Int. Journ. of Unconventional Computing, Vol. 3, pp. 15–34
Reprints available directly from the publisher
Photocopying permitted by license only

©2007 Old City Publishing, Inc.
Published by license under the OCP Science imprint,
a member of the Old City Publishing Group

Experimental Implementation of Direct-Proportional Length-Based DNA Computing for Numerical Optimization of the Shortest Path Problem

ZUWAIRIE IBRAHIM[1,2], YUSEI TSUBOI[2], OSAMU ONO[2] AND MARZUKI KHALID[3]

[1]*Department of Mechatronics and Robotics, Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor Darul Takzim, Malaysia*
E-mail: zuwairie@fke.utm.my
[2]*Institute of Applied DNA Computing, Meiji University, 1-1-1 Higashi-mita, Tama-ku, Kawasaki-shi, Kanagawa-ken, 214-8571 Japan*
E-mail: {zuwairie, tsuboi, ono}@isc.meiji.ac.jp
http://www.isc.meiji.ac.jp/~i3erabc/IADC.html
[3]*Center for Artificial Intelligence and Robotics (CAIRO), Universiti Teknologi Malaysia, City Campus, Jalan Semarak, Kuala Lumpur, Malaysia*
E-mail: marzuki@utmkl.utm.my

Received: 1 September 2005; Accepted: 1 December 2005

DNA computing has emerged as an interdisciplinary field that draws together chemistry, molecular biology, computer science, engineering, and mathematics. From DNA computing point of view, it has been proven that it is possible to solve weighted graph problems such as Traveling Salesman Problem (TSP) and the shortest path problem by exploiting some characteristics of DNA. Those characteristics are length, concentration, and melting temperature of DNA. In this paper, we present an alternative length-based DNA computing approach whereby the cost of each path is encoded by the length of the oligonucleotides in a proportional way. The advantage is such that, after an initial pool generation and amplification, polyacrylamide gel electrophoresis (PAGE) can be performed to separate the respective DNA duplex according to their length which directly decodes the results. For an efficient initial pool generation, parallel overlap assembly (POA) is employed. After amplification is done by polymerase chain reaction (PCR), the result of the computation is visualized by PAGE. The experimental results show the effectiveness of the proposed direct-proportional

length-based computation and prove that the shortest path problem has been successfully solved on a DNA computer.

Keywords: DNA computing, length-based approach, optimization, the shortest path problem, hybridization/ligation, parallel overlap assembly

1 INTRODUCTION

Gordon E. Moore [1] has observed an exponential growth in the number of transistors per integrated circuit against time. This is the definition of Moore's Law, meaning that more and more transistors can be crammed into a single chip until the silicon itself reaches its finite atomic scale limitation. Since the traditional silicon based computer is restricted by its fundamental physical limitation, researchers have been searching for alternative medium for computation and DNA would turn out to be the answer.

A new computing paradigm based on DNA molecules has appeared in 1994 when Leonard M. Adleman [2] launched a novel *in vitro* approach to solve the so-called Hamiltonian Path Problem (HPP) with seven vertices by DNA molecules. The goal of the HPP is to determine whether a path exists that commences at the 'start city' and finishes at the 'end city', and passes through each of the remaining cities exactly once. While in conventional silicon-based computers, information is stored as binary numbers in silicon-based memories, in this novel approach, the information of the vertices is encoded by random DNA sequences. The computation is performed in bio-molecular reactions fashion which involves hybridization, denaturation, ligation, magnetic bead separation, and polymerase chain reaction (PCR). The output of the computation, also in the form of DNA molecules can be read and visualized by electrophoretical fluorescence operation.

Four years later, in 1998, a length-based DNA computing, which we called constant-proportional length-based DNA computing specifically for Traveling Salesman Problem (TSP) is proposed by Narayanan and Zorbalas [3]. A constant increase of DNA strands is encoded according to the actual length of the distance. A drawback of this method is that, there is a possibility of an occurrence of concatenated DNA strands of two distances which could be longer than the DNA strand of the longest distance that has been encoded. This may lead to errors in computing the shortest path [4]. This scheme, however, has not been realized by any laboratory experiment.

Yamamoto *et al*. carried out concentration-controlled DNA computing for accomplishing a local search for solving shortest path problem [5] by avoiding the generation of hopeless solutions. The avoidance is possible by encoding the numerical weights of edges by relative concentrations. Basically, the concentration is the highest for the smallest weight and smallest for the biggest weight, whereas the intermediate weights are transformed

into various concentrations relatively. Since the rate of biochemical reactions depends heavily on the reaction rate constant and reactant concentrations, as the concentration of DNA strands increase, the paths including them can be generated more frequently and the hopeful DNA paths can be generated with high concentration. The resultant solution is then subjected to denaturation gradient gel electrophoresis (DGGE) and constant denaturant gel electrophoresis (CDGE) [6] for separation and selection.

Lee *et al.* [7] proposed a DNA computing approach based on DNA melting temperature for solving TSP problem. Denaturation temperature gradient polymerase chain reaction (DTG-PCR) has been introduced where DNA duplex of correct solutions will be denatured and amplified by the PCR operation. As the denaturation temperature increases, other DNA strands will be also subsequently amplified. However, the amount of correct solutions will also be exponentially increased which does affect the final solution.

Due to the unsolved disadvantages, the constant-proportional length-based DNA computing has not yet been implemented in any laboratory experiments. Hence, with the aim to solve the limitation of the constant-proportional length-based approach, an alternative approach called direct-proportional length-based DNA computing is proposed. The shortest path problem has been selected for consideration of using the proposed technique. In this approach, the cost of an edge is encoded as a direct-proportional length oligos. After an initial pool generation and amplification, since numerous numbers of solution candidates are generated, by using the standard bio-molecular laboratory operations, it is possible to extract the optimal combination which represents the solution to the shortest path problem.

