The Fourteenth International Symposium on Artificial Life and Robotics 2009 (AROB 14th '09),
B-Con Plaza, Beppu, Oita, Japan, February 5 - 7, 2009

Implementation of Ant Colony System for DNA Sequence Optimization

Zuwairie Ibrahim, Tri Basuki Kurniawan, Noor Khafifah Khalid, and Marzuki Khalid

Centre for Artificial Intelligence and Robotics (CAIRO), Department of Mechatronics and Robotics, Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor Darul Takzim, Malaysia.
(zuwairiee@fke.utm.my; tribasukikurniawan@yahoo.com; ifah_khalid@yahoo.com; marzuki@utmkl.utm.my)

Abstract: DNA computation as a new paradigm has the computational power of molecules for information processing and many computational models have been proposed for solving mathematical problems in laboratory experiments. In order to achieve the correct computation, a set good of DNA sequences is crucial, because the code determines the way to process information on sequences in the experiments. Much works have focused on designing the DNA sequences to archive a reliable molecular computation and many algorithms have been proposed to obtain a set of good DNA sequences. In this paper, Ant Colony System (ACS) is proposed to solve the DNA sequence design problem. ACS used some ants to get their solutions based on the pheromone in their colony. A model is prepared which consists of four nodes representing four DNA bases. The results of the proposed approach are compared with the other methods such as Genetic Algorithm.

Keywords: Ant Colony Optimization, Ant Colony System, DNA sequence optimization**.**

I. INTRODUCTION

DNA computation has been extensively researched as a new computation paradigm in recent ten years. It has the ability to perform calculation using specific biochemical reaction between different DNA bases by Watson-Crick complementary base pairing, and has a number of useful properties such as massive parallelism and a huge memory capacity [1].

Although there have been many achievements, DNA computing faces some hurdles due to the technological difficulty of handling biochemistry process. To overcome these drawbacks, some works have focused on the design of DNA sequences to reduce the possibility for illegal reactions [2].

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]. In 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 to the requirements of all laboratory experiments [3]. 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 [4].

Various kinds of methods and strategies for DNA sequence optimization have been proposed to date, such as template-map strategy [5], graph method [6], stochastic methods [7], and nearest-neighbour thermodynamic [8]. An Ant Colony Optimization (ACO) approach for DNA sequence design has been previously proposed [9], which used thermodynamic values as heuristic information. However, in DNA sequence design, since there is actually no information could be used as heuristic information, in this study, DNA sequences are designed based on Ant Colony System (ACS) without any heuristic information.

II. THE DNA SEQUENCE OPTIMIZATION

The objective of the DNA sequence optimization 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 paper, the objective functions and constraints from [9] are used. 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 in order to prevent secondary structure of a DNA sequence. Furthermore, two constraints, which are $GC_{content}$ and melting temperature, are used to keep uniform chemical characteristics.

DNA sequence optimization is actually a multi-objective optimization problem. However, the problem is converted into single-objective problem, which can be formulated as follows:

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

subjected to Tm and GCcontent constraints, where fi is the objective function for each i$\in$ {$H_{measure}$, *similarity, hairpin, continuity*}, and ω_i is the weight for each f_i. In this study, the weights are defined by the user.

III. ANT COLONY SYSTEM

The studies of natural systems and the models of these systems have been beneficial for solving difficult

and complex real-world problems. ACO, as one of the natural systems, is inspired by the behavior of ants in finding the shortest path from the nest to the food place.

There are several additional or modification algorithms of ACO have been proposed, such as ACS, which has achieved performance improvements through the introduction of new mechanisms based on ideas not included in the original AS [10] Those mechanisms are state transition rule, global updating rule, and local pheromone updating rule [11].

A. State transition rule

Difference between ACS and AS is in the decision rule used by the ants during the construction process. The ACS transition rule, also referred as a *pseudo-random-proportional* rule was developed to explicitly balance the exploration and exploitation abilities of the algorithm. In ACS the probability for an ant to move from city *i* to city *j* depends on a random variable *q* and q_0 likes shown in Eq. (2);

$$j = \begin{cases} \arg \max_{u \in J_k(r)} \left\{ [\tau(r,u)] \cdot [\eta(r,u)]^{\beta} \right\} & \text{if } q \le q_0 \\ S & \text{otherwise} \end{cases} \quad (2)$$

where *q* is a uniformly distributed random variable [0,1], q_0 value is between 0 and 1, and *S* is another random variable selected according to the probability distribution.

