

DNA Sequence Optimization Based on Continuous Particle Swarm Optimization for Reliable DNA Computing and DNA Nanotechnology

¹Noor Khafifah Khalid, ¹Zuwairie Ibrahim, ¹Tri Basuki Kurniawan, ¹Mohamad Shukri Zainal Abidin, ¹Marzuki Khalid and ²Andries P. Engelbrecht

¹Centre for Artificial Intelligence and Robotics, Department of Mechatronics and Robotics
University Teknologi Malaysia, Skudai, Johor

²Department of Computer Science, University of Pretoria, South Africa

Abstract: Problem statement: In DNA based computation and DNA nanotechnology, the design of good DNA sequences has turned out to be an essential problem and one of the most practical and important research topics. Basically, the DNA sequence design problem is a multi-objective problem and it can be evaluated using four objective functions, namely, H_{measure} , similarity, continuity and hairpin. **Approach:** There are several ways to solve multi-objective problem, however, in order to evaluate the correctness of PSO algorithm in DNA sequence design, this problem is converted into single objective problem. Particle Swarm Optimization (PSO) is proposed to minimize the objective in the problem, subjected to two constraints: melting temperature and GC_{content} . A model is developed to present the DNA sequence design based on PSO computation. **Results:** Based on experiments and researches done, 20 particles are used in the implementation of the optimization process, where the average values and the standard deviation for 100 runs are shown along with comparison to other existing methods. **Conclusion:** The results achieve verified that PSO can suitably solves the DNA sequence design problem using the proposed method and model, comparatively better than other approaches.

Key words: Particle swarm optimization, DNA sequence design, optimization, user-defined weights

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. A sequence of DNA can be read from 5'-end (the ribose end) of one sequence and the 3'-end (the phosphate end) of the other sequence.

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 initiated^[1]. Usually, in DNA computing, the calculation process consists of several chemical reactions, where the successful wet lab experiment depends on DNA sequences we used. Thus, DNA sequence design turns out to be one of the approaches to achieve high computation accuracy and become one of the most practical and 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

Corresponding Author: Zuwairie Ibrahim, Center for Artificial Intelligence and Robotics (CAIRO), Department of Mechatronics and Robotics, Universiti Teknologi Malaysia, Skudai, Johor

experiments, a method for the systematic 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 this chapter, DNA sequences are designed based on Particle Swarm Optimization (PSO)^[7]. Even though the DNA sequence design is a multi-objective problem, using weighted sum method, it is converted into single objective problem. Weighted sum method scalarized a set of objectives into a single objective by pre-multiplying each objective with a user-supplied weight. This method is the simplest approach and the most widely used classical approach. However, the value of the weights is difficult to determine, it depends on the importance of each objective in the context of the problem and a scaling factor^[8].

DNA sequence design: 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^[1-6,9-23]. Seeman *et al.*^[9,10] designed sequences using overlapping subsequences to enforce uniqueness. The approach is based on the "repairing" of sequences. Baum suggested a method to design unique sequences by avoiding multiple usages of subsequences, restricting the choice of nucleotides at the ends of the sequences^[11].

Hartemink *et al.*^[12] designed sequences for the programmed mutagenesis, using the exhaustive search method, "SCAN". Although the method is successful, it took much computational time. Penchovsky and Ackermann^[13] implemented a random search algorithm

to design DNA sequences. Binary information was encoded in DNA strands a twelve-bit DNA library was demonstrated.

Furthermore, Tanaka *et al.*^[14] proposed some sequence fitness criteria and generated the sequences using simulated annealing^[15]. The objective is to find proper combinations of the proposed fitness functions in order to find more promising solutions. Marathe *et al.*^[16] chose a dynamic programming approach to design DNA sequences based on Hamming distance. A dynamic programming based algorithm for the selection of sequences with a given free energy was also presented.

Feldkamp *et al.*^[17] used a directed graph to design DNA sequences. The nodes in the graph represent base strands, where each node can be extended into 4 strands that can appear as successors in a longer sequence. Then, by travelling the graph from root to leaf, DNA sequences are formed. This approach can also find a set of orthogonal DNA sequences within a predefined error rate quickly.

Frutos *et al.*^[18] and Arita and Kobayashi^[5] developed a template-map strategy to choose a huge number of dissimilar sequences while having to design only a significantly smaller number of templates and maps. Deaton *et al.*^[19,20] used genetic algorithms to generate a set of unique DNA sequences using Hamming distance for measuring the uniqueness of DNA sequences and found better sequences than Adleman's original sequences^[3].

