International Journal of Innovative Computing, Information and Control
Volume **8**, Number **5(B)**, May **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 3441–3450

DNA SEQUENCE DESIGN FOR DNA COMPUTATION BASED ON BINARY PARTICLE SWARM OPTIMIZATION

Zuwairie Ibrahim[1], Noor Khafifah Khalid[2], Salinda Buyamin[2]
Ismail Ibrahim[2], Jameel Abdulla Ahmed Mukred[2], Zulkifli Md. Yusof[2]
Mohd Saberi Mohamad[3], N. Mokhtar[4], Muhammad Faiz Mohamed Saaid[4]
and Andries Engelbrecht[5]

[1]Faculty of Electrical and Electronic Engineering
Universiti Malaysia Pahang
26600 Pekan, Pahang, Malaysia
zuwairie@fke.utm.my

[2]Faculty of Electrical Engineering
[3]Faculty of Computer Science and Information Systems
Universiti Teknologi Malaysia
81310 Skudai, Johor, Malaysia
{ salinda; jameel; zmdyusof }@fke.utm.my
ifah.khalid@gmail.com; ismail_xyz@yahoo.com; saberi@utm.my

[4]Faculty of Engineering
University of Malaya
50603 Kuala Lumpur, Malaysia
{ norrimamokhtar; mfms }@um.edu.my

[5]Department of Computer Science
University of Pretoria
Pretoria 0028, South Africa
engel@cs.up.ac.za

Received November 2010; revised March 2011

Abstract. *Deoxyribonucleic Acid (DNA) has certain unique properties such as self-assembly and self-complementary in hybridization, which are important in many DNA-based technologies. DNA computing, for example, uses these properties to realize a computation in vitro, which consists of several chemical reactions. Other DNA-based technologies such as DNA-based nanotechnology and polymerase chain reaction also depend on hybridization to assemble nanostructure and to amplify DNA templates, respectively. Hybridization of DNA can be controlled by properly designing DNA sequences. In this paper, sequences are designed such that each sequence uniquely hybridizes to its complementary sequence, but not to any other sequences. Objective functions involved are similarity, $H_{measure}$, continuity, and hairpin. Binary particle swarm optimization (BinPSO) is employed to minimize those objectives subjected to two constraints: melting temperature and $GC_{content}$. It is found that BinPSO can provide a set of good DNA sequences, better than basic PSO algorithm in terms of aggregated fitness value.*

Keywords: Particle swarm optimization, Binary PSO, DNA sequence design, Optimization

1. **Introduction.** Deoxyribonucleic acid (DNA) is a nucleic acid that contains the genetic instructions used in the development and functioning of all known living organisms and some viruses. However, DNA molecules are presently used in many areas far beyond its traditional function. In 1994 for example, a DNA-based computation has been introduced by Adleman [1] to solve a Hamiltonian path problem (HPP). The success of the

DNA-based computation 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.

Various kinds of methods and strategies have been proposed to date to obtain good DNA set of DNA sequences. These methods are exhaustive search method [2], random search algorithm [3], simulated annealing [4], dynamic programming approach [5], graph method [6], template-map strategy [7,8], genetic algorithms [9,10], and multi-objective evolutionary optimization [11]. Cui et al. have designed DNA sequences using PSO in which the DNA sequences are connected one by one in the same direction to form a long DNA sequence [12]. On the other hand, Zhou et al. have proposed a multi-objective PSO (MOPSO) to design a set of good DNA sequences [13].

In previous work, we employed basic PSO algorithm to solve DNA sequence design problem [14]. Continuous search space is utilized in which position in each dimension indicates the possible DNA sequence. Compared with [12] and [13], different representation is used such that A = 00, C = 01, G = 10, and T = 11. This algorithm has a certain limitation. Since the search space is in continuous value, the positions are in floating points. However, the representation of DNA sequences is discrete. Therefore, the floating points need to be eliminated by approximating the floating values.