2 THE SHORTEST PATH PROBLEM

Even though the shortest path problem is belonging to the class P, such that, it is not hard to solve this problem, it is worth to be solved by DNA computing because numerical evaluations are required during the computation. The input to the shortest path problem is a weighted directed graph $G = (V, E, \omega)$, a start node u and an end node v. The output of the shortest path problem is a (u,v) path with the smallest cost. In the case given in Figure 1, if u is V_1 and v is V_5, the cost for the shortest path will be given as 100 and the optimal path is clearly shown as $V_1 - V_3 - V_4 - V_5$.

3 DNA AND BIO-MOLECULAR OPERATIONS

DNA molecules are composed of single or double DNA fragments or often called oligonucleotides (oligos, for short) or strands. Nucleotides form the

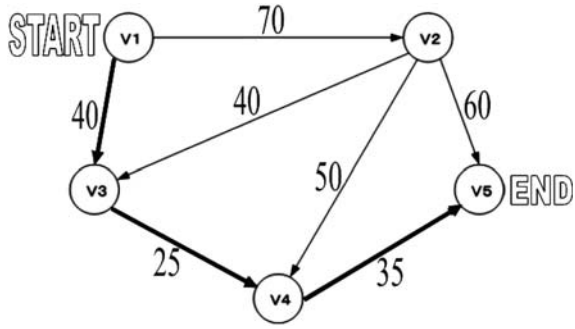

FIGURE 1
Example showing a weighted directed graph $G = (V, E)$ with the shortest path shown as $V_1 - V_3 - V_4 - V_5$.

basis of DNA. A single-stranded fragment has a phospho-sugar backbone and four kinds of bases denoted by the symbols A, T, G, and C for the bases adenine, thymine, guanine, and cytosine respectively. These four nucleic acids, which can occur in any order, combine in Watson-Crick complementary pairs to form a double strand helix of DNA. Due to the hybridization reaction, A is complementary with T and C is complementary with G. As an example, any sequence oligonucleotides, such as 5′-ACCTG-3′ has a complementary sequence, 3′-TGGAC-5′. Digits 5′ and 3′ denote orientation of DNA oligonucleotides.

In order to understand how to make DNA performs computation, it is required to study several biochemical reactions such as DNA synthesis, hybridization, denaturation, polymerization, PCR, and gel electrophoresis.

3.1 Synthesizing DNA

Short chemically synthesized single stranded molecules are called oligonucleotides or simply oligos. There are useful in genetic engineering as well as in DNA computing. Due to current technology, 70–80 sequences can be chemically synthesized without much error. At present, it is possible to get a test tube containing approximately 10^{18} DNA molecules with a desired sequence. Some commercial DNA synthesis companies are available, which provide a reasonable price for this reason.

3.2 Hybridization and denaturation

Hybridization is defined as a sequence-specific annealing of two or more single stranded DNAs, forming a dsDNA product. This sequence-recognition property is very useful for DNA computing because hybridization means computation, from DNA computing sense. This operation is normally caused by cooling down the test tube reaction solution [8].

There are basically three cases, on how the hybridization could occur: bi-molecular hybridization, multi-molecular hybridization, and uni-molecular

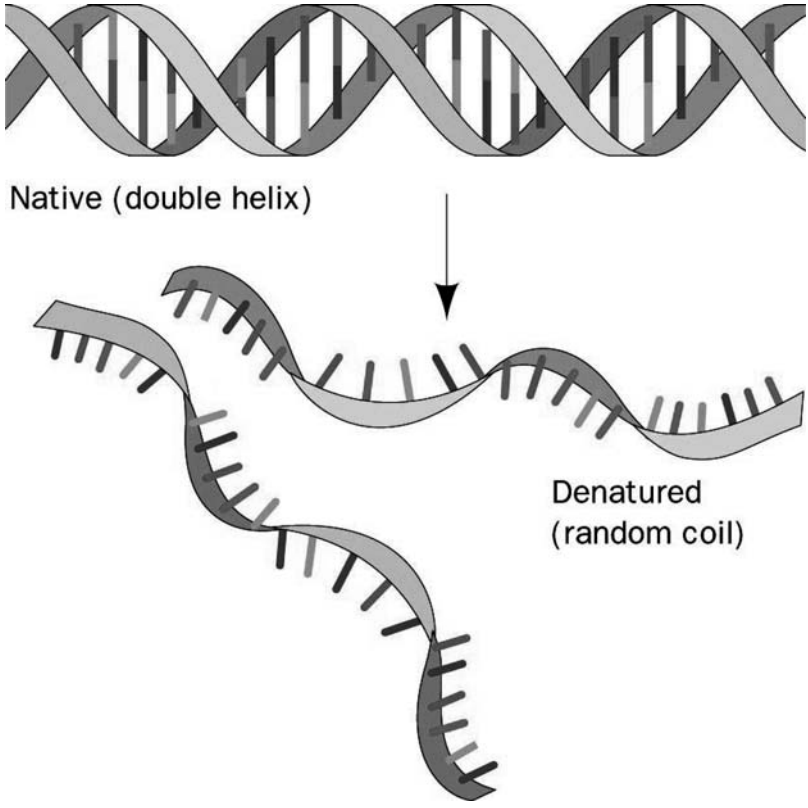

FIGURE 2
Bi-molecular hybridization and denaturation of DNA.

hybridization. For the first case, bi-molecular hybridization involves two kinds of single stranded DNAs (ssDNAs) as shown in Figure 2 [9]. For the multi-molecular hybridization, three strands are involved during the annealing. Multi-molecular hybridization is the essence of Adleman's type of DNA computing [2] for solving an instance of HPP. Thirdly, for uni-molecular hybridization or self-hybridization, a hairpin formation of ssDNAs could be formed if a complementary portion exists in the same ssDNAs.

By heating up the solution to about 85–95°C, double stranded DNAs (dsDNAs) will come apart because the hydrogen bonds between complementary nucleotides are much weaker than the covalent bonds between nucleotides adjacent in the two strands. The separation is called melting or denaturation. Thus, two strands can be separated without breaking the single strands [10], as depicted in Figure 2 [9]. The same effect can be achieved by washing the double stranded DNAs in doubly distilled water.