B. Global updating rule

The global pheromone update is applied at the end o f the each iteration by only one ant, which can be either the *iteration-best* or the *best-so-far*. The global updatin g rule is formulated as,

$$\tau(r,s) = (1-\rho) \cdot \tau(r,s) + \Delta\tau(r,s) \quad (3)$$

where ρ is the pheromone evaporation for global u pdating and $\Delta\tau(r,s)$ shown in Eq. (4).

$$\Delta\tau(r,s) = \begin{cases} 1/Q & \text{if } (r,s) \in \text{sequence done by ant } k \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where *Q* is the sum of all objective calculated for a sequence.

C. Local pheromone updating rule

The local pheromone update is performed by all the ants after each construction step using Eq. (5);

$$\tau(r,s) = (1-\zeta) \cdot \tau(r,s) + \tau_0 \quad (5)$$

where $\zeta \in [0, 1]$ is the pheromone decay coefficient, and τ_0 is the initial value of the pheromone and in Eq. (5);

$$\tau_0 = 1/C \quad (6)$$

where *C* is average values of *Q* for a set sequences, which are obtained randomly in the initialization process.

After that, each ant applied a state transition rule as defined in Eq. (2) to construct the solution until all ants have build a complete solution. After that, a local pheromone updating is applied. The objective function is calculated based on the problem being solved and then the global updating rule is applied based on Eq. (3) for all solutions.

IV. METHODOLOGY

The implementation of process in solving the DNA sequence design problem has been started by modeling the problem based on ACO methods. After that, the algorithms are developed to achieve the best solution of the problem.

In order to model the DNA sequence design problem into ACO methods, a model similar to finite state machine, which has four nodes, is proposed. In this model, the nodes represent A, C, G, and T of DNA bases. Every node is connected to each other, including its own node, as shown in Fig 1.

As illustrated in Fig 1, if an ant is placed (randomly) at node A (Fig 1a), and then if the ant moves to node T (Fig 1b), the formed path by the ant can be translated into 'AT' sequence of DNA. Next, if the ant moves from node T to node C (Fig 1c), the DNA sequence 'ATC' is formed. The tours of the ant continue until the number of required sequences has been produced.

Since DNA sequence design problem offers no information, which can be directly used as heuristic information, this model only uses pheromone information for ACO computations. Taillard and Gambardella [12], in their proposed approach, Fast Ant, also have used pheromone information only for Quadratic Assignment Problem (QAP).

During the initialization step, all parameters, such as α and ρ are set determined based on the default parameters for ACO [13] as presented in Table 1. The DNA parameters are initialized as listed in Table 2 [4].

In this paper, a multi-objective optimization problem is simplified into single-objective problem. Since it difficult to find the proper weight value to every objective [14], the weights in Eq. (1) are set as 1.

In the main process, every ant is placed randomly at the start node, at first. After that, every ant will be moving from one node to the other nodes to construct the DNA sequence. During the tour, the ant chooses the next node by applying the state transition rule, as in Eq. (2).

Since the required solution is a set of DNA sequences, a mechanism is needed to store the DNA sequence in an archive to be analyzed. The updating archive process is done only when the total of DNA

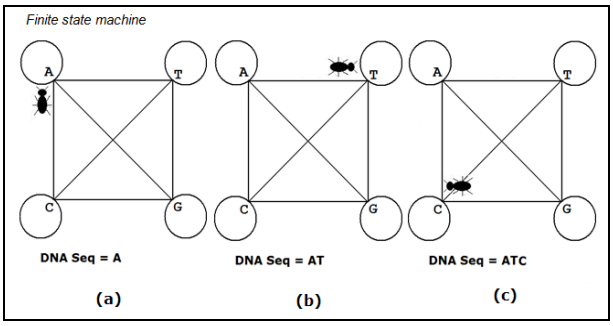

Fig 1. Finite state machine as a model for constructing a DNA sequence.