In this paper, DNA sequences are designed based on binary particle swarm optimization (BinPSO) [15]. The representation of DNA bases is taken from [14]. Each dimension in BinPSO has 2 possible binary values, 0 or 1. Thus, a base of DNA sequences is represented by 2 bits of binary number.

2. Objectives and Constraints in DNA Sequence Design.

Given several short single-stranded DNAs in a test tube, if the temperature is decreased, these DNAs tend to hybridize to other molecules in the tube subjected to Watson-Crick complementary. Intuitively, the main objective of the DNA sequence design is to avoid this hybridization. The probability of a DNA molecule to hybridize with itself and other DNAs can be measured using $H_{measure}$, *similarity*, *hairpin*, and *continuity*. These objectives are subjected to $GC_{content}$ and melting temperature constraints.

Generally, given a number of objective functions and constraints, formal objective of the DNA sequence design is to produce a set of good DNA sequences with minimized values of these objective functions. If this condition is achieved, it can be said that the sequences in the set are unique and cannot hybridize to each another. In this paper, two objective functions, namely $H_{measure}$ and *similarity* are chosen to estimate the uniqueness of each DNA sequence. Another two additional objective functions, *hairpin* and *continuity*, are used to prevent the secondary structure of a DNA sequence. Hence, the DNA sequence design is formulated as follows:

$$\min f_{DNA} = f_1 + f_2 + f_3 + f_4 \tag{1}$$

subjected to T_m and $GC_{content}$ constraints, where f_1, f_2, f_3, and f_4 are the objective function for $H_{measure}$, *similarity*, *hairpin*, and *continuity*. The formulations for all objectives and constraints have been published in [16].

3. Particle Swarm Optimization.

Particle swarm optimization (PSO) is a population-based stochastic optimization technique developed by Kennedy and Eberhart in 1995 [17]. In past years, PSO has been successfully applied in both in computer science [18] and engineering [19]. In original PSO algorithm, each particle knows its best value so far (pbest), velocity, and position. Additionally, each particle knows the best value in its swarm (gbest). A particle modifies its position based on its current velocity and position. The velocity of each particle is calculated using

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

where $\boldsymbol{v}_i^k$, $\boldsymbol{v}_i^{k+1}$, and $\boldsymbol{s}_i^k$, 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 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 called inertia weight, which is employed to control the impact of the previous history of velocities on the current velocity of each particle. The ω 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, sampled from uniform distribution between 0 and 1, i.e., $\sim \mathrm{U}(0,1)^n$. The modified position vector, $\boldsymbol{s}_i^{k+1}$, is obtained using

$$\boldsymbol{s}_i^{k+1} = \boldsymbol{s}_i^k + \boldsymbol{v}_i^{k+1} \tag{3}$$

The binary particle swarm optimization (BinPSO) algorithm has been introduced to allow the PSO algorithm to operate in binary problem spaces [15]. It uses the concept of velocity converted to a probability that a bit (position) takes on a value of 1 or 0. In Binary PSO, Equation (2) of updating a velocity remains unchanged, but Equation (3) of updating a position is re-defined by the following rule [12]:

$$\boldsymbol{s}_{ij}^{k+1} = \begin{cases} 0 & \text{if } r_3 \geq S(\boldsymbol{v}_{ij}^{k+1}) \\ 1 & \text{if } r_3 < S(\boldsymbol{v}_{ij}^{k+1}) \end{cases} \tag{4}$$

with $r_3 \sim \mathrm{U}(0,1)$ and S is the sigmoid function used to transform the velocity to a probability constrained to the interval $[0.0, 1.0]$ as follows:

$$S(\boldsymbol{v}_{ij}^{k+1}) = \frac{1}{1 + e^{-\boldsymbol{v}_{ij}^{k+1}}} \tag{5}$$

4. Optimization of DNA Sequence Based on BinPSO.

In order to design a set of DNA sequences based on Bin PSO, a sequence is represented as binary vector, where A, C, G, and T, are encoded as 00_2, 01_2, 10_2, and 11_2, respectively. In order to find a set of n-sequences with l-mer length, a search space of $(n \times l \times 2)$ dimensions is required.