Arita *et al.*^[1] developed a DNA sequence design system using a genetic algorithm with three fitness criteria. Self-complementary sequences were designed for the Whiplash model and compared the results to that of a random generate-and-test algorithm. Shin *et al.*^[21] developed NACST/Seq that implements multi-objective evolutionary optimization to generate sets of DNA sequences. NACST/Seq generated DNA sequences which satisfied all constraints which are H_{measure} , similarity, hairpin and continuity.

Guangzhou *et al.*^[22] designed DNA sequences using PSO based on four fitness criteria. Twenty strands of DNA sequence with 20-mer were connected one by one in the same direction to form a strand of a 400-mer DNA sequence. The strand is denoted as a particle, where 10 particles are created to form a swarm. However, the employed model cannot utilize high -dimensional search space and two of the fitness criteria, GC_{content} and melting temperature are not suitable to be the objective functions. Zhao *et al.*^[23] implemented a multi-objective PSO to design DNA sequences based on three fitness criteria. However, the

study failed to describe the modeling of DNA sequence design using PSO and several parameters in the algorithm also have not been explained. Pareto front solutions were not employed.

Compare to previous works^[22,23] this chapter proposes different model of DNA sequence design problem. The length and number of sequences can be chosen by user and the particle in this model carries a set of DNA sequences. Dimensions in PSO computation represented strands of DNA sequences and the continuous search space are utilized.

Objectives and constraints in DNA sequence design:

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. Another two additional objective functions, hairpin and continuity, are used to prevent the secondary structure of a DNA sequence. GC_{content} and melting temperature are used as the constraints, which the ranges for these constraints are set by user preference. The formulations for all objectives and constraints can be referred to^[24].

DNA sequence design is actually a multi-objective optimization problem. However, in this chapter, the problem is converted into a single objective problem, formulated as follows:

$$\min f_{\text{DNA}} = \sum_i \omega_i f_i \quad (1)$$

subjected to T_m and GC_{content} constraints, where f_i are the objective function for each $i \in (H_{\text{measure}}, \text{similarity}, \text{hairpin}, \text{continuity})$ and w_i are the weights for each f_i In this study, the weights are defined by the user.

Particle swarm optimization: Particle Swarm Optimization (PSO) is a population-based stochastic optimization technique developed by Kennedy and Eberhart in 1995^[7]. This method finds an optimal solution by simulating social behavior of bird flocking. The PSO algorithm consist 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.

PSO has been successfully applied to solve many optimization problems, such as power system design^[25], data classification^[26], robotic applications^[27], decision making for stock market^[28] and simulation and identification of emergent systems^[29].

With reference to the original PSO, each particle knows its best value so far (*pbest*), velocity and position. Additionally, each particle knows the best value in its neighborhood (*gbest*). A particle modifies its position based on its current velocity and position. The velocity of each particle is calculated using:

$$v_i^{k+1} = \omega v_i^k + c_1 r_1 (pbest_i - s_i^k) + c_2 r_2 (gbest^k - s_i^k) \quad (2)$$

Where:

- v_i^k, v_i^{k+1} and s_i^k = The velocity vector, modified velocity vector and positioning vector of particle i at generation k , respectively
- $pbest_i^k$ = The best position found by particle i
- $gbest^k$ = The best position found by the particle’s neighborhood or the entire swarm
- c_1 and c_2 = 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 = The 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, whereas 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.

The modified position vector, s_i^{k+1} is obtained using:

$$s_i^{k+1} = s_i^k + v_i^{k+1} \quad (3)$$

The standard PSO algorithm to find the best positioning vector in PSO using i number of particles can be summarized as:

- Randomly initialize all position vectors, $s_i = [s_{i1}, s_{i2}, s_{i3}, \dots, s_{in}]$ and velocity vector, $v_i = [v_{i1}, v_{i2}, v_{i3}, \dots, v_{in}]$ can be initialized to zero
- Velocity vector v_i^{k+1} of particle i is calculated using Eq. 25
- New positioning vector s_i^k of particle i is calculated using Eq. 26
- If $F(s_i^k)$ is better than the $F(p_{best_i}^k)$, the $p_{best_i}^k$ is set to position vector s_i^k
- If $F(p_{best_i}^k)$ is better than the $F(g_{best}^k)$, the $p_{best_i}^k$ is set to g_{best}^k
- If a pre-determined number of generations is reached or a sufficiently good fitness is obtained, the process stops. Otherwise, the process continues to step 2

