DNA Computing Adopting DNA Coding Method for NP-complete Problems

Sang-yong Lee[*1] Hyo-gun Yun[*2] Eun- gyeong Kim[*3]

[*1] Division of Information & Communication Engineering, [*2,3] Dept. of Computer Engineering,

Kongju National University, Gongju, Korea

DNA computing suggests the potential of DNA with immense parallelism and huge storage capacity by solving NP-complete problems such as the Traveling Salesman Problem (TSP). If DNA computing is used to solve TSP, however, weights between vertexes in sequence design cannot be encoded effectively. This study proposes an algorithm for code optimization (ACO) that applies DNA coding method to DNA computing in order to encode weights in TSP effectively. By applying ACO to TSP, we designed sequence more effectively than Adleman's DNA computing algorithm. Furthermore, we could find the shortest path quickly and reduce biological error rate.

1. Introduction

In 1994, Adleman demonstrated that computing on a molecular level is available by solving the Hamiltonian path problem (HPP) using DNA's capacity of massive parallelism, data storage, and Watson-Crick's complementarity [Watson 92][Adleman 94]. Since then, there have been many researches on DNA computing using the characteristics of DNA to solve difficult problems such as NP-complete problems, which cannot be handled with existing computers [Jonoska 01].

However, when current DNA computing is applied to NP-complete problems such as TSP, the following three problems are raised [Rose 99]. First, it takes too much time and efforts to find solutions because it uses simple synthesis and separation process. Second, as it uses the operators of biological experiment methods, it contains the possibility of errors in experiments. Third, when converting graphs into DNA codes, it cannot fully reflect the characteristics of DNA.

There have been many studies to solve these three problems. The first problem was resolved through a repetitive process using genetic algorithm [Deaton 98]. To resolve the second problem, research to understand the mechanism of biological experiment methods has been carried out and achieved significant progresses [Yamamoto 99]. However, the third problem has not found a clear solution yet.

2. DNA Coding Method

DNA coding method is a variation of genetic algorithm proposed by Yoshikawa in 1995 [Yoshikawa 97]. General genetic algorithms use 0 and 1, but DNA coding method uses A(Adenine), G(Guanine), T(Thymine) and C(Cytosine). In addition, three codons out of A, T, G and C designate amino acid, a unit of meaning, and the number of sequences is 20 excluding redundant ones. The characteristics of DNA coding method are its efficient encoding of the redundancy of chromosomes and easy encoding of knowledge thanks to multiple codons forming an amino acid. Furthermore, the length of chromosomes is changeable because crossover points are given arbitrarily. Particularly when chromosome is long, the method of encoding changeable length is much more efficient and creates more various populations than the method of encoding fixed length. Such characteristics enable biologically closer model of the function and behavior of chromosomes.

3. ACO

ACO solved TSP by applying DNA coding method to DNA computing, and used effective DNA codes in encoding edges containing weights, which had been a problem in the application.

ACO is composed of the preprocess level and the postprocess level (Fig. 1).

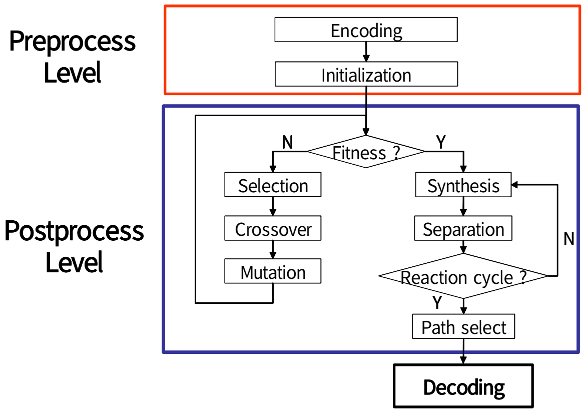

Fig.1 The Flow of ACO

Contact: Sang-yong Lee, Division of Information & Communication Engineering, Kongju National University, 182 Shingwan-dong, Kongju, Chungnam, Korea, +82-41-850-8523, sylee@kongju.ac.kr

First, the preprocess level is divided into the process of encoding method determination and that of initialization. In determining the encoding method, given DNA codes are converted into vertexes and edges containing weights through DNA coding method. Because vertexes and edges cannot be encoded directly, they are generated through the following sequence. First, the position of start codon(ATG) is identified, and DNA code of from the (i)th start codon position to the codon in front of the (i+1)th start codon position is expressed as a vertex. Then DNA code of from the (i+1)th start codon position to the codon in front of the (i+2)th start codon position is expressed as a weight (Fig. 2). However, if DNA code does not start with a start codon, a vertex is from the beginning of the DNA code to the codon in front of ith start codon position.

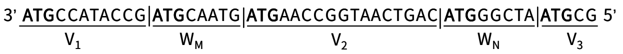

Fig. 2 Examples of vertexes and weights

Edges that connect vertexes encoded in this way are produced in the following sequence. First, designate AT* (ATT, ATC, ATA), which appears first in vertex V_i, as $E_{(i)}$ and stop codons TAA, TGA and TAG, which appear first in $V_{(i+1)}$ as $E_{(i+1)}$, and then encode an edge between the two vertexes (Fig. 3). If there is no stop codon, take DNA code of 1/2bp (base pair) of $V_{(i+1)}$ as the edge.

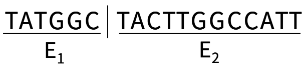

Fig. 3 Examples of edges

Then, as shown in Fig. 4, put a weight between edges and create a path by complementing only the weight.