In this study, 20 particles are employed and randomly initialized in the search space. The values of the constraints are $30\% < GC_{content} < 80\%$ and $50°\mathrm{C} < T_m < 80°\mathrm{C}$. The T_m was computed based on the nearest-neighbor (NN) method [20]. Table 1 summarizes the values of PSO control parameters used in the experiments. In this study, a decreasing inertia weight is used. A large starting value of ω is used to initially accommodate more exploration, and is dynamically reduced to speed up the convergence to the global optimum at the end of the search process [21]. The values of cognitive and social factors, as stated in Table 1, are chosen as these values have been found to work well in many studies [22].

5. Result and Discussion.

The results of the proposed approach, BinPSO, are compared with existing approaches, taken from Deaton et al. [9], Tanaka et al. [4], Guangzhou et al. [12], and Zhao et al. [13]. For each comparison, 100 runs have been performed by BinPSO and the average performance is calculated in terms of the mean value and the standard deviation of the fitness values.

Table 1. The value of PSO control parameters

Parameter	Value
Cognitive and social factors, c_1 and c_2	2.0
Inertia weight, ω	0.9-0.4
Random values: r_1, r_2	$[0, 1]$
No. of particles	20
Max iteration	1000

The BinPSO method is first compared with results given in [9], which were obtained using a genetic algorithm. The method produced 7 good sequences with the length of 20-mer. Results of the two algorithms are compared in Table 2 and Figure 1. BinPSO reached lower values in the total objectives, compared with the GA. The sequences generated by BinPSO surpassed the sequences from the GA in three objectives. Sequences designed by BinPSO show lower values of *hairpin, continuity*, and *similarity,* while sequences from Deaton et al. are better than BinPSO in the $H_{measure}$ objective.

Table 3 and Figure 2 contain the comparison between the results of the BinPSO with the DNA sequences generated by simulated annealing (SA) [4]. A set of sequences, which consists of 14 DNA sequences of 20-mer length, has been designed by simulated annealing.

Table 2. Comparison of the sequences in [9] and the sequences generated by BinPSO

Sequences	C[1]	Ha[2]	Hm[3]	S[4]	Total
GA [10]					
Fitness value	11.71	0.57	20.43	13.14	45.85
Standard Deviation	14.80	1.51	7.14	7.43	–
BinPSO					
Average	1.29	0.00	20.57	11.29	33.15
Standard Deviation	3.019	0.326	4.346	2.140	–

[1]C, [2]Ha, [3]Hm, and [4]S are *continuity*, *hairpin*, $H_{measure}$, and *similarity* objectives values, respectively.

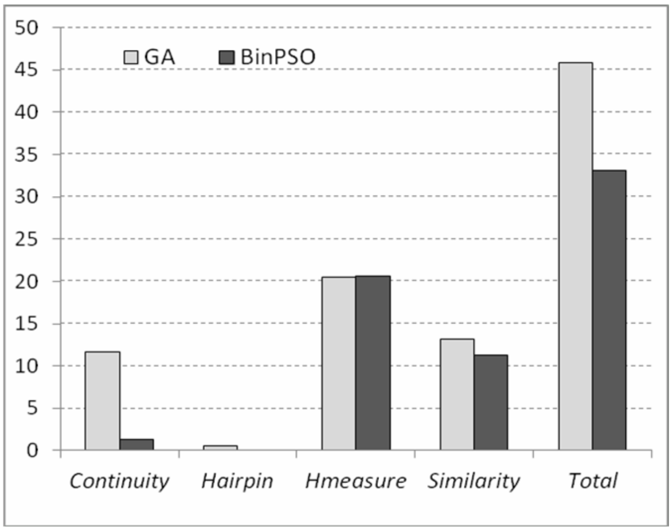

Figure 1. Average fitness comparison results between [9] and the proposed approaches, with 7 sequences and length of 20-mer

BinPSO significantly outperformed SA for most of the objectives, namely, Hmeasure, similarity, and hairpin. However, the sequences obtained from SA showed lower values in continuity. However, BinPSO still outperformed SA if the performance is measured based on total fitness value.