Optimization of DNA sequence based on PSO: For DNA sequence design application, the proposed approach is based on basic PSO algorithm. A DNA sequence is represented in binary, where A, C, G and T, are encoded as 00₂, 01₂, 10₂ and 11₂, respectively. A sequence contains of several bases, where one sequence represents one dimension. For each dimension, the length is depended on the length of DNA sequence (l) with the general formula of $(4^l - 1)$. For 5-mer nucleic acid sequence, the range of the search space is $(4^5 - 1)$, from 0 to 1023, in decimal, which represents sequence AAAAA to TTTTT. For a 10-mer DNA sequence, the range of the search space is $(4^{10} - 1)$, from 0 to 1048576 and for 20-mer DNA sequence, the range is $(4^{20} - 1)$, from 0 to 1099511627776.

To obtain a set of 3 DNA sequences, for example, 3 dimensions should be used in a search space. Therefore, each particle in the search space carries 3 DNA sequences. In this study, 20 particles are employed and randomly initialized in the search space.

The basic PSO has been developed for continuous-valued search spaces. In order to eliminate the floating point in the computation, caused by the random values and coefficient factors, the floating values are approximated to the nearest decimal numbers. The decimal numbers are converted to binary and binary representations are converted into sequences. For example, the decimal number of 908.8 is approximated to 909₁₀, which is equals to 1110001101₂, in binary and is converted into "TGATC" DNA sequence. The values of the constraints are 30%-80% for GC_{content} and 50°C-80°C for T_m. The T_m was computed based on the Nearest-Neighbor (NN) method^[30]. Table 1 shows the values of PSO control parameters used in the experiments. In this study, a decreasing inertia weight

Table 1: The value of PSO control parameters

Parameter	Value
Cognitive factor, c_1	1.4
Social factor, c_2	1.4
Inertia weight,	0.9-0.4
Random values: r_1, r_2	[0,1]
No. of particles	20
Max iteration	1000

(Eq. 4) is used, where 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^[31].

$$\omega k' = \omega_{max} - \left(\frac{\omega_{max} - \omega_{min}}{k_{max}} \times k \right) \quad (4)$$

RESULTS

The results of the proposed approach are compared with existing approaches, taken from Deaton *et al.*^[19], Guangzhou *et al.*^[22] and Zhao *et al.*^[23]. For each comparison, 100 runs have been performed by PSO and the average performance is exhibited in terms of the mean value and the standard deviation of the objective function evaluations. Results for all of the aforementioned comparisons are reported in Table 3, 4 and 5.

Table 2 summarizes parameter values for the objectives and constraints of the DNA sequence design problem. Since there are several ways to determine the weights in Eq. 1 and the weights depended on user preference, in this experiment, the weights are all set to default value, which is 1.

The PSO method is first compared with results given in^[19], 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 3 and Fig. 1. PSO reached lower values in the total objectives, compared to the GA. The sequences generated by PSO surpassed the sequences from the GA in three objectives.

Sequences designed by PSO show lower values of H_{measure}, continuity and similarity, while sequences from Deaton *et al.*^[19, 20] are better than PSO in the hairpin objective. Fig. 2 demonstrates that the fitness function of f_{DNA} leads to convergence after 390 iterations.

The PSO method is then compared with results given in Guangzhou *et al.*^[22], which were obtained also using PSO.

Table 2: The parameters for objectives and constraints

Parameter	Value
H_{con}, S_{con}	6
H_{dis}, S_{dis}	17%
t (for continuity)	2
p, r (for hairpin)	6
Na^+	1 M
C_T	10 nM

Table 3: Comparison of the sequences in^[19] and the sequences generated by PSO

Sequences	C^1	Ha^2	Hm^3	S^4	Total
ATAGAGTGGATAGTTCTGGG	9	0	57	55	121
CATTGGCGGCGGTAGGCTT	0	0	65	44	109
CTTGTGACCGCTTCTGGGGA	16	0	67	60	143
GAAAAAGGACCAAAAAGAGAG	41	0	58	40	139
GATGGTGCTTAGAGAAGTGG	0	0	54	51	105
TGTATCTCGTTTTAACATCC	16	4	72	41	133
TTGTAAGCCTACTGCGTGAC	0	0	71	47	118

