Jurnal Teknologi, 54 (Sains & Kej.) Keluaran Khas, Jan. 2011: 331–342

FUNCTION MINIMIZATION IN DNA SEQUENCE DESIGN BASED ON BINARY PARTICLE SWARM OPTIMIZATION

ZUWAIRIE IBRAHIM[1*], NOOR KHAFIFAH KHALID[2], ISMAIL IBRAHIM[3], LIM KIAN SHENG[4], SALINDA BUYAMIN[5], ZULKIFLI MD. YUSOF[6], & MOHD SAUFEE MUHAMMAD[7]

Abstract. In DNA based computation and DNA nanotechnology, the design of good DNA sequences has turned out to be an elementary problem and one of the most practical and important research topics. Although the design of DNA sequences is dependent on the protocol of biological experiments, it is highly required to establish a method for the systematic design of DNA sequences, which could be applied to various design constraints. Basically, the fitness of DNA sequences can be evaluated using four objective functions, namely, *similarity*, $H_{measure}$, *continuity*, and *hairpin*. In this paper, binary particle swarm optimization (BinPSO) is proposed to minimize those objective functions individually, subjected to two constraints: *melting temperature* and $GC_{content}$. An implementation of the optimization process is presented using 20 particles and the results obtained shows the correctness of PSO computation, where the minimized values for each objective can be achieved.

Keywords: DNA sequence design; binary particle swarm optimization; objective function; constraints; fitness function

Abstrak. Dalam DNA berasaskan pengiraan dan nanoteknologi DNA, reka bentuk jujukan-jujukan DNA baik telah menjadi menjadi satu masalah asas dan salah satu topik-topik penyelidikan penting dan paling praktikal. Walaupun reka bentuk DNA berjujukan adalah bergantung kepada pada protokol eksperimen biologi, ia amat diperlukan bagi mewujudkan satu kaedah untuk reka bentuk bersistem DNA berjujukan, yang boleh digunakan untuk pelbagai reka bentuk kekangan. Pada asasnya, kecergasan DNA berjujukan boleh dinilaikan menggunakan empat fungsi-fungsi objektif, iaitu, *similarity*, $H_{measure}$, *continuity*, dan *hairpin*. Dalam kertas kerja ini, pengoptimuman kerumunan zarah perduaan (BinPSO) dicadangkan untuk meminimumkan fungsi-fungsi objektif tersebut secara individu, tertakluk kepada dua kekangan: *melting temperature* dan $GC_{content}$. Satu perlaksanaan proses pengoptimuman dibentangkan menggunakan 20 zarah-zarah dan keputusan yang diperolehi menunjukkan kebenaran pengiraan PSO, di mana nilai-nilai yang minimum untuk setiap objektif dapat dicapai.

[1-6] Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310, UTM Johor Bahru, Johor Darul Ta'azim, Malaysia

[7] Faculty of Engineering, Universiti Malaysia Sarawak, 94300, Kota Samarahan, Sarawak, Malaysia

* Corresponding author: zuwairie@fke.utm.my

Kata kunci: Reka bentuk jujukan DNA; pengoptimuman kerumunan zarah perduaan; fungsi objektif; kekangan; fungsi kecergasan

1.0 INTRODUCTION

A nucleic acid is a macromolecule composed of chains of monomeric nucleotide. In biochemistry, these molecules carry genetic information or form structures within cells. The most common nucleic acids are deoxyribonucleic acid (DNA) and ribonucleic acid (RNA). DNA, in particular, is universal in living things, as they are found in all cells and viruses. DNA is a polymer, which is strung together from a series of monomers. Monomers, which form the building blocks of nucleic acids, are called nucleotides. Each nucleotide contains a sugar (deoxyribose), a phosphate group, and one of four bases: Adenine (A), Thymine (T), Guanine (G), or Cytosine (C). A single stranded DNA consist a series of nucleotides. The two of single-stranded DNA are held together by hydrogen bonds between pairs of bases, which called duplex or double stranded DNA based on Watson-Crick complement.

DNA has certain unique properties such as self-assembly and self-complementary, which makes it able to save an enormous amount of data and perform massive parallel reactions. With the view of the utilization of such attractive features for computation, DNA computation research field has been instigated [1]. Usually, in DNA computing, the calculation process consists of several chemical reactions, where the successful wet lab experiment depends on DNA sequences used. Thus, DNA sequence design turns out to be one of the approaches to achieve high computation accuracy and become one of the most important research topics in DNA computing.