The BinPSO method is then compared with results given in Guangzhou et al. [12], which were obtained using PSO. However, the model was different from the BinPSO model, where the sequences were represented by modulus 4, and the weights for the

Table 3. Comparison results of the sequences in [4] and the sequences by BinPSO

Sequences	C^1	Ha^2	Hm^3	S^4	Total
SA [5]					
Fitness Value	0	1.71	103.71	62.71	168.13
Standard Deviation	0	1.70	6.39	4.20	–
BinPSO					
Average	20.71	0.57	76.86	61.86	160.00
Standard Deviation	1.594	0.12	3.15	0.568	–

[1]C, [2]Ha, [3]Hm, and [4]S are *continuity*, *hairpin*, $H_{measure}$, and *similarity* objectives values, respectively.

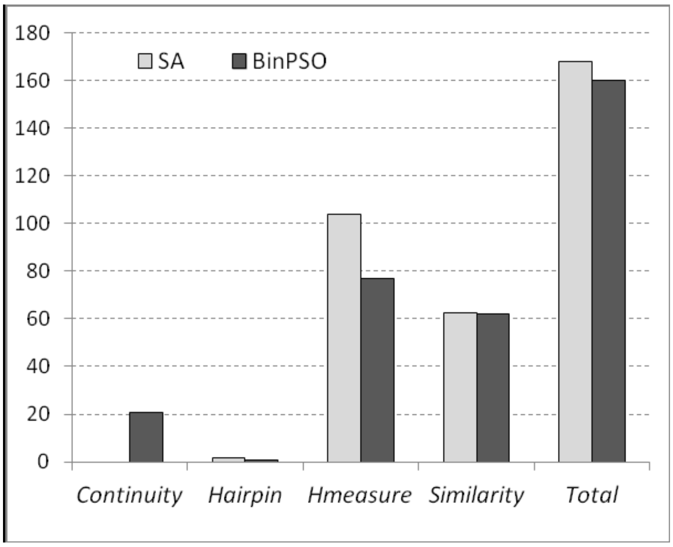

Figure 2. Average fitness comparison results between Tanaka et al. [4] and BinPSO method with 14 sequences and length of 20

Table 4. Comparison of the sequences in [12] and the sequences generated by BinPSO

Sequences	C^1	Ha^2	Hm^3	S^4	Total
PSO [13]					
Average	13.86	3.14	177.71	120.71	315.42
Standard Deviation	7.125	2.447	5.146	5.428	3.521
BinPSO					
Average	18.43	2.43	164.29	135.14	320.29
Standard Deviation	3.154	0.113	1.421	2.571	1.618

[1]C, [2]Ha, [3]Hm, and [4]S are *continuity*, *hairpin*, $H_{measure}$, and *similarity* objectives values, respectively.

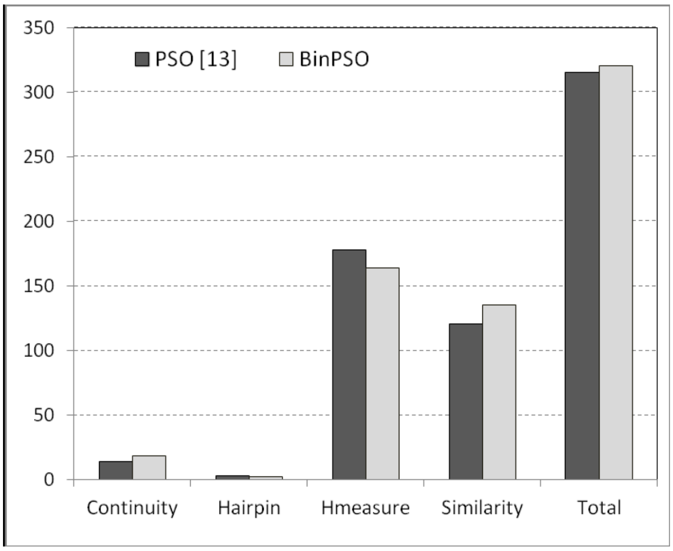

Figure 3. Average fitness comparison results between [12] and the proposed approach