Table 1. Parameter for Ant Colony Optimization

Parameter	ACS
α	-
β	*0*
ζ	*0.1*
ρ	*0.1*
q_0	*0.9*
N	*half of ants*
Number of Sequences = 7 (*no. of ants- n_k*)	
Length of DNA Sequence = 20 (*no. of tours*)	
Max. Number of Iteration (*tmax*) = 500	

Table 2. DNA Sequence Parameter

DNA Sequence Parameter		
Parameter		**Value**
$h_{measure}$	h_{con}	6
	h_{dis}	0.17%
similarity	s_{con}	6
	s_{dis}	0.17%
t (continuity threshold)		2
hairpin	R_{min}	6
	P_{min}	6
GC%	*Min*	50
	Max	60
Tm	*Min*	40°C
	Max	80°C

The ACS algorithm for DNA sequence optimization is summarized in Algorithm 1;

Algorithm 1. Ant Colony System for DNA Sequence Optimization

// --- **Initialization step**
Initialize parameter t, α, ρ, q_0, n_k, N, and all DNA parameters, such as h_{con}, h_{dis}, s_{con}, s_{dis};
Calculate τ_o;
For each link(i, j) **do**
  $\tau(i, j) = \tau_o$; // --- **Pheromone initialize**
end
t= 0; // --- **initialize no of iteration**.
Repeat
  Repeat
    Place all ants, k =1, …, n_k; // --- (n_k = **number of ants**)
    For each ant k =1, …, n_k **do**
      Repeat // --- **State Transition Rule**
        Each ant applies a ***state transition rule*** (Eq. 34) to incrementally build a DNA sequence;
        A ***local pheromone updating rule*** (Eq. 32) is applied;
      Until all ants have build a complete a DNA sequence;
    Next
    If no. of DNA sequences in archive are equal with the number of ants (full) **Then**
      Calculate all DNA sequences in archive bases on four objective functions and then sort in descending order ;
      N-DNA sequences which have higher value in archive is removed;
    EndIf
    Calculate all DNA sequences bases on four objective functions and then sort in ascending order;
    For each DNA sequence **do**
      Checked for $GC_{content}$ and T_m constraints;
      If passed and archive is not full **Then** stored DNA sequence to archive;
    Next
    A ***global pheromone updating rule*** (Eq. 30) is applied;
  Until number of DNA sequences in archive are equal with the number of ants;
  $t = t + 1$;
// --- **maximum no. of (tmax) iteration is reached**.
Until End_Condition

sequences in archive is equal to number of ants (n_k). The process calculates the objective values for each DNA sequence and sorted them in descending order. The N-first worst DNA sequences will be selected and the next process, storing archive, remove them from the archive to be replaced by N new DNA sequences.

The storing archive process also calculates the objective values for each new DNA sequence and sorted them by ascending order. The DNA sequences are placed in the archive started from the smallest objective values, if the range of $GC_{content}$ and *melting temperature* constraints are satisfied. The process continues until the archive is full. In the last process, global updating rule is applied for all new DNA sequences.

V. RESULTS AND DISCUSSION

Based on the proposed model and algorithm, one hundred independent runs of the ACS approach for DNA sequence design have been executed, and average of over these runs reported. The comparison results between our result with previous work [9], and GA approach taken from the result of Deaton *et al* [15] as shown in Table 3 and Fig 2.

Since this optimization process is finding the minimum values for the objective function, the smallest value is the best. Also, since the multi-objective problem is converted into single objective problem, the overall results only considered by the total objective values.

Table 3 and Fig 2 show the new proposed approach obtained the much lower in total objective values than the GA [15] and ACS [9]. The new approach has quite lower in *continuity*, *hairpin*, and $H_{measure}$ objective

Table 3. The comparison result of ACS approach, ACS [9], and GA [15]

The Average value of objective for 100 times running of ACS approach					
	C	*Hr*	*Hm*	*Sm*	*Total*
Average	1.0	0.1	35.4	58.8	95.4
Standard Deviation	*1.2*	*0.3*	*5.0*	*4.5*	*2.31*
The DNA Sequences taken from [9]					
Average	0.0	0.0	54.1	51.9	106.0
Standard Deviation	*0.0*	*0.0*	*12.4*	*7.8*	*10.5*
The DNA Sequences taken from [15]					
Average	11.7	0.6	63.4	48.3	124.0
Standard Deviation	14.8	1.5	7.1	7.4	14.7

