Seoul, Korea, in 1986 and 1988, respectively, and the Ph.D. degree in computer science from University of Bonn, Bonn, Germany, in 1992.

He is an Associate Professor at the School of Computer Science and Engineering and of the Graduate Programs in Bioinformatics, Brain Science, and Cognitive Science at Seoul National University (SNU), and directs the Biointelligence Laboratory and the Center for Bioinformation Technology (CBIT). Prior to joining SNU, he had been a Research Associate at the German National Research Center for Information Technology (GMD) from 1992 to 1995. He has been a Visiting Professor at the Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology (MIT), Cambridge, from August 2003 to August 2004. His research interests include probabilistic models of learning and evolution, biomolecular/DNA computing, and molecular learning/evolvable machines.

Dr. Zhang serves as an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, *Advances in Natural Computation*, *and Genomics and Informatics*. He is on the Editorial Board of *Genetic Programming and Evolvable Machines and Applied Soft Computing*.