The necessity of DNA sequence design appears not only in DNA computation, but also in other biotechnology fields, such as the design of DNA chips for mutational analysis and for sequencing [2]. For these approaches, sequences are designed such that each element uniquely hybridizes to its complementary sequence, but not to any other sequence. Due to the differences in experimental requirements, however, it seems impossible to establish an all-purpose library of sequences that effectively caters for the requirements of all laboratory experiments [3]. Since the design of DNA sequences is dependent on the protocol of biological experiments, a method for the systematically design of DNA sequences is highly required [4].

The ability of DNA computer to perform calculations using specific biochemical reactions between different DNA strands by Watson–Crick complementary base pairing, affords a number of useful properties such as massive parallelism and a huge memory capacity [5]. However, due to the technological difficulty of biochemical experiment, the *in vitro* reactions may

result in incorrect or undesirable computation. Sometimes, DNA computers fail to generate identical results for the same problem and algorithm. Furthermore, some DNA strands or sequences could be wasted because of the undesirable reactions. To overcome these drawbacks, much work has focused on improving the reliability (correctness) and efficiency (economy) of DNA computing [6].

In DNA computing, perfect hybridization between a sequence and its base-pairing complement is important to retrieve the information stored in the sequences and to operate the computation processes. For this reason, the desired set of good DNA sequences, which have a stable duplex with their complement, are highly required. It is also important to ensure that two sequences are not complements of one another.

Various kinds of methods and strategies have been proposed to date to obtain good DNA sequences. These methods are exhaustive search method [7], random search algorithm [8], simulated annealing [9], dynamic programming approach [10], graph method [11], template-map strategy [5, 12], genetic algorithms [13-14], and multi-objective evolutionary optimization [15].

The objective of the DNA sequence design problem is basically to obtain a set of DNA sequences where each sequence is unique or cannot be hybridized with other sequences in the set. In this work, two objective functions, namely $H_{measure}$, and *similarity* are chosen to estimate the uniqueness of each DNA sequence. Moreover, two additional objective functions, which are *hairpin* and *continuity,* are used to prevent the secondary structure of a DNA sequence. In previous work, these four objective functions have been minimized using a standard particle swarm optimization (PSO) algorithm [16]. In this paper, binary PSO (BinPSO) is used to minimize these four objective functions individually. *Melting temperature* and $GC_{content}$ are included as constraints in the BinPSO computation. The formulations for all objectives and constraints can be referred to [17].

2.0 MINIMIZATION OF $H_{measure}$, *Similarity*, *Continuity*, AND *Hairpin* BASED ON BINARY PSO

Particle swarm optimization (PSO) is a population-based stochastic optimization technique developed by Kennedy and Eberhart in 1995 [18]. This method finds an optimal solution by simulating social behaviour of bird flocking. The PSO algorithm consists of a group of individuals named "particles". Each particle is a potential solution to an *n*-dimensional problem. The group can achieve the solution effectively by using the common information of the group and the information owned by the particle itself. The particles change their state by "flying" around in an *n*-dimensional search space based on the velocity updated until a relatively unchanging state has been encountered, or until computational

limitations are exceeded. The binary particle swarm optimization (BinPSO) algorithm, which was also introduced by Kennedy and Eberhart, allows the PSO algorithm to operate in binary problem spaces [19]. It uses the concept of velocity as a probability that a bit (position) takes on one or zero.

In this study, a DNA sequence is represented in binary, where A, C, G, and T, are encoded as 2 bits of 00_2, 01_2, 10_2, and 11_2, respectively. Dimensions represent bits of binary number, thus, 2 dimensions are needed to form one base (1-mer length) of DNA sequence, as illustrated in Figure 1. For instance, to produce 10-mer length sequence, 20 dimensions are needed. In order to find a set of n-sequences with l-mer length, the sequences consists of ($n \times l \times 2$) dimensions.