GA					
Fitness value	11.71	0.57	20.43	13.14	45.85
Standard deviation	14.80	1.51	7.14	7.43	-

PSO					
Average	1.773	0.456	21.742	12.156	36.127
Standard deviation	1.366	0.342	1.408	1.096	-

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

Table 4: Comparison of the sequences in^[22] and the sequences generated by PSO

Sequences	C^1	Ha^2	Hm^3	S^4	Total
GTCAAATTCCTCTATCGTC	18	0	67	37	122
AGCGATAGTAGATCACCTGA	0	0	66	39	105
CACGATATAGCTTCGAGCCG	0	0	60	51	111
AATACACCGCTCACCAAGGA	0	0	72	42	114
AACAGGGAAGAATGCAGAGG	9	0	60	34	103
CCTCTACCAGCCAATGATGC	0	0	64	33	97
TTAGGACTCGACGCCACTCC	0	0	67	39	106
CCATGACCGAGGATCCACGT	0	0	49	43	92
CGCCATTATCAGGCCTTTAC	9	0	60	46	115
ACACAGTGGACGCACATACA	0	0	58	38	96
TTATCCCGCTTCTTCCGT	9	0	60	48	117
AATACGGTTCAAGCGGCTTC	0	4	68	38	110
TAAAGGCGCGTGATCGGAAG	9	0	56	41	106
TTGTTCCGGATTGAGCAACT	9	5	60	43	117
GTCAGTGCAGCACTCAT	0	4	68	46	118
CCATAAAGTCCAGCTCGCG	9	0	63	41	113
CAACATAGAGTCAGGCGCTG	0	0	61	53	114
CCAATGAGTCACCTCGTTTCG	0	9	61	49	119
GGGGTGGAGGCCAACTATT	25	0	59	39	123
CAGCGTCTGAACCTCCATA	0	0	65	45	110

PSO ^[22]					
Average	13.86	3.14	177.71	120.71	315.42
Standard deviation	7.125	2.447	5.146	5.428	-

PSO					
Average	21.25	0.57	203.44	117.31	342.57
Standard deviation	2.331	0.115	1.736	1.842	-

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

However, the model was different from the proposed model in this chapter, while the weights for the fitness functions were obtained from Tanaka *et al.*^[32].

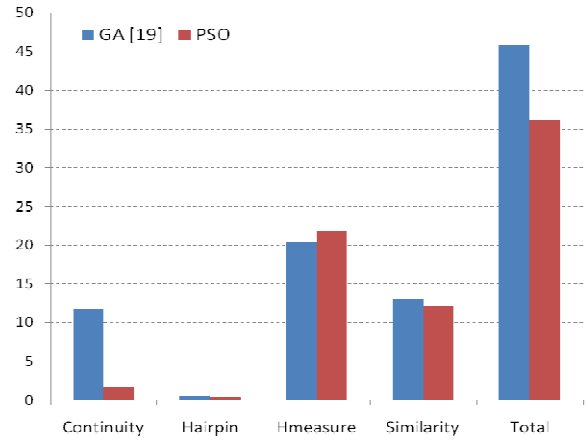


Fig. 1: Average fitness comparison results between^[19] and the proposed approaches, with 7 sequences and length of 20 mer

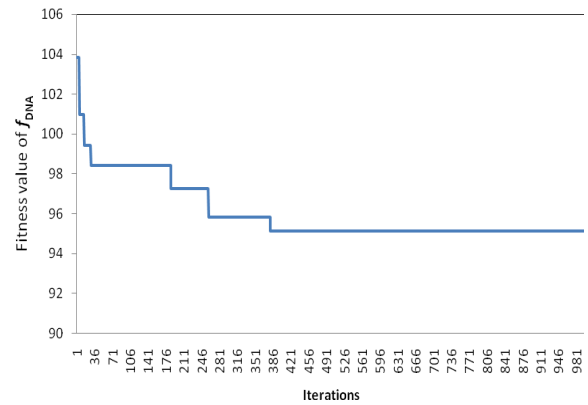


Fig. 2: Convergence pattern of f_{DNA} for the PSO algorithm, with 7 sequences and length of 20 mer

The method from^[22] produced 20 good sequences with the length of 20-mer. Results of the two algorithms are compared in Table 4 and Fig. 3. The total values for all the objectives for the proposed approach were not satisfying, where PSO^[22] obtained better values. However, the sequences generated by PSO surpassed the sequences from the PSO^[22] in two objectives. Sequences designed by PSO show lower values of hairpin and similarity, while sequences from PSO^[22] are better than PSO in the $H_{measure}$ and continuity objectives. Fig. 4 demonstrates that the fitness function of f_{DNA} leads to convergence after 307 iterations.