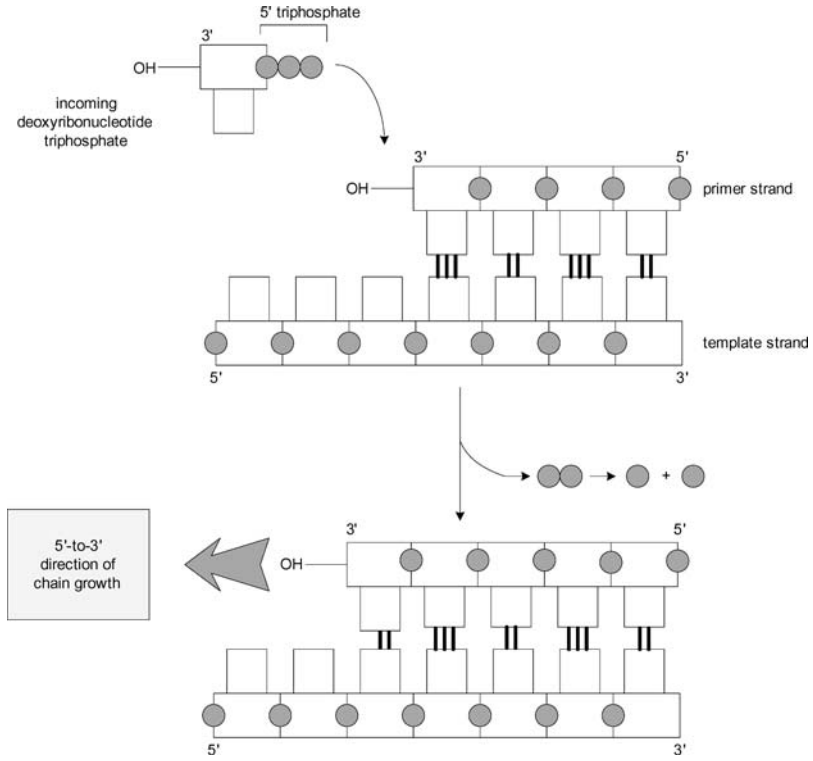

FIGURE 3
Polymerization in action.

3.3 Polymerization

The substrates required for polymerization are a template strand to be copied, a primer strand to be 3′-extended, incoming dNTP monomers, which act as both base and energy sources, and DNA polymerase. DNA polymerase implements a 5′ to 3′ copying operation as depicted in Figure 3 [11]. During the copying operation, 3′ end of a primer strand is extended. Note that there is no 3′ to 5′ copying operation ever observed. This operation also depends on Watson-Crick complementarity. In other words, A is copied to T and G is copied to C, and so on.

3.4 Polymerase chain reaction (PCR)

PCR is an incredible sensitive copying machine for DNA. It also can be used for DNA detection. Given a site-specific single molecule DNA, a million or even billion of similar molecules can be created by PCR process. In n steps, it can produce 2^n copies of the same molecules. PCR needs a number of sub-sequence strands called 'primers', which are usually about 20 base long to signal a specific start and end site at a

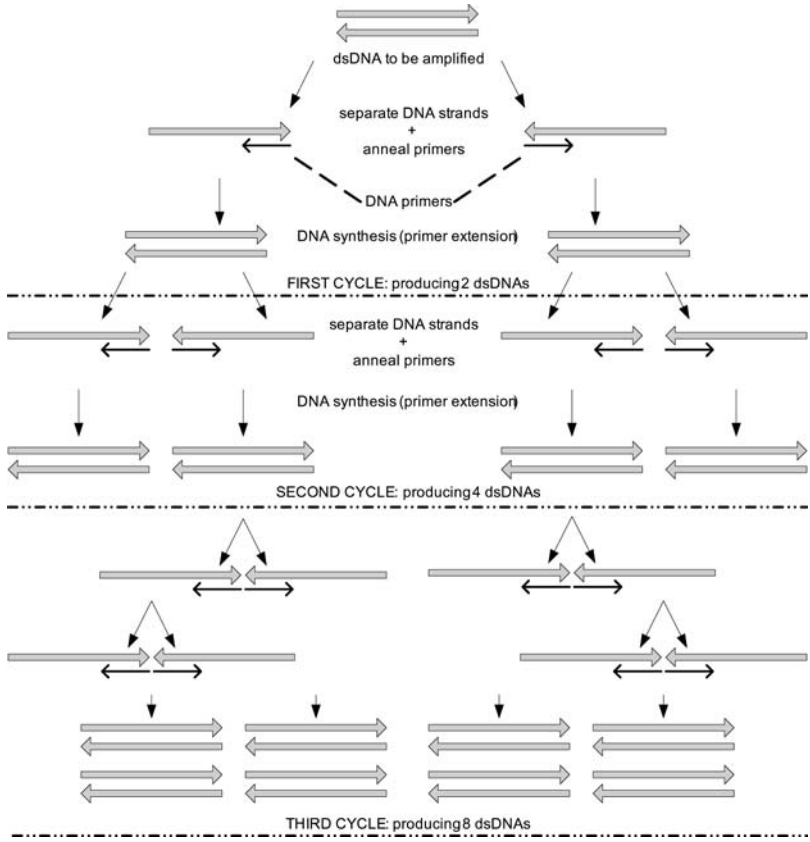

FIGURE 4
Polymerase chain reaction.

template for replication. PCR normally runs for 20–30 cycles of 3 phases: separating base pair strands of DNA at about 95°C, annealing at 55°C, and extension at 74°C [12]. It takes about two to three hours normally in order to complete the cycles. Figure 4 shows the operations of PCR up to third cycles, whereas, an example of thermal cycle of PCR is shown in Figure 5.