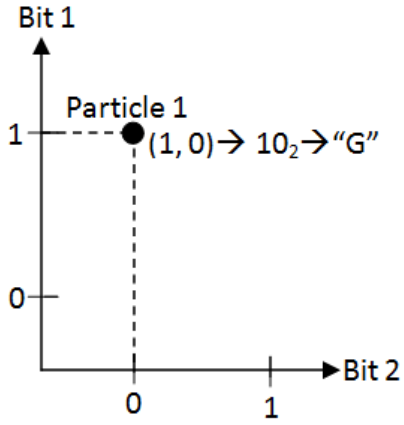


Figure 1 The example of the proposed model, where the position of particle 1 created one base DNA sequence in 2 dimensions binary search space]

To develop a set of 10 DNA sequences with the length of 20-mer, for example, (10 × 20×2) = 400 dimensions should be used in a search space. However, each particle in the search space carries DNA sequence with (10 × 20) = 200-mer length. In this study, 20 particles are employed and randomly initialized in the search space.

The BinPSO algorithm for objective function minimization is shown in Figure 2. The algorithm begins with the initialization of DNA parameters and PSO control parameters. The values of the constraints are 30%-80% for $GC_{content}$ and 50°C-80°C for T_m. PSO control parameters are shown in Table 1. In addition, a decreasing inertia weight is used, where

$$\omega^{iteration} = \omega_{\max} - \left(\frac{\omega_{\max} - \omega_{\min}}{iteration_{\max}} \times iteration \right) \quad (1)$$

A large starting value of ω is used to initially accommodate more exploration, and is dynamically reduced to speed the convergence to the global optimum at the end of the search process [20].

After the fitness or objective functions are calculated, with reference to the original PSO [18], each particle knows its best value so far (*pbest*), velocity, and position. Additionally, each particle knows the best value in its neighbourhoods (*gbest*). A particle modifies its position based on its current velocity and position. The velocity of each particle is calculated using

$$\mathbf{v}_i^{k+1} = \omega \mathbf{v}_i^k + c_1 r_1 (\boldsymbol{pbest}_i - \mathbf{s}_i^k) + c_2 r_2 (\boldsymbol{gbest}^k - \mathbf{s}_i^k) \tag{2}$$

where $\mathbf{v}_i^k$, $\mathbf{v}_i^{k+1}$, and $\mathbf{s}_i^k$, are the velocity vector, modified velocity vector, and positioning vector of particle i at generation k, respectively. $\boldsymbol{pbest}_i^k$ is the best position found by particle i and $\boldsymbol{gbest}^k$ is the best position found by the particle's neighbourhood or the entire swarm. c_1 and c_2 are the cognitive and social coefficients, respectively, used to bias the search of a particle toward its own best experience (*pbest*) and the best experience of the whole swarm (*gbest*). ω is inertia weight, which is employed to control the impact of the previous history of velocities on the current velocity of each particle. The parameter regulates the trade-off between the exploration and exploitation ability of the swarm. Large values of ω facilitate exploration and searching new areas, while small values of ω navigate the particles to more refined search. The velocity equation includes two different random parameters, represented by a variable, r_1 and r_2, to ensure good exploration of the search space and to avoid entrapment in local optima.