Table 5 and Fig. 5 compares the results of the PSO with Multi-Objective PSO (MOPSO) as given^[23]. The sequences generated by MOPSO also have 7 DNA sequences of 20 mer length, similar to sequences

Table 4: Comparison of the sequences in^[22] and the sequences generated by PSO

Sequences	C ¹	Ha ²	Hm ³	S ⁴	Total
GTCAAATTCCTCTATCGTC	18	0	67	37	122
AGCGATAGTAGATCACCTGA	0	0	66	39	105
CACGATATAGCTTCGAGCCG	0	0	60	51	111
AATACACCGCTCACCAAGGA	0	0	72	42	114
AACAGGGAAGAATGCAGAGG	9	0	60	34	103
CCTCTACCAGCCAATGATGC	0	0	64	33	97
TTAGGACTCGACGCCACTCC	0	0	67	39	106
CCATGACCGAGGATCCACGT	0	0	49	43	92
CGCCATTATCAGGCCTTTAC	9	0	60	46	115
ACACAGTGGAGCGACATACA	0	0	58	38	96
TTATCCCGCCTCTTCTCCGT	9	0	60	48	117
AATACGGTTCFAAGCGGCTTC	0	4	68	38	110
TAAAGGCGCGTGATCGGAAG	9	0	56	41	106
TTGTTCGGGATTGAGCAACT	9	5	60	43	117
GTCAGTGCAGCACTCAT	0	4	68	46	118
CCATAAACTGCCAGCTCGCG	9	0	63	41	113
CAACATAGATCAGGCGCTG	0	0	61	53	114
CCAATGAGTCACCTCGTTTCG	0	9	61	49	119
GGGGTGGAGGCCCAACTATT	25	0	59	39	123
CAGCGGTCTGAACCTCCATA	0	0	65	45	110
PSO ^[22]					
Average	13.86	3.14	177.71	120.71	315.42
Standard deviation	7.125	2.447	5.146	5.428	-
PSO					
Average	21.25	0.57	203.44	117.31	342.57
Standard deviation	2.331	0.115	1.736	1.842	-

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

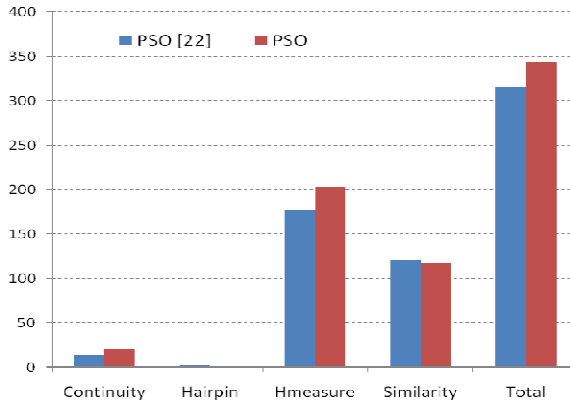


Fig. 3: Average fitness comparison results between^[22] and the proposed approach, with 20 sequences and length of 20 mer

generated by SA. PSO significantly outperformed MOPSO for two objectives, namely, H_{measure} and continuity. However, the sequences obtained from MOPSO showed lower values in similarity and hairpin. For the total overall objectives, PSO achieved better minimum value than MOPSO. The convergence pattern of PSO is illustrated in Fig. 6. The particles converge within 800-820 iterations.

Table 5: Comparison results of the sequences in^[23] and the sequences by PSO method

Sequences	C ¹	Ha ²	Hm ³	S ⁴	Total
CATCAGCCGGACTCGTCAGT	0	0	24	12	36
AGATCGCATGTAAAGGAGTG	9	0	26	13	48
AAAGCAGGGTGTATCAGTCA	18	0	26	14	58
TACAGGCGCTAATTAGCTCC	0	0	18	10	28
GCGGACCAACACATATGAG	9	0	23	12	44
ATCATCATTTTCATGGGGCAA	25	0	20	9	54
GGGATCGACGTATATTAACG	9	0	20	8	37
MOPSO ^[23]					
Average	10.000	0.00	22.43	11.14	43.57
Standard Deviation	14.795	1.512	7.138	7.432	-
PSO					
Average	1.773	0.456	21.742	12.156	36.127
Standard Deviation	1.366	0.342	1.408	1.0964	-