3.5 Gel electrophoresis

DNA strands in a solution can be separated in terms of its length by means of gel electrophoresis. In fact, the molecules are separated according to their weight, which is almost proportional to their length [10]. This technique relies on the fact that DNA molecules are negatively charged [13]. Hence, by putting them in an electric field, they will move towards the positive electrode at different speed. If electrical field is applied through

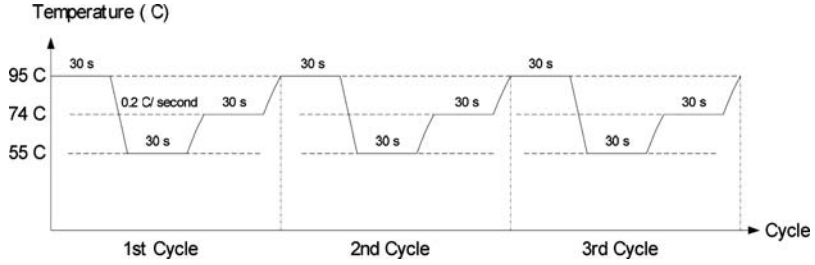

FIGURE 5
Example of a thermal cycle of PCR.

the gel, longer molecules will remain behind the shorter ones, as shown in Figure 6 [14]. The speed of DNA mixture in a gel depends heavily on the gel porosity and the magnitude of the electrical field. Polyacrylamide gel is used for separation of shorter dsDNAs, which is from 10 base-pairs (bps) until 500 bps. On the other hand, agarose gel is frequently used for longer dsDNAs, which is more than 500 bps. An example of the output of gel electrophoresis is well depicted in Figure 7. In DNA computing, this technique is used to visualize the results of computation. Normally, at the end of this process, the gel is photographed for convenience.

4 DNA SEQUENCE DESIGN AND SYNTHESIS

Consider a directed graph and the output of the shortest path computation as shown in Figure 1. Let n be the total number of nodes in the graph. The DNA sequences correspond to all nodes and its complements are designed. Let $V_i(i = 1, 2, \ldots, n)$ and $\overline{V_i}(i = 1, 2, \ldots, n)$ be the 20-mer DNA sequences correspond to the ith node in the graph and its complement respectively. By using the available software for DNA sequence design, DNASequenceGenerator [15], the DNA sequences V_i is designed and listed in Table 1. Melting temperature, T_m is calculated based on Sugimoto nearest neighbor thermodynamic parameter [16]. The GC contents (GC%) and melting temperature (T_m) of each sequence are also shown. Table 2, on the other hand, shows the complement of the node sequences.

TABLE 1
DNA sequences for nodes

Node, V_i	20-mer Sequences (5′–3′)	GC%	Melting Temperature, T_m (°C)
V_1	AAAGCTCGTCGTTTAGGAGC	50	60.9
V_2	GCACTAGGGATTTGGAGGTT	50	60.3
V_3	GCTATGCCGTAGTAGAGCGA	55	60.5
V_4	CGATACCGAACTGATAAGCG	50	60.6
V_5	CGTGGGTGGCTCTGTAATAG	55	60.5

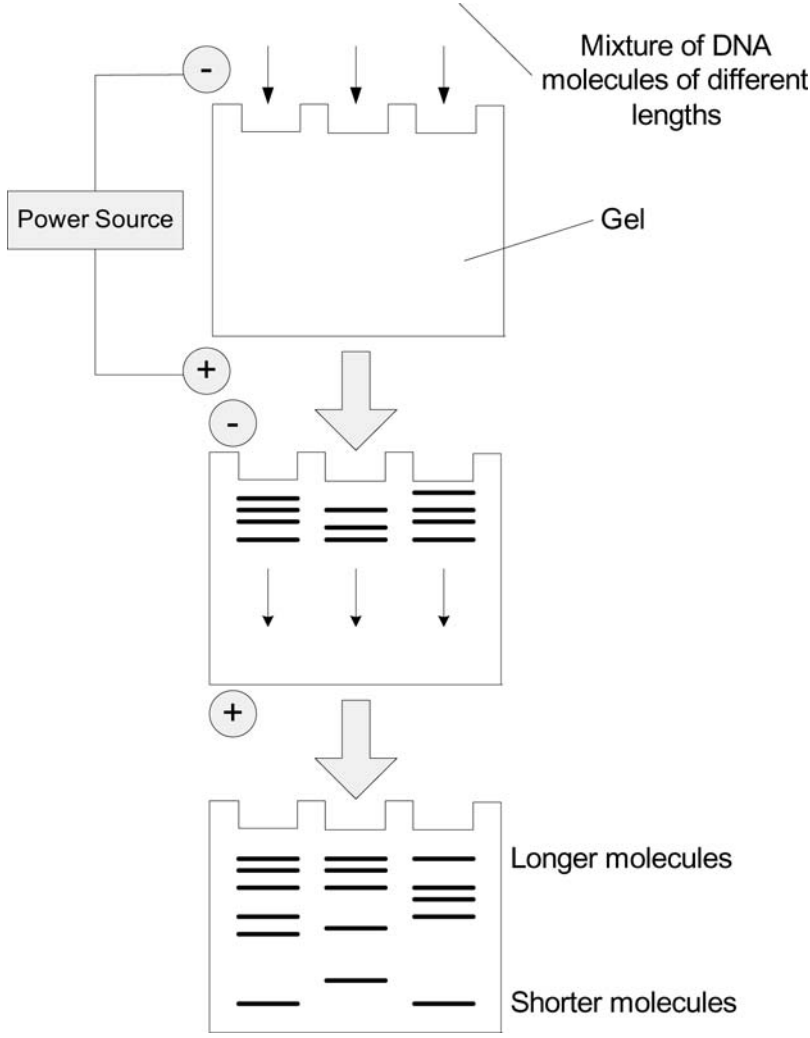

FIGURE 6
Polyacrylamide gel electrophoresis (PAGE).