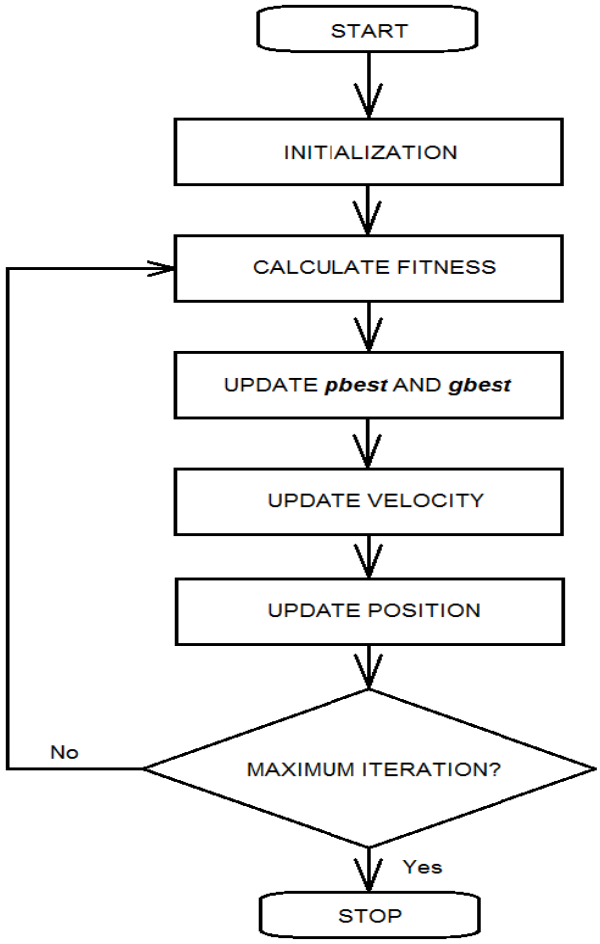


Figure 2 The BinPSO algorithm for objective function minimization

Table 1 The value of PSO control parameters

PARAMETER	VALUE
Cognitive factor, c_1	2
Social factor, c_2	2
Inertia weight, ω	0.9-0.4
Random values: r_1, r_2	[0,1]
No. of particles	20
Max iteration	400

In BinPSO, the modified position vector, $\mathbf{s}_i^{k+1}$, is obtained by the following rule [19]:

$$\boldsymbol{s}_i^{k+1} = \begin{cases} 0 & if \quad r_3 \geq S(\boldsymbol{v}_i^{k+1}) \\ 1 & if \quad r_3 < S(\boldsymbol{v}_i^{k+1}) \end{cases} \tag{3}$$

with $r_3 \sim U(0,1)$ and $S()$ is a sigmoid function for transforming the velocity to the probability constrained to the interval [0.0, 1.0] as follows

$$S(\boldsymbol{v}_i^{k+1}) = \frac{1}{1+e^{-\boldsymbol{v}_i^{k+1}}} \tag{4}$$

where $S(v) \in (0,1)$, $S(0) = 0.5$ and r_3 is a quasi random number selected from a uniform distribution in [0.0, 1.0].

3.0 EXPERIMENTAL RESULTS

The proposed approach has been implemented using Visual Basic 6.0. The algorithm has been executed for 10 times and the average values and standard deviations were calculated. The experiment considered a set of 10 DNA sequences with the length of 20-mer.

Figure 3 and Table 2 show the results from $H_{measure}$ computation. Although PSO algorithm can find the minimum value for $H_{measure}$, the values for other objectives, which are *similarity* and *continuity,* are high, except for *hairpin*. The fitness value of $H_{measure}$ leads to convergence after 340 iterations.

For *similarity* computation, the results are displayed in Figure 4 and Table 3. The minimized value for *similarity* has been obtained; nevertheless the values for other objectives are not minimized. The fitness function of *similarity* leads to convergence after 5 iterations.

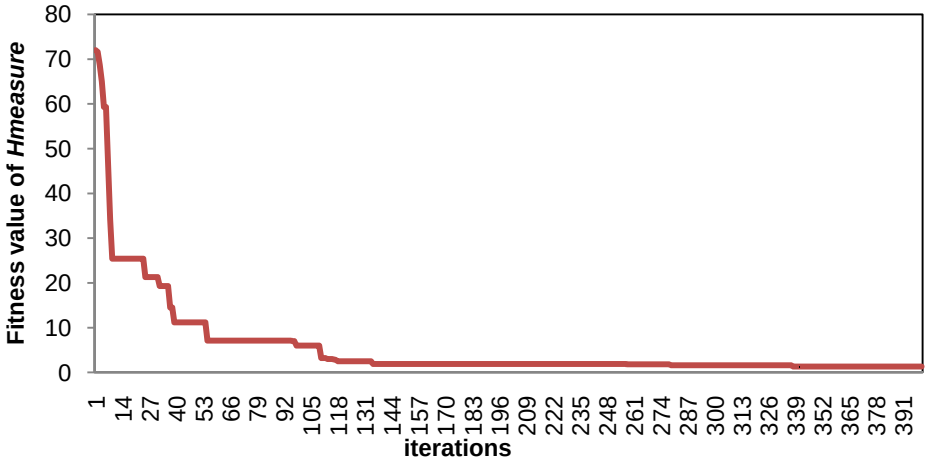
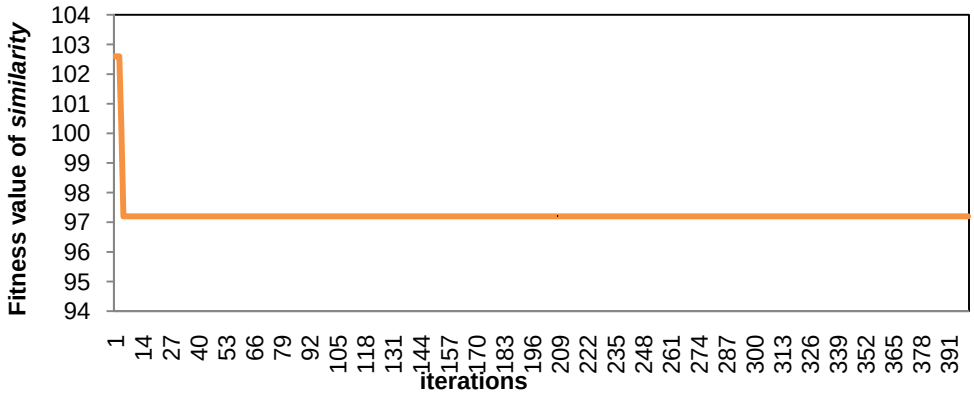