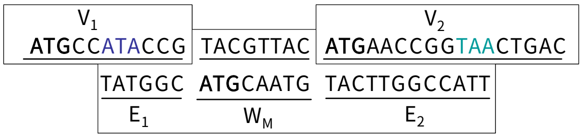

Fig. 4 An example of path creation containing a weight

$$F_i = \begin{cases} \left| \sqrt{\dfrac{Ne_i}{Sv} - \dfrac{We_i}{Sw}} \right| & if \left| \sqrt{\dfrac{Ne_i}{Sv} - \dfrac{We_i}{Sw}} \geq \theta \right| \\ otherwise\ is\ 0 \end{cases}$$ Equation (1)

Equation (1) is to obtain the weight of an edge using the value of hydrogen bond conversion function for edge i (Ne_i), the actual weight of edge i (We_i), the sum of weights in the entire graph (S_w), the sum of hydrogen bonds of all edges (S_v) and a threshold (θ) determined through experiments. That is, an edge containing a weight is generated by including the number of hydrogen bonds for the pair of A/T and that for the pair of G/C in the edge with a low weight and that with a high weight respectively. Using a weight conversion equation, we can adjust the length of DNA code in encoding weights, which significantly expands the scope of encoding weights and makes it possible to encode a wide range of weights with short codes. The process of deciding the encoding method as described above is followed by initialization, which translates DNA code into amino acid code based on the amino acid code table.

In the postprocess level of ACO, if the fitness is not satisfactory, it is re-evaluated using the operators of DNA coding method, which are selection, crossover and mutation. If it is satisfactory, a superior route that underwent synthesis and separation as many times as the reaction cycle is taken as the final solution.

Table 1. Amino acid code

Phe	16	Pro	3	His	15	Glu	13
Leu	7	Thr	5	Gln	11	Cys	6
Ile	8	Ala	1	Asn	9	Trp	19
Met	14	Tyr	18	Lys	12	Arg	17
Ser	2	Val	4	Asp	10	Gly	0

Fitness is determined by an inverse function of roulette wheel applying the amino acid code in Table 1. Conditions such as wrong synthesis or the shift of synthesis position, which may cause errors in biological experiment, are removed in advance. If fitness is not satisfactory, select DNA codes with the highest fitness and perform on them two-point crossover, which occurs only on the sequences of vertexes, and then select points of crossover at random. For mutation, select arbitrary base pairs among the sequences of vertexes and change one base pair of them, and repeat the process as many times as the number of generations.

Out of codes generated in this way, select those that meet the required level of fitness and perform synthesis and separation on them as many times as the given reaction cycle. In the process of separation, those that are unlikely to be solutions are removed in advance using biological operations such as antibody affinity reaction, polymerase chain reaction(PCR) and gel electrophoresis. Lastly, the sequences in specific parts are amplified using PCR once again. Then, a particular length of DNA sequence is abstracted with gel electrophoresis, and the path that passes all vertexes on the graph only once is selected as the final solution using antibody affinity.

4. Experiments

To verify the performance of ACO, we compared ACO with Adleman's DNA computing algorithm using TSP with 8 vertexes and 15 edges. The simulation was implemented in C language on a PC of 1GHz PⅣ and 256M RAM. Originally, Adleman's DNA computing algorithm is composed of just once with synthesis and separation process, but our study performed total reaction cycle of 1000 times (reaction cycle x max recycle) like ACO (Table 2). In addition, the DNA code was set to variable lengths in ACO, while it had been set to fixed lengths between 10bp~20bp in Adleman's DNA computing algorithm

Table 2. Parameters

parameter		ACO	Adleman's DNA computing algorithm
population size		1500	1500
generation		100	100
crossover rate		0.4	0.4
mutation rate		0.1	0.1
code length		0.1	0.1
total reaction cycle	max recycle	10	1
	reaction cycle	100	1000
error rate in biology experiment		0.01	0.01

It was confirmed that ACO could express DNA codes of variable lengths more efficiently than Adleman's DNA computing algorithm could. In addition, the length of time for search and error rate in biological experiment were lowered by around 38% and 56% respectively, which means the quicker and more accurate route search.

5. Conclusion

In this paper, we analyzed problems in solving TSP using DNA computing and proposed as a solution ACO adopted DNA coding method. The experiment confirmed that ACO was efficient in encoding the given number of hydrogen bonds as the weight of the edge. What is more, it could encode a wide range of weights using short DNA codes.

References

[Watson 92] J. D., Watson, et al., "Recombinant DNA," Scientific American Books, New York, 1992.

[Adleman 94] L. M., Adleman, "Molecular computation of solutions to combinatorial problems," Science, No. 266, pp.1021-1024, 1994.

[Jonoska 01] N. Jonoska & N. C. Seedman (Eds.), "Preliminary Proceedings of 7th International Meeting on DNA Based Computers," University of South Florida, pp. 10-13, 2001.

[Rose 99] J. A., Rose, et al., "A Statistical Mechanical Treatment of Error in the Annealing Biostep of DNA Computation," GECCO99, pp. 1829-1834, 1999.

[Deaton 98] R. Deaton, et al., "Reliability and efficiency of a DNA-based computation," Physical Review Letters, 82(2), pp. 417-420, 1998.

[Yamamoto 99] M. Yamamoto, et al., "A Study on the Hybridiztation Process In DNA Computing," DNA-V, pp. 99-108, 1999.

[Yoshikawa 97] T. Yoshikawa, T. Furuhashi, Y. Uchidawa. "The Effect of Combination of DNA Coding Method with Pseudo-Bacterial GA" Proceeding of the 1997 IEEE International Intermag. 97 Magnetics Conference 1997.