TABLE 2
Complement node

Complement Node, $\overline{V_i}$	20-mer Complement Sequences (3′-5′)
$\overline{V}_1$	TTTCGAGCAGCAAATCCTCG
$\overline{V}_2$	CGTGATCCCTAAACCTCCAA
$\overline{V}_3$	CGATACGGCATCATCTCGCT
$\overline{V}_4$	GCTATGGCTTGACTATTCGC
$\overline{V}_5$	GCACCCACCGAGACATTATC

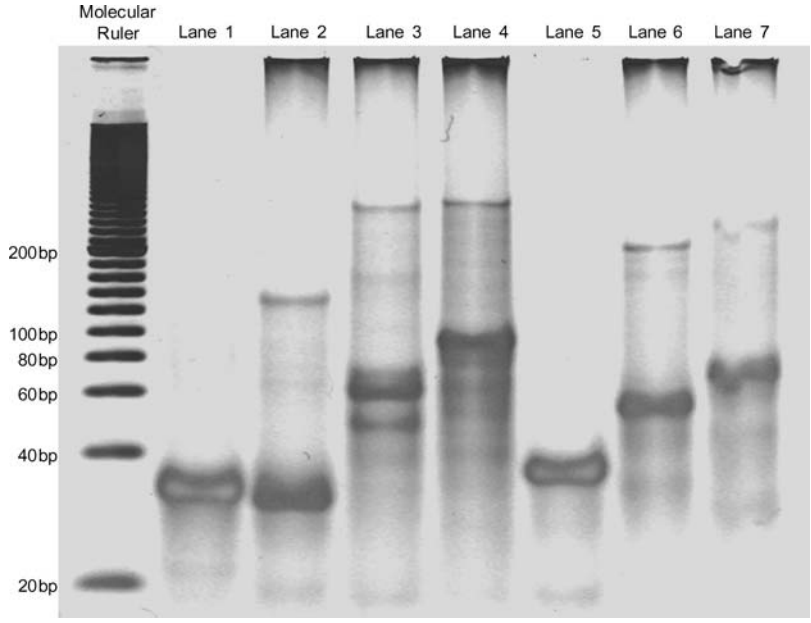

FIGURE 7
An example of the output of gel electrophoresis.

We introduce three rules to synthesize oligos for each edge in the graph as follows:

(i) If there is a connection between V_1 to V_j, where $j \neq n$, synthesize the oligo for edge as $V_1(20) + W_{1j}\,(\omega - 30) + V_j(20)$
(ii) If there is a connection between V_i to V_j, where $i \neq 1$, $j \neq n$, synthesize the oligo for edge as $V_i(20) + W_{ij}\,(\omega - 20) + V_j(20)$
(iii) If there is a connection between V_i to V_n, where $i \neq 1$, synthesize the oligo for edge as $V_i(20) + W_{in}\,(\omega - 30) + V_n(20)$

where V, W, and '+' denote the DNA sequences for nodes, DNA sequences for weight, and 'join' respectively. The synthesized oligos consist of three segments; two node segments and an edge segment. 'ω' denotes the weight value for corresponding DNA sequences for weight W_{ij}, where W_{ij} denotes the DNA sequences representing a cost between node V_i and V_j. The value in parenthesis indicates the number of DNA bases or nucleotides for each segment. The oligo is synthesized so that the number of DNA bases of that oligo and the cost at the corresponding edge are similar. Table 3 lists all the synthesized oligos based on the proposed synthesis rules. Again, DNASequenceGenerator [15] is employed. The node segment and edge segment are distinguished by capital and small letters respectively. The complement sequences of each node are synthesized as well.

TABLE 3
DNA sequences for edges

Edge	DNA Sequences
$V_4-W_{45}-V_5$	5′-CGATACCGAACTGATAAGCGccaagCGTGGGTGGCTCTGTAATAG-3′
$V_3-W_{34}-V_4$	5′-GCTATGCCGTAGTAGAGCGAccgtcCGATACCGAACTGATAAGCG-3′
$V_1-W_{13}-V_3$	5′-AAAGCTCGTCGTTTAGGAGCacgtcggttcGCTATGCCGTAGTAGAGCGA-3′
$V_2-W_{23}-V_3$	5′-GCACTAGGGATTTGGAGGTTccgtcttttacccaagtaat GCTATGCCGTAGTAGAGCGA-3′
$V_2-W_{24}-V_4$	5′-GCACTAGGGATTTGGAGGTTacgtgttttaaggaagtacggtaagctgcg CGATACCGAACTGATAAGCG-3′
$V_2-W_{25}-V_5$	5′-GCACTAGGGATTTGGAGGTTgcgtcgcgtaaggcagtaccggactctgcc CGTGGGTGGCTCTGTAATAG-3′
$V_1-W_{12}-V_2$	5′-AAAGCTCGTCGTTTAGGAGCcggtggtttaacgaagtcctgtactatgggttatttgcag GCACTAGGGATTTGGAGGTT-3′

5 DIRECT-PROPORTIONAL LENGTH-BASED DNA COMPUTING

Currently, there are two kind of initial pool generation methods for solving weighted graph problem: hybridization/ligation and parallel overlap assembly (POA). The hybridization/ligation method has been firstly introduced by Adleman [2] to solve a HPP. For hybridization/ligation method, during the operation, the link oligos hybridize through the hydrogen bonds by enzymatic reaction. The hybridization/ligation reaction is well shown in Figure 2 [8].

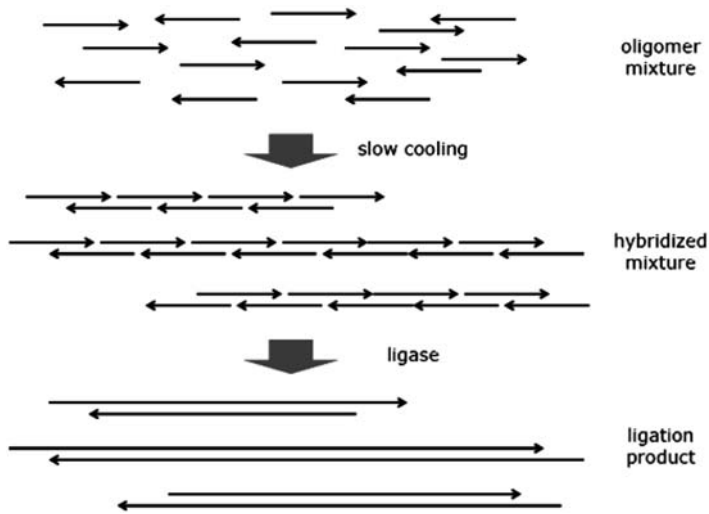

FIGURE 8
Hybridization/ligation method for initial pool generation. The arrowhead indicates the 3′ end.