Figure 3 Result and convergence curve of $H_{measure}$.

Table 2 The results obtained to minimize $H_{measure}$ function. Standard deviations are shown in parentheses

DNA SEQUENCES	$H_{measure}$	*Similarity*	*Continuity*	*Hairpin*
CTAAGTCAACCAAGCACACA AAACACATATCGACTGGGAG GACAACGTGGCGACTGTGCC ATTCTGACGCCTTTAAGTTA TCTCCGCAATTAAAAGGAGT CATAGCCCTAGGTCCAGTAA ATAAAAACGAACCAAAACCC GCGGCTGGCAAACAAAATGC CCCATTCAACTACCAGTTTT TCCTCGCCGCAATACGCAAG	4.7 (6.842)	286.08 (80.316)	311.6 (69.069)	3.23 (3.494)



ve

Figure 4 Result and convergence curve of *similarity*

Table 3 The results obtained to minimize *similarity* function. Standard deviations are shown in parentheses

DNA SEQUENCES	$H_{measure}$	*Similarity*	*Continuity*	*Hairpin*
CTCTGCGCTATACACTGTTG TCTTTTCGCTCCTGAAAACG CGCGGAAAAGGTATGGCTCA TGCGTTTTGCCCAGGAAAAG CTCAACACGCGTTTTGGACA CAACGATAGATTCTACATTA GTCGTTCGAGCCACCCAGAC AGTCTGATGTAAAAGTTCCC TCTATTGGGAAGCACGGAGG CGCCGTCAAGTCGGCCCAGG	97.38 (0.3944)	65.47 (20.127)	17.55 (4.455)	70.06 (4.872)

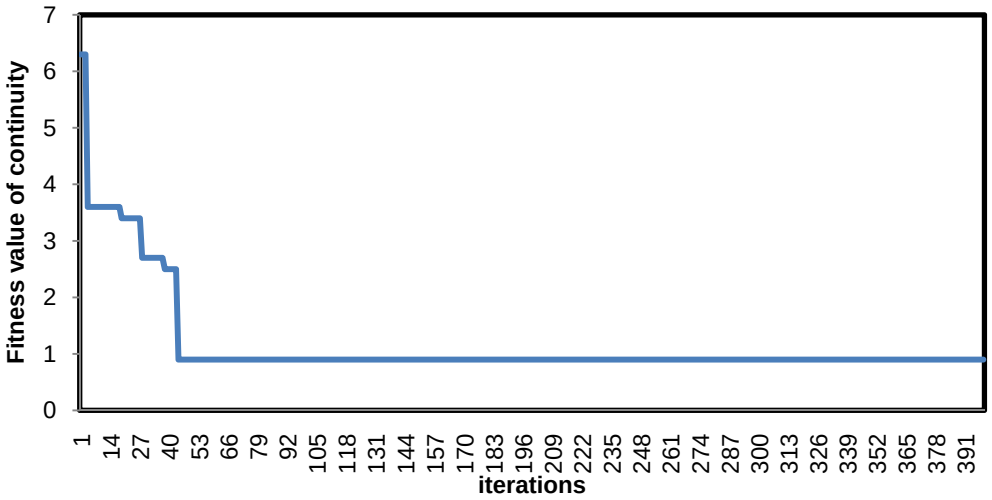


Figure 5 Result and convergence curve of *continuity*

In Figure 5 and Table 4, the sample of the results from *continuity* objective are shown, where the same outcome for previous objectives computation occurred. The particles of PSO algorithm can attain the minimum value for *continuity* objective, but not the minimized value for other objectives. In this computation, the particles converge and the fitness value of *continuity* is satisfied within 40-46 iterations.