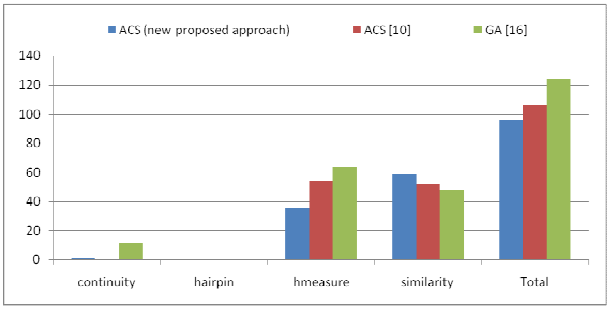

Fig 2. The comparison result of ACS approach, ACS [9], and GA [15]

values, but has higher in *similarity* than other results. For sequences generated by ACS [9], no *continuity* is observed, whereas the *continuity* value of sequences generated by the proposed approach and GA [15] are 1.0 and 11.7, respectively.

VI. CONCLUSION

ACS was implemented without heuristic information for DNA sequence optimization with four objective functions: $H_{measure}$, *similarity*, *continuity*, and *hairpin* and two constraints: $GC_{content}$ and *melting temperature*. The DNA sequences obtained from proposed approach were compared with those designed by previous work and GA approach. The results show that ACS can generate relatively better in total objective values than other approaches.

ACKNOWLEDGEMENT

This research is supported financially by the Ministry of Science, Technology, and Innovation (MOSTI), Malaysia, under eScienceFund Research Funding (Vot 79034) and the Ministry of Higher Education (MOHE), Malaysia, under Fundamental Research Grant Scheme (FRGS) (Vot 78225).

REFERENCES

[1] C.C. Maley, DNA computing: Theory, practice, and prospects, *Evol. Comp*, vol 6(3), 1998, pp. 201-229.

[2] A. Brenneman and A. Condon, Strand design for biomolecular computation, *Theory: Comput. Sci,* vol. 287, 2002, pp. 39-58.

[3] S. Kashiwamura, A. Kameda, M. Yamamoto, A. Ohuchi, Two-step Search for DNA Sequence Design, *Proceedings of the 2003 (ITC-CSCC 03)*, 2003, pp. 1815–1818.

[4] S.Y. Shin, I.H. Lee, D. Kim, B.T. Zhang, Multiobjective evolutionary optimization of DNA sequences for reliable DNA computing, *IEEE Transaction on Evolutionary Computation*, vol. 9(2), 2005, pp. 143-158.

[5] 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, *Proceedings of the 3rd DIMACS Workshop DNA Based Computers*, 1997.

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

[7] F. Tanaka, M. Nakatsugawa, M. Yamamoto, T. Shiba, and A. Ohuchi, Developing support system for sequence design in DNA computing, *Proceedings of the 7th International Workshop DNA Based Computers*, 2001, pp. 340–349.

[8] F. Tanaka, A. Kameda, M. Yamamoto, and A. Ohuchi, Thermodynamic parameter based on nearest-neighbor model for DNA sequences with a single-bulge loop, *Biochemistry*, vol. 43(22), 2004, pp. 7143–7150.

[9] T.B. Kurniawan, N.K. Khalid, Z. Ibrahim, M. Khalid, and M. Middendorf, An ant colony system for DNA sequence design based on thermodynamics, *Proceedings of IASTED-ACST*, 2008, pp. 144-149.

[10] M. Dorigo, Optimization, learning and natural algorithms, *PhD Thesis," Dipartimento di Elettronica, Politechico di Milano, Italy*, 1992.

[11] M. Dorigo, M.G. Luca, Ant Colony System: A cooperative learning approach to the Traveling Salesman Problem, *IEEE Trans. Ev. Com,* vol. 1, 1997.

[12] E.D. Taillard, L,-M. Gambardella, 1997, Adaptive memories for the Quadratic Assignment Problems, *Technical report* IDSIA-87-97, IDSIA, Lugano.

[13] M. Dorigo, and T. Stutzle, Ant Colony Optimization, *Massachusets Institute of Technology*, 2004.

[14] Y. Jin. M. Olhofer, B. Sendhoff, 2001. Dynamic Weighted Aggregation for Evolutionary Multi-Objective Optimization: How does it work and How?, *In Proceeding of GECCO 2001 Conference.*

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