fitness functions were obtained from Tanaka et al. [23]. The method from [12] produced 20 good sequences with the length of 20-mer. Results of the two algorithms are compared in Table 4 and Figure 3. The total values for all the objectives for BinPSO were not satisfying, where PSO [12] obtained better values. However, the sequences generated by BinPSO surpassed the sequences from the PSO [12] in two objectives. Sequences designed by BinPSO show lower values of *hairpin* and $H_{measure}$, while sequences from PSO [12] are better than BinPSO in the *similarity* and *continuity* objectives.

Table 5 and Figure 4 compare the results of the BinPSO with multi-objective PSO (MOPSO) [13]. The sequences generated by MOPSO also have 7 DNA sequences of 20-mer length, similar to sequences generated by SA. BinPSO significantly outperformed MOPSO for two objectives, namely, $H_{measure}$ and *continuity*. The sequences obtained from MOPSO showed lower values in *similarity*, while the values of *hairpin* for both approaches are equal to zero. For the total overall objectives, BinPSO achieved better minimum value.

From the computation point of view, an initial hypothesis is that BinPSO model is better than Continuous PSO model [14] since floating point does not exist in BinPSO computation. The number of sequence is chosen as 7 and the length for each sequence is 20-mer. Results of these two algorithms are compared in Table 6 and Figure 5.

Table 5. Comparison results of the sequences in [13] and the sequences by BinPSO

Sequences	C[1]	Ha[2]	Hm[3]	S[4]	Total
MOPSO [14]					
Average	10.00	0.00	22.43	11.14	43.57
Standard Deviation	1.795	2.532	1.534	2.034	1.356
BinPSO					
Average	1.29	0.00	20.57	11.29	33.15
Standard Deviation	3.019	0.326	4.346	2.140	1.112

[1]C, [2]Ha, [3]Hm, and [4]S are *continuity*, *hairpin*, $H_{measure}$, and *similarity* objectives values, respectively.

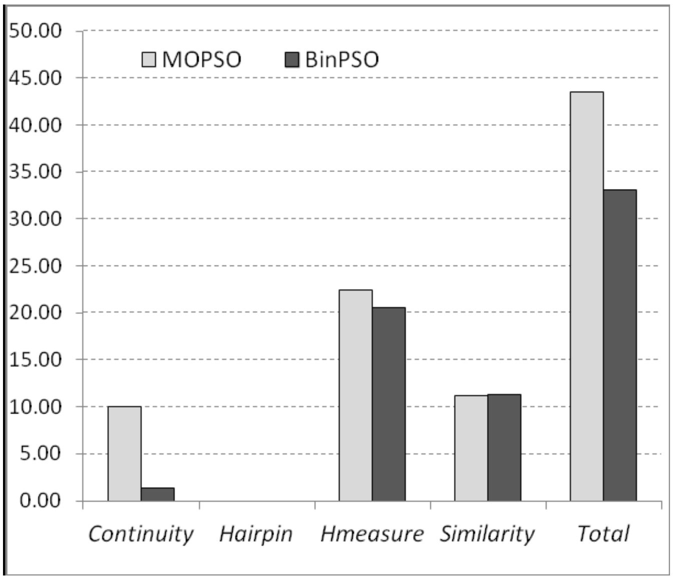

Figure 4. Average fitness comparison results between Zhao et al. [13] and BinPSO method with 7 sequences and length of 20

Table 6. Comparison of the sequences from continuous PSO [14] and BinPSO

Sequences	C^1	Ha^2	Hm^3	S^4	Total
Continuous PSO					
Average	1.773	0.456	56.77	50.82	109.82
Standard Deviation	1.366	0.342	1.408	1.096	1.025
BinPSO					
Average	0.00	0.57	54.22	43.95	98.74
Standard Deviation	0.321	1.502	1.211	0.745	0.912

C^1, Ha^2, Hm^3, and S^4 are *continuity*, *hairpin*, $H_{measure}$, and *similarity* objectives values, respectively.