Figure 6 and Table 5 show the results from *hairpin* computation, where the fitness value is the lowest value obtained compare to previous computation. Although PSO algorithm can find the minimum value for *hairpin*, the values for other objectives, which are *similarity*, $H_{measure}$, and *continuity*, are high. The particles converge and the fitness value of *hairpin* is satisfied within 310-320 iterations.

Table 4 The results obtained to minimize *continuity* function. Standard deviations are shown in parentheses

DNA SEQUENCES	$H_{measure}$	*Similarity*	*Continuity*	*Hairpin*
TGATAATGGAGCATGACACC GTAGCGGATGATTCTCGCGT ATGGATCTAGCTCATCTTCA GTACCAAGTTGCGCTAGCAA ATATAACATTCACTGTCCAT GGTGCGCCCAGGAACGGAGC CCAACACTCTCCGACAGTCG TTCAGCCAGAAGCTCGCCAT CTGTGACTATGAGTTGGCAC TTACATAGCGTATGCGCATA	75.02 (1.369)	19.44 (3.264)	0.45 (0.45)	107.9 (2.373)

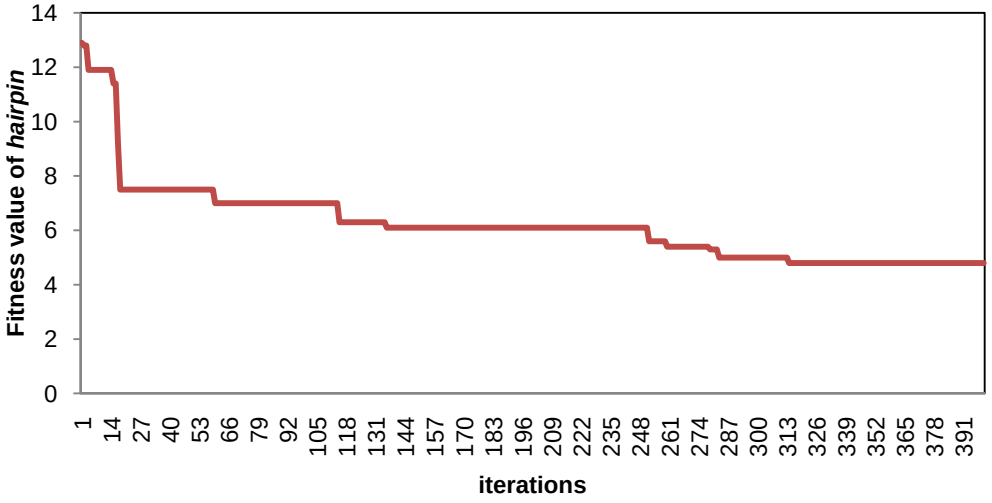


Figure 6 Result and convergence curve of *hairpin*

Table 5 The results obtained to minimize *hairpin* function. Standard deviations are shown in parentheses

DNA SEQUENCES	$H_{measure}$	*Similarity*	*Continuity*	*Hairpin*
CGACTTCTGGCAATGCGAAT TCTTGTGAGTTAGTGTGCTG CGAATAGTTGATCTGGGGAT CAACGGGAGCTAATCGAACT GGTCTCCCTCACTCCTAGAC CATGTGTCGCTCAGGAGCGC TTATTGCGGTTAAATTCTGC CTCTAGGATCTGCATGACTT TTAGGAAAAACGCGCACAGT CGCCATATGCGGGAGGATGT	203.1 (79.414)	39.15 (13.176)	168.84 (72.853)	2.82 (2.033)

4.0 CONCLUSIONS

This paper proposes an application of PSO in DNA sequence design. An approach based on BinPSO is presented to minimize four objective functions individually. These objective functions are H_{measure}, *similarity*, *hairpin*, and *continuity*. GC_{content} and T_{m} constraints were embedded in the computation to ensure that the DNA sequences obtained are within the acceptable range.

From the results, the minimum value of each objective in DNA sequence design can be obtained using the proposed approach. However, DNA sequence design is a multi-objective optimization problem and the objectives should be minimized simultaneously. Therefore, future works will include the solution of multi-objective problem in DNA sequence design using Pareto optimality concept and vector evaluated PSO.