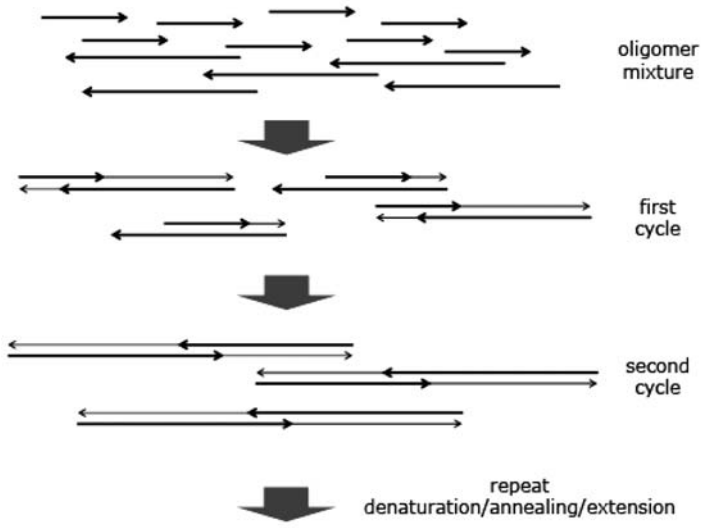

FIGURE 9
POA for an initial pool generation. The thick arrows represent the synthesized oligos which are the input to the computation. The thin arrows represent the elongated part during polymerization. The arrowhead indicates the 3′ end.

POA has been used [18] and broadly applied in gene construction [19–21], gene reconstruction [22], and DNA shuffling [23]. POA involves thermal cycle and during the thermal cycle, the position strings in one oligo anneals to the complementary strings of the next oligo. The 3′ end side of the oligo is extended in the presence of polymerase enzyme to form a longer dsDNA. One cycle of POA is depicted in Figure 9 [17]. After a number of thermal cycles, a data pool with all combinations could be built.

Recently, Lee *et al.* [17] did a comparison between hybridization/ligation method and POA for initial pool generation of DNA computing. They came out with a conclusion that for the initial pool generation of weighted graph problems, POA method is more efficient than that of hybridization/ligation method. According to [17], the advantages of POA over hybridization/ligation method for initial pool generation are as follows:

(i) The initial pool size generated from the same amount of initial oligos is about twice larger than that of hybridization/ligation method. Though, if a larger problem is considered, the initial pool size is too small to contain the complete pool. POA, however, with more cycle and large experimental scale could include the practical pools.
(ii) Initially, two single-stranded DNA molecules partially hybridize in the annealing step and then they are extended by polymerase. The elongated DNA molecules are denatured to two single-stranded DNA in the next denaturation step, and they are subjected to the

annealing reaction at the next cycle. Therefore, POA does maintain the population size and the population size can be decided by varying the initial number of oligos.

(iii) In hybridization/ligation method, the population size decreases as reaction progress. The population size decreased by a factor of the number of components composing it in hybridization/ligation method. As the problem size increases, the required initial pool size increases dramatically. Moreover, initial pool generation by POA requires fewer strands than hybridization/ligation method to obtain similar amount of initial pool DNA molecules because complementary strands are automatically extended by polymerase.

(iv) POA does not require phosphorylation of oligos which is prerequisite for the ligation of oligos.

(v) POA demands less time than hybridization/ligation method. Hybridization requires one and half hour while ligation required more than 12 hours. Hence, POA for 34 cycles requires only two hours. Therefore, POA is much more efficient and economic method for initial pool generation.

As stated in [4], "*In addition, the fact that larger weights are encoded as longer sequences is contrary to the biological fact that; the longer the sequences are, the more likely they hybridize with other DNA strands, though we have to find the shortest DNA strands*". From the biological point of view, this argument is definitely true. In order to overcome the limitation of general length-based DNA computing, the authors discovered that by utilizing POA for initial pool generation, a phase where numerous combinations of random routes of the graph are generated in the solution, a shortcoming, which is the biological influence contributed by the length of the oligos could be eliminated.

In order to generate the initial pool of the direct-proportional length-based DNA computing for the example problem by using POA method, the input to the computation are all the synthesized oligos as listed in Table 3 and the complement sequences for each nodes, which are listed in Table 2. These inputs are poured into a test tube and the cycles begin. In fact, the operation of POA is similar as PCR but the difference is that POA operates without the use of primers. As PCR, one cycle consists of three steps: denaturation, hybridization, and extension.

At this stage, an initial pool of solution has been produced and it is time to filter out the optimal combinations among the vast alternative combinations of the problem. Unlike conventional filtering, this process is not merely throwing away the unwanted DNA duplex but rather copying the target DNA duplex exponentially by using the incredibly sensitive PCR process. This can be done by amplifying the DNA duplex that contain the start node V_1 and end node V_5 using primers. After the PCR operation is

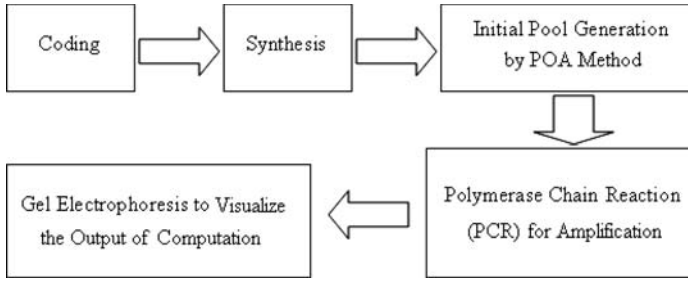

FIGURE 10
The overall procedures of direct-proportional length-based DNA computing.

accomplished, there should be numerous number of DNA strands representing the start node V_1 and end node V_5 traveling through a possible number of nodes.