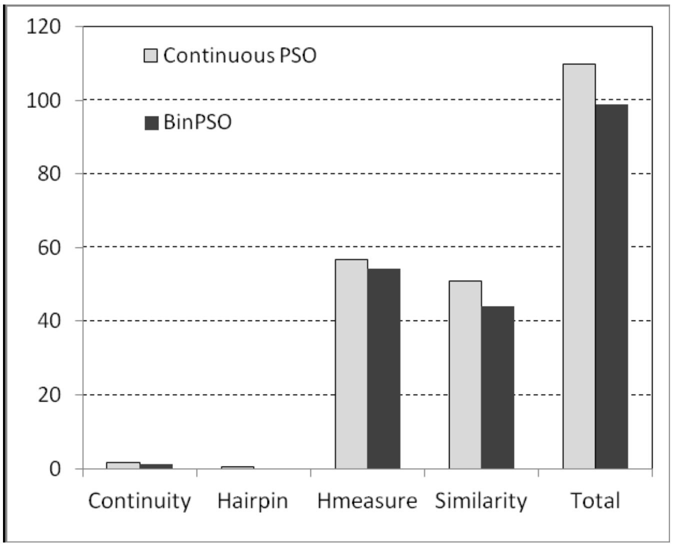

Figure 5. Average fitness comparison results between continuous PSO [14] and BinPSO

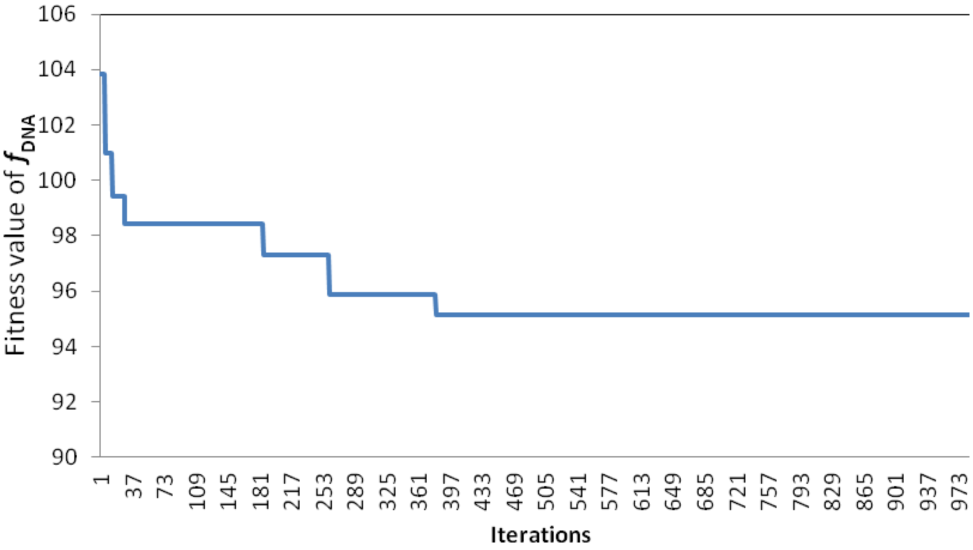

Figure 6. Convergence pattern of f_{DNA} for the continuous PSO algorithm for the case of 7 sequences with 20-mer length