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

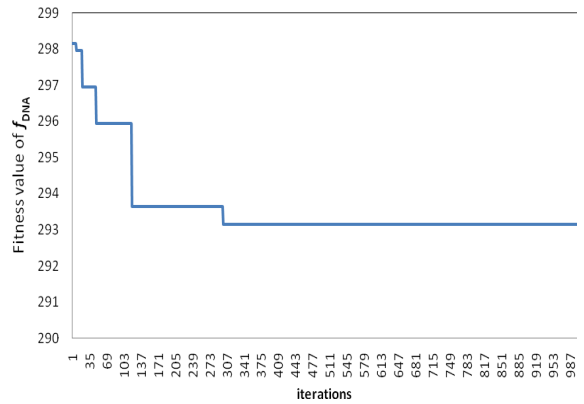


Fig. 4: Convergence pattern of f_{DNA} for the PSO algorithm, with 20 sequences and length of 20 mer

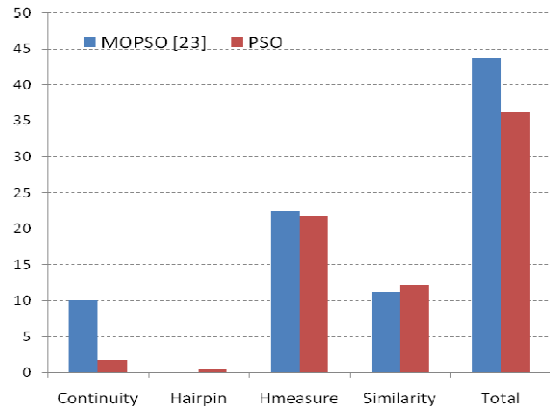


Fig. 5: Average fitness comparison results between Zhao *et al.*^[23] and PSO method with 7 sequences and length of 20.

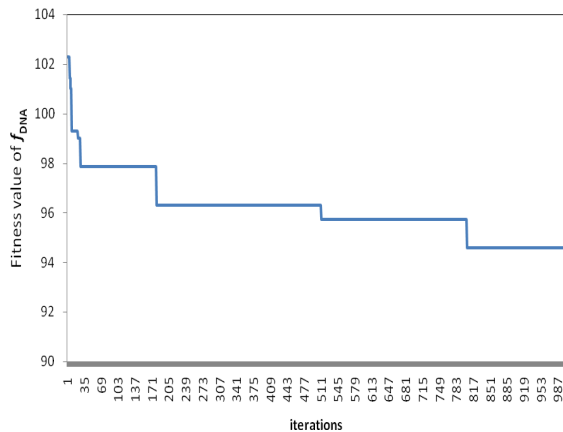


Fig. 6: Convergence pattern of f_{DNA} for the PSO algorithm, with 7 sequences and length of 20 mer

CONCLUSION

This study presented an application of particle swarm optimization in DNA sequence design. PSO was implemented with four objectives, namely $H_{measure}$, similarity, continuity and hairpin and subjected to two constraints, $GC_{content}$ and T_m . However, the problem is converted to single objective problem, using weight aggregation.

A model to implement PSO for DNA sequence design was presented, where each particle searches for the minimum value of the objective function in an n-dimensional search space. Each particle carries n sequences, where the sequences are represented by binary strings.

The results of the PSO were compared to results from a GA, MOPSO and other PSO model. It was shown that PSO can generate better or comparative sequences in several objectives than other systems. However, the proposed approach has to be improved and explored further. Future research will include improvements of the method by considering a multi-objective PSO such as the Vector Evaluated PSO (VEPSO).

ACKNOWLEDGEMENT

This research is supported financially by the Ministry of Science, Technology and Innovation (MOSTI), Malaysia under eScience Fund Research Funding (Vot 79034) and Ministry of Education under Fundamental Research Grant Scheme (FRGS) (Vot 78225 & 78226). The authors are indebted to MOSTI, MOHE and Universiti Teknologi Malaysia for

granting them a financial support and opportunity to do this research.