The output solution of the PCR operation then undergoes gel electrophoresis operation. During this operation, the DNA molecules can be separated in terms of its length and hence, the DNA duplex $V_1 - V_3 - V_4 - V_5$ representing the shortest path starting from V_1 and ending at V_5 can be visualized. The overall procedure of direct-proportional length-based DNA computing is given in Figure 10.

6 EXPERIMENTAL PROTOCOLS, RESULTS, AND DISCUSSIONS

The initial pool generation by POA is performed in a 100 μl solution containing 12 μl oligos (Proligo Primers & Probes, USA), 10 μl dNTP (TOYOBO, Japan), 10 μl 10 x KOD dash buffer (TOYOBO, Japan), 0.5 μl KOD dash (TOYOBO, Japan), and 67.5 μl dd$H_2$0 (Maxim Biotech). The reaction consists of 25 cycles and for each cycles, the appropriate temperature are as follow:

- 94°C for 30s
- 55°C for 30s
- 74°C for 10s

The product of POA is shown in Figure 11. According to the gel image, the band in lane 1 is blurs and thus, it is expected that all the candidate answers are successfully generated. In order to select the paths that begin at V_1 and end at V_5, DNA amplification is done by employing PCR. The PCR is performed in a 25 μl solution consists of 0.5 $\mu l$for each primers, 1 μl template, 2.5 μl dNTP (TOYOBO, Japan), 2.5 μl 10x KOD dash buffer (TOYOBO, Japan), 0.125 μl KOD dash (TOYOBO, Japan), and

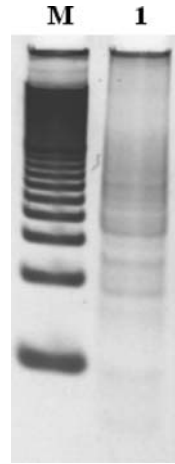

FIGURE 11
Experimental results of gel electrophoresis on 10% polyacrylamide gel. Lane M denotes 20-bp ladder and lane 1 is the product of POA.

15.875 μl ddH_20 (Maxim Biotech). The reaction consists of 25 cycles and for each cycles, the appropriate temperature are as follow:

- 94°C for 30s
- 55°C for 30s
- 74°C for 10s

which is the same as POA. The sequences used as primers are AAAGCTCGTCGTTTAGGAGC (V_1) and GCACCCACCGAGACATTATC ($\overline{V_5}$).

In order to visualize the result of the computation, the product of PCR is subjected to polyacrylamide gel electrophoresis for 40 minutes at 200 V. After that, gel electrophoresis, the gel is stained by SYBR Gold (Molecular Probes) and the gel image is captured. Figure 12 shows the product of PCR. Lane 1 consists of four bands showing that all the path that start at V_1 and end at V_5 have been successfully amplified. Those paths are $V_1 - V_3 - V_4 - V_5$ (100 bp), $V_1 - V_2 - V_5$ (130 bp), $V_1 - V_2 - V_4 - V_5$ (155 bp), and $V_1 - V_2 - V_3 - V_4 - V_5$ (165 bp). Clearly, the amplified paths have been sorted in term of length by gel electrophoresis and the output of the shortest path computation appears as the shortest band in lane 1.

7 DISCUSSIONS

According to the gel image of Figure 12, it is clear that *in vitro* computation is able to produce several shortest paths during the computation. In this case, up to four shortest paths are produced and visualized by PAGE.

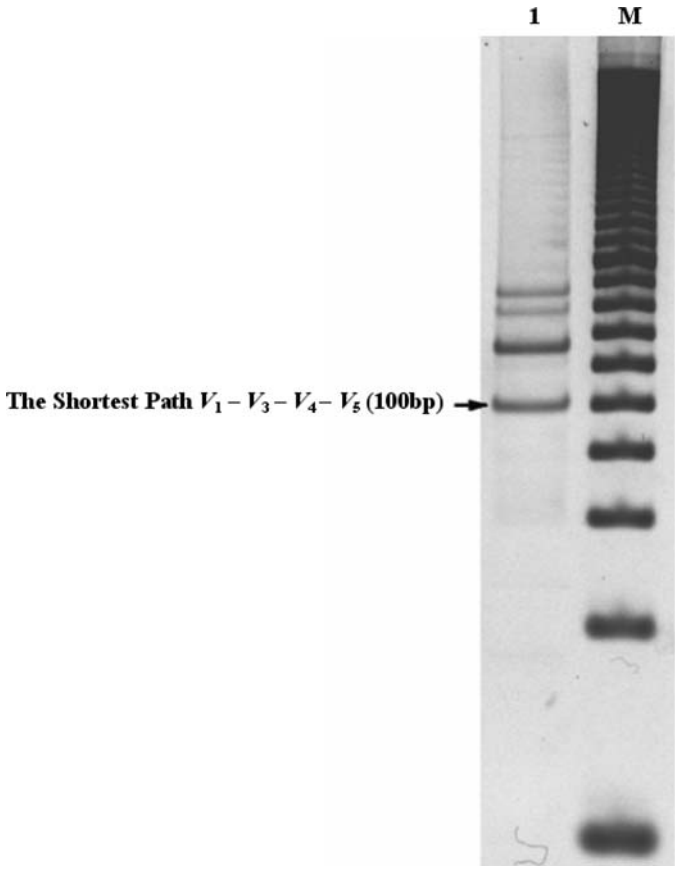

FIGURE 12
Computation output on 10% polyacrylamide gel. Lane M denotes 20-bp ladder and lane 1 is the product of PCR.

Since four kinds of paths including the shortest path can be successfully separated and listed by PAGE, this is the main evidence that the proposed DNA computer is able to perform the shortest paths computation *in vitro*.