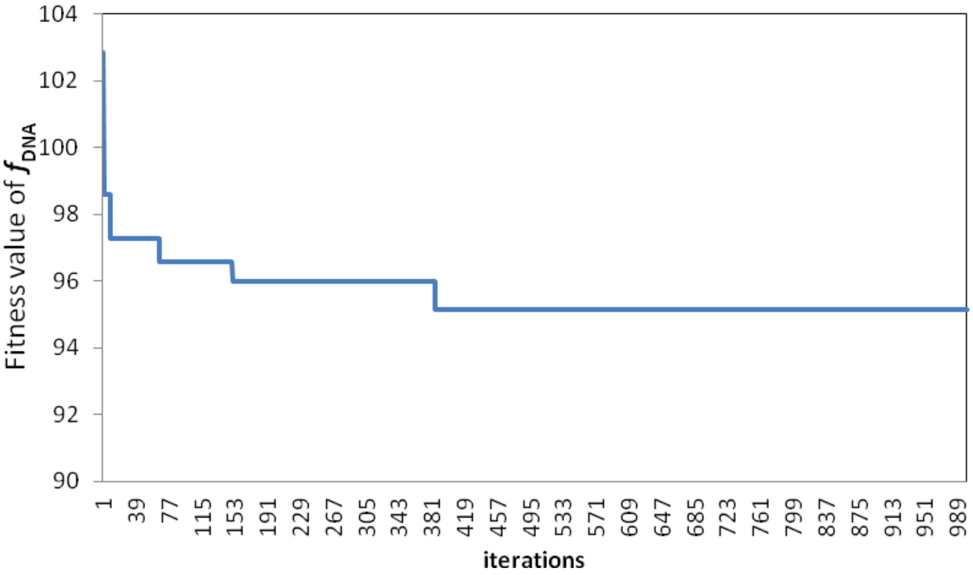

Figure 7. Convergence pattern of f_{DNA} for the BinPSO algorithm for the case of 7 sequences with 20-mer length

The observations show that BinPSO algorithm is able to provide the lowest values in the total objectives, compared with Continuous PSO. The sequences generated by BinPSO surpassed the sequences from Continuous PSO not only in certain objectives, but in all the objectives. Figure 6 demonstrates that the fitness function of f_{DNA} for Continuous PSO algorithm leads to convergence after 390 iterations, while the convergence pattern of BinPSO algorithm is illustrated in Figure 7. The particles for BinPSO converge after 380 iterations.

6. **Conclusions.** This study presented an application of binary particle swarm optimization in DNA sequence design. BinPSO was implemented to minimize four objectives, namely $H_{measure}$, *similarity*, *continuity*, and *hairpin*, and subjected to two constraints, $GC_{content}$ and T_m. The results of the BinPSO were compared to results obtained from

GA, SA, PSO and MOPSO. It was shown that BinPSO able to generate better or comparative sequences in several objectives than other approaches. Results from Continuous PSO also have been shown and compared with BinPSO. Continuous PSO has certain limitation, which can be overcome by BinPSO to produce better results. Future research will include improvements of the method by considering multi-objective PSO algorithms such as the vector evaluated PSO (VEPSO).

However, the necessity of DNA sequence design appears not only in DNA computing, but also in other fields, such as DNA nanotechnology and biotechnology [2]. In these fields, sequences are designed such that each element uniquely hybridizes to its complementary sequence, but not to any other sequence. The base sequences of the single strands DNA sequences determine the resulting DNA structure. Taking the wrong sequences would produce undesired structures. Therefore, many works have concentrated on producing good DNA sequences to avoid incorrect results.

Acknowledgement. The authors also would like to thank Universiti Teknologi Malaysia for supporting this research by UTM GUP Research Fund (Vote Q.J130000.7123.01J31). This research is also partly supported by the Ministry of Higher Education (FRGS Vote 78533), University of Malaya Post Graduate Research Grant (PPP PS078/2010B), University Malaya Research Grant (UMRG-025/09AET), and Ministry of Science, Technology, and Innovation (SciendFund MOSTI-13-02-03-3075). Muhammad Faiz Mohamed Saaid is indebted to University of Malaya for granting him a financial support under Bright Spark Scheme University of Malaya (SBSUM) and opportunity to do this research.