REFERENCES

1. Arita, M., A. Nishikawa, M. Hagiya, K. Komiya, H. Gouzu and K. Sakamoto, 2000. Improving sequence design for DNA computing. Proceedings of the Conference on Genetic and Evolutionary Computation, July 8-12, Morgan Kaufmann, Las Vegas, Nevada, USA., pp: 875-882. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.29.3358>.
2. Reece, R.J., 2004. Analysis of Genes and Genomes. 1st Edn., Wiley, New York, USA., ISBN: 10: 0470843802, pp: 490.
3. Adleman, L., 1994. Molecular computation of solutions to combinatorial problems. Science, 266: 1021-1024. <http://www.ncbi.nlm.nih.gov/pubmed/7973651>.
4. Kashiwamura, S., A. Kameda, M. Yamamoto and A. Ohuchi, 2004. Two-step search for DNA sequence design. Proceedings of the Conference on International Technical Circuits/Systems, Computers and Communications, July 7-9, Bo-Kwang Phoenix Park, Korea, E87-A: 1446-1453. <http://ci.nii.ac.jp/naid/110003213054/>.
5. Arita, M. and S. Kobayashi, 2002. DNA sequence design using templates. New Generat. Comput., 20: 263-277. <http://portal.acm.org/citation.cfm?id=639002.639007&coll=GUIDE&dl=GUIDE>.
6. Kobayashi, S. and T. Kondo, 2002. On template method for DNA sequence design. Proceeding of 8th International Meeting on DNA Based Computers, June 10-13, Springer-Verlag, Berlin, pp: 205-214. <http://portal.acm.org/citation.cfm?id=647863.736629&coll=&dl=>
7. Kennedy, J. and R.C. Eberhart, 1995. Particle swarm optimisation. Proceeding of IEEE International Conference on Neural Networks, Nov. 27-Dec 1, IEEE Service Center, Perth, Australia, pp: 1942-1948. <http://www.engr.iupui.edu/~shi/Coference/psopap4.html>.
8. Kalyanmoy Deb, 2001. Multi-objective Evolutionary Optimization. 4th Edn., John Wiley and Sons, England, pp: 50.
9. Seeman, N.C. and N.R. Kallenbach, 1983. Design of immobile nucleic acid junctions. Biophys. J., 44: 201-209. <http://www.ncbi.nlm.nih.gov/pubmed/6197102>.