REFERENCES

[1] Arita, M., Nishikawa, A., Hagiya, M., Komiya, K., Gouzu, H. and Sakamoto, K. 2000. Improving Sequence Design for DNA Computing. *Proc. Genetic Evol. Comput. Conf. (GECCO)*. 875-882.

[2] Reece, R. J. 2004. *Analysis of Genes and Genomes*. John Wiley & Sons.

[3] Adleman, L. 1998. Molecular Computation of Solutions to Combinatorial Problems. *Science*. 266:1021-1024.

[4] Kashiwamura, S., Kameda, A., Yamamoto, M. and Ohuchi. A. 2003. Two-step Search for DNA Sequence Design. Proceedings of the 2003 International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 03). 1815-1818.

[5] Arita, M. and Kobayashi. S. 2002. DNA Sequence Design using Templates. *New Generation Comput*. 20: 263-277.

[6] Kobayashi, S. and Kondo. T. 2002. On Template Method for DNA Sequence Design. Preliminary Proceeding of 8th International Meeting on DNA Based Computers. 115-124.

[7] Hartemink, A. J., Gifford, D. K. and Khodor. J. 1998. Automated Constraint Based Nucleotide Sequence Selection for DNA Computation. *Proc. 4th DIMACS Workshop DNA Based Computer*. 227-235.

[8] Penchovsky. R. and Ackermann. J. 2003. DNA Library Design for Molecular Computation. *J. Comput. Bio*. 10(2): 215-229.

[9] Tanaka, F., Naktsugawa, M., Yamamoto, M., Shiba, T. and Ohuchi. A. 2002. Toward a General-Purpose Sequence Design System in DNA Computing. *Proc. Congr. Evil. Comput (CEC)*. 73-78.

[10] Marathe, A., Condon, A. E. and Corn. R. M. 1999. On Combinatorial DNA Word Design. Proceedings of the 5th International Meeting on DNA Based Computers.

[11] Feldkamp, U., Saghafi, S., Banzhaf. W. and Rauhe. H. 2001. DNA Sequence Generator–A Program for the Construction Of DNA Sequences. *Proc. 7th Int. Workshop DNA Based Computer*. 179-188.

[12] Frutos, A. G., Thiel, A. J., Condon, A. E., Smith. L. M. and Corn. R. M. 1997. DNA Computing at Surfaces: Four Base Mismatch Word Designs. *Proc. 3rd DIMACS Workshop DNA Based Computer*. 238.

[13] Deaton, R., Murphy, R. C., Garzon, M. D., Franceschetti. T. S. and Stevens Jr. E. 1996. Good Encodings for DNA–based Solutions to Combinatorial Problems. *Proceedings of the Second Annual Meeting on DNA Based Computers*. 159-171.

[14] Deaton, R., Murphy, R. C., Rose, J. A., Garzon, M. D., Franceschetti. T. and Stevens Jr. S. E. 1996. Genetic Search for Reliable Encodings for DNA–based Computation. *First Conference on Genetic Programming*.

[15] Shin, S.Y., Lee, I.H., Kim D. and Zhang. B.T. 2005. Multi-objective Evolutionary Optimization of DNA Sequences for Reliable DNA Computing. *IEEE Transaction on Evolutionary Computation*. 9(2): 143-158.

[16] Khalid, N. K., Ibrahim, Z., Kurniawan, T. B., Khalid, M., Sarmin. N. H. and Engelbrecht. A. P. 2009. Function Minimization in DNA Sequence Design Based on Continuous Particle Swarm Optimization. *Innovative Computing, Information and Control Express Letters (ICIC Express Letters)*. 3(1): 27-32.

[17] Kurniawan, T. B., Khalid, N. K., Ibrahim, Z., Khalid. M. and Middendorf. M. 2008. An Ant Colony System for DNA Sequence Design Based On Thermodynamics. Proceedings of the Fourth IASTED International Conference Advances in Computer Science and Technology *(ACST 2008).* 144-149.

[18] Kennedy. J. and Eberhart. R.C. 1995. Particle Swarm Optimization. Proceeding of IEEE International Conference on Neural Networks. 1942-1948.

[19] Kennedy. J. and Eberhart. R. C. 1997. A discrete binary version of the particle swarm algorithm. *Proc. Conf. Systems.*Piscataway: *NJ.* 4104-4108.

[20] Eberhart. R.C. and Shi. Y. 2000. Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization. *Proceedings of IEEE Congress Evolutionary Computation.* 84-88.