REFERENCES

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

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

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

[4] F. Tanaka, M. Naktsugawa, M. Yamamoto, T. Shiba and A. Ohuchi, Toward a general-purpose sequence design system in DNA computing, *Proc. of Congr. Evil. Comput.*, pp.73-78, 2002.

[5] A. Marathe, A. E. Condon and R. M. Corn, On combinatorial DNA word design, *Proc. of the 5th International Meeting on DNA Based Computers*, 1999.

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

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

[8] 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, *Proc. of the 3rd DIMACS Workshop DNA Based Comput.*, 1997.

[9] R. Deaton, R. C. Murphy, M. Garzon, D. T. Franceschetti and S. E. Stevens Jr., Good encodings for DNA-based solutions to combinatorial problems, *Proc. of the 2nd Annual Meeting on DNA Based Computers*, Princeton University, pp.159-171, 1996.

[10] R. Deaton, R. C. Murphy, J. A. Rose, M. Garzon, D. T. Franceschetti and S. E. Stevens Jr., Genetic search for reliable encodings for DNA-based computation, *The 1st Conference on Genetic Programming*, 1996.

[11] S. Y. Shin, I. H. Lee, D. Kim and B. T. Zhang, Multi-objective evolutionary optimization of DNA sequences for reliable DNA computing, *IEEE Transactions on Evolutionary Computation*, vol.9, no.2, pp.143-158, 2005.

[12] G. Cui, Y. Niu, Y. Wang, X. Zhang and L. Pan, A new approach based on PSO algorithm to find good computational encoding sequences, *Progress in Natural Science*, vol.17, no.6, pp.712-716, 2007.

[13] S. Zhao, Q. Zhang, J. Zhao and J. Li, DNA encoding based on multi-objective particle swarm, *Journal of Computational and Theoretical Nanoscience*, vol.4, pp.1249-1252, 2007.

[14] N. K. Khalid, Z. Ibrahim, T. B. Kurniawan et al., DNA sequence optimization based on continuous particle swarm optimization for reliable DNA computing and DNA nanotechnology, *Journal of Computer Science*, vol.4, no.11, pp.942-950, 2008.

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

[16] T. B. Kurniawan, N. K. Khalid, Z. Ibrahim, M. Khalid and M. Middendorf, An ant colony system for DNA sequence design based on thermodynamics, *Proc. of the 4th IASTED International Conference Advances in Computer Science and Technology*, Langkawi, Malaysia, pp.144-149, 2008.

[17] J. Kennedy and R. C. Eberhart, Particle swarm optimization, *Proc. of IEEE International Conference on Neural Networks*, Perth, WA, Australia, pp.1942-1948, 1995.

[18] H.-C. Liu, J.-M. Yih, W.-C. Lin and T.-S. Liu, Fuzzy c-means algorithm based on PSO and Mahalanobis distances, *International Journal of Innovative Computing, Information and Control*, vol.5, no.12(B), pp.5033-5040, 2009.

[19] Y. Wakasa, K. Tanaka, T. Akashi and Y. Nishimura, PSO-based simultaneous tuning method for PID controllers and dead-zone compensators and its application to ultrasonic motors, *International Journal of Innovative Computing, Information and Control*, vol.6, no.10, pp.4593-4604, 2010.

[20] J. Santa Lucia Jr., A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbour thermodynamics, *Proc. of Nat. Acad. Sci. USA*, vol.95, pp.1460-1465, 1998.

[21] R. C. Eberhart and Y. Shi, Comparing inertia weights and constriction factors in particle swarm optimization, *Proc. of IEEE Congress Evolutionary Computation*, San Diego, CA, pp.84-88, 2000.

[22] F. van der Bergh and A. P. Engelbrecht, *An Analysis of Particle Swarm Optimizers*, Ph.D. Thesis, University of Pretoria, 2002.

[23] F. Tanaka, M. Nakatsugawa, M. Yamamoto, T. Shiba and A. Ohuchi, Developing support system for sequence design in DNA computing, *Proc. of the 7th Int. Workshop DNA Based Computer*, pp.340-349, 2001.