10. Seeman, N.C., 1990. De novo design of Sequences for nucleic acid structural engineering. *J. Biomol. Struct. Dynam.*, 8: 573-581. <http://www.ncbi.nlm.nih.gov/pubmed/2100519>.
11. Baum, E.B., 1996. DNA sequences useful for computation. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.55.2306>.
12. Hartemink, A.J., D.K. Gifford and J. Khodor, 1998. Automated constraint based nucleotide sequence selection for DNA computation. *Proceeding 4th DIMACS Workshop DNA Based Computer*, June 16-19, Philadelphia, Pennsylvania, USA., pp: 227-235. DOI: 10.1016/S0303-2647(99)00050-7.
13. Penchovsky, R. and J. Ackermann, 2003. DNA library design for molecular computation. *J. Computat. Biol.*, 10: 215-229. <http://www.ncbi.nlm.nih.gov/pubmed/12804092>.
14. Tanaka, F., M. Naktusugawa, M. Yamamoto, T. Shiba and A. Ohuchi, 2002. Toward a general-purpose sequence design system in DNA computing *Proceeding Congress Evolutionary Computing*, May 12-17, IEEE Publisher, USA., pp: 73-78. DOI: 10.1109/CEC.2002.1006212.
15. Kirkpatrick, S.C.D., Gelatt Jr. and M.P. Vecchi, 1983. Optimization by simulated annealing. *Science*, 220: 671-680. <http://www.ncbi.nlm.nih.gov/pubmed/17813860>.
16. Marathe, A., A.E. Condon and R.M. Corn, 1999. On combinatorial DNA word design. *Proceedings of the 5th International Meeting on DNA Based Computers*, June 14-15, Massachusetts Institute of Technology DIMACS, USA.
17. Feldkamp, U., S. Saghafi, W. Banzhaf and H. Rauhe, 2001. DNA sequence generator-A program for the construction of DNA sequences. *Proceeding 7th International Workshop DNA Based Computing*, pp: 179-188.
18. Frutos, A.G., A.J. Thiel, A.E. Condon, L.M. Smith and R.M. Corn, 1997. DNA computing at surfaces: Four base mismatch word design. *Proceeding of the 3rd DIMACS Workshop DNA Based Computing*, June 23-25, Philadelphia, PA., DIMACS, USA., pp: 238.
19. Deaton, R., R.C. Murphy, M. Garzon, D.T. Franceschetti and S.E. Stevens, 1996. Good Encodings for DNA-based solutions to combinatorial problems. *Proceedings of the Second Annual Meeting on DNA Based Computers*, June 10-12, Princeton University, DIMACS, USA., pp: 159-171. <http://citeseerx.ist.psu.edu/showciting?sessionid=7DF7A3302BA04E1E1B704C547FC1BC5C?cid=58871>.
20. Deaton, R., R.C. Murphy, J.A. Rose, M. Garzon, D.T. Franceschetti and S.E. Stevens, 1996. Genetic search for reliable encodings for DNA-based computation, *Proceeding of the 1st Conference on Genetic Programming*, July 28-31, California, USA., pp: 9-15. http://www.cs.bham.ac.uk/~wbl/biblio/gp-html/deaton_1996_gsreDNA.html.
21. Shin, S.Y. I.H. Lee, D. Kim and B.T. Zhang, 2005. Multi-objective evolutionary optimization of DNA sequences for reliable DNA computing. *IEEE Trans. Evolut. Comput.*, 9: 143-158. DOI: 10.1109/TEVC.2005.844166.
22. Cui Guangzhao, Niu Yunyun, Wang Yangfeng, Zhang Xuncaai and Pan Linqiang, 2007. A new approach based on PSO algorithm to find good computational encoding sequences. *Progress Natural Sci.*, 17: 712-716. DOI: 10.1080/10002007088537464.
23. Shihua Zhou, Qiang Zhang, Jing Zhao and Jinsong Li, 2007. DNA encoding based on multi-objective particle swarm. *J. Comput. Theoret Nanosci.*, 4: 1249-1252. DOI: 10.1166/jctn.2007.005.
24. Kurniawan, T.B., N.K. Khalid, Z. Ibrahim, M. Khalid and M. Middendorf, 2008. An ant colony system for DNA sequence design based on thermodynamics. *Proceedings of the 4th IASTED International Conference Advances in Computer Science and Technology*, Langkawi, Malaysia, pp: 144-149.
25. Das, T.K. and G.K. Venayagamoorthy, 2006. Optimal design of power system stabilizers using a small population based PSO. *Power Engineering Society General Meeting*, June 18-22, IEEE Compute Society, USA., pp: 7.
26. Cai, X. and D.C. Wunsch, 2005. Engine data classification with simultaneous recurrent network using hybrid PSO-EA algorithm. *Proceeding of International Joint Conference on Neural Network*, Montreal, Quebec, Canada. IEEE, July 31-Aug. 4, 4: 2319-2323. DOI: 10.1109/IJCNN.2005.1556263.
27. Doctor, S., G.K. Venayagamoorthy and V.G. Gudise, 2004. Optimal PSO for collective robotic search applications. *Cong. Evolutionary Computation (CEC 2004)*, 2: DOI: 1390-1395. 10.1109/CEC.2004.1331059.
28. Nenortaite, J. and R.D. Simutis, 2005. Decision-making model for stock markets based on particle swarm optimization algorithm. *New Math. Natl. Comput.*, 1: 261-274. http://econpapers.repec.org/article/wsinmncxx/v_3A01_3Ay_3A2005_3Ai_3A02_3Ap_3A261-274.htm.

29. Voss, M.S. and X. Feng, 2002. A new methodology for emergent system identification using Particle Swarm Optimization (PSO) and the Group Method of Data Handling (GMDH). Proceeding of Genetic and Evolutionary Computation Conference, July 9-13, Morgan Kaufmann Publishers, USA., pp: 1227-1232. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.13.5124>.
30. SantaLucia, J., 1998. A unified view of polymer, dumbbell and oligonucleotide DNA nearest-neighbour thermodynamics. Proc. Natl Acad. Sci. USA., No. 95. pp: 1460-1465. <http://www.ncbi.nlm.nih.gov/pubmed/9465037>.
31. Eberhart, R.C. and Y. Shi, 2000. Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization. Proceedings of IEEE Congress Evolutionary Computation, July 16-19, San Diego, CA., USA., pp: 84-88. DOI: 10.1109/CEC.2000.870279.
32. Tanaka, F., M. Nakatsugawa, M. Yamamoto, T. Shiba and A. Ohuchi, 2001. Developing support system for sequence design in DNA computing. Proceeding 7th International Workshop DNA Based Computing, June 10-13, Springer-Verlag, Berlin, pp: 129-137. <http://portal.acm.org/citation.cfm?id=736483>.