Scaling is certainly the main problem of DNA computing, especially for generate-and-test DNA computing. In order to extend the proposed approach to a larger problem, two issues should be considered: molecular's weight and the capability to select the final solution. As an example, if Adleman's work for solving HPP is further examined, a 70-node problem requires 10^{25} kg of nucleotides, and this is quite a lot for a small test tube [24]. Hence, an advanced high reaction facility, such as microreactor [25], is highly important. In this research, we showed how to improve the scalability of our approach in two steps. The first step is during initial pool

generation, where POA is more preferred rather than hybridization/ligation. This is mainly because, as an example, POA is able to generate two times bigger initial pool, in term of size, than that of hybridization/ligation, by using the same amount of input molecules. The second step of improvement on scalability is during the computation, in term of the amount of DNA used for the computation. By using POA, only two kinds of oligos required for computation, which are the oligos for nodes and edges, whereas, in the case of hybridization/ligation, another one kind of oligos that is the oligos for weights are also indispensable for the computation.

In DNA computing for weighted graph problems, after the *in vitro* computation, a subsequent reactions or bio-molecular operations should be employed in order to detect the final solution. As an example, for the concentration-controlled DNA computing, separation as DGGE and CDGE should be used, whereas, in temperature gradient based DNA computing, another separation method, that is DT-PCR should be performed. This operations are relatively complicated than normal gel electrophoresis. In our approach, the adopted protocol for detecting the final solution is simple, where a simple PAGE is already enough to visualize the result of the computation.

The direct-proportional length-based DNA computing is proposed essentially to overcome the shortcoming of constant proportional length-based DNA computing approach. However, by using this approach, the minimum weight of edges that can be encoded is limited and the weight falls in a very narrow range. This is because, the length of the solution is not only proportional to the length of the path it encodes but it also includes the number of vertices in the path. Hence, the lower bound, in term of minimum weight that can be encoded by direct-proportional length-based DNA computing is analyzed. Basically, according to the proposed DNA synthesis rules, the lower bound is achieved when:

$$\omega - \frac{3}{2}\beta = 0 \tag{1}$$

Hence, the minimum weight, $\omega_{\min}$, which can be encoded by oligos is attained as:

$$\omega_{\min} = \frac{3}{2}\beta \tag{2}$$

where β is the number of nucleotides used to encode each node of the input graph [26].

8 CONCLUSIONS

We have presented a new alternative approach called 'direct-proportional length-based approach' to solve weighted graph problems using DNA computing. Based on this approach, it is proposed that the directly proportional length of DNA could be used to encode the cost of each edge. After the initial pool generation and amplification, the DNA duplex is subjected to gel electrophoresis for the separation in term of length. It has been shown that the shortest path is represented by the shortest length of DNA duplex. For the sake of initial pool generation, two kinds of methods are reviewed: hybridization/ligation and POA. For a successful demonstration of direct-proportional length-based DNA computing, we found that POA for initial pool generation is critically important. Finally, it is expected that the proposed approach, would extend the applicability of DNA computing for solving intractable weighted graph problems.

ACKNOWLEDGEMENTS

This research was supported partly by the IEEE Computational Intelligence Society (CIS) Walter J Karplus Student Summer Research Grant 2004 for a research visit in September 2004 at the DNA Computing Laboratory, Graduate School of Information Science and Technology, Hokkaido University, Sapporo, Hokkaido, Japan. The first author would like to thank Masahito Yamamoto for discussions that led to improvements in this work and also the permission to practice various kinds of biochemical experiments in the laboratory. Also, the first author is sincerely grateful to Atsushi Kameda, Satoshi Kashiwamura, and members of DNA Computing Laboratory of Hokkaido University for fruitful explanations and kind assistance during the practice of biochemical experiments, and anonymous reviewers for their important comments. Lastly, the first author is very thankful to Universiti Teknologi Malaysia (UTM) for granting a study leave in Meiji University under SLAB-JPA scholarship.

REFERENCES

[1] Moore, G. E. (1965). Cramming more components onto integrated circuits, *Electronics, 38*(8).

[2] Adleman, L. (1994). Molecular computation of solutions to combinatorial problems, *Science, 266*, 1021–1024.

[3] Narayanan, A. and Zorbalas, S. (1998). DNA algorithms for computing shortest paths, *Proc. of Genetic Programming*, 718–723.

[4] Lee, J. Y., Shin, S. Y., Augh, S. J., Park, T. H. and Zhang, B. T. (2003). Temperature gradient-based DNA computing for graph problems with weighted edges, *Lecture Notes in Computer Science, 2568*, 73–84.

[5] Yamamoto, M., Kameda, A., Matsuura, N., Shiba, T., Kawazoe, Y. and Ohuchi, A. (2002). A separation method for DNA computing based on concentration control, *New Generation Computing, 20*, 251–262.

[6] Yamamoto, Y., Kameda, A., Matsuura, N., Shiba, T., Kawazoe Y. and Ohuchi, A. (2002). Local search by concentration-controlled DNA computing, *International Journal of Computational Intelligence and Applications, World Scientific, 2*, 447–455.

[7] Lee, J. Y., Shin, S. Y., Augh, S. J., Park, T. H. and Zhang, B. T. (2002). Temperature gradient-based DNA computing for graph problems with weighted edges, *Preliminary Proceedings of the Eighth International Meeting on DNA Based Computers*, 41–50.

[8] Ausubel, F. and Struhl, K. (1995). Short protocol in molecular biology: a compendium of methods from current protocols in molecular biology, Third Edition, Wiley & Sons.

[9] School of Medicine, Southern Illinois University, October 6, 2005, http://web.siumed.edu/∼bbartholomew/images/chapter5/F05-14.jpg

[10] Calude, C. S. and Paun, G. (2001). Computing with cells and atoms: an introduction to quantum, DNA, and membrane computing, New York: Taylor & Francis Inc.

[11] Albert, B., Johnson, A., Lewis, J., Raff, M., Roberts, K. and Walter, P. (2002). Molecular biology of the cell, Fourth Edition, Garland Science.

[12] Fitch, J. P. (2002). An engineering introduction to biotechnology, SPIE.

[13] Paun, G., Rozenberg, G. and Salooma, A. (1998). DNA computing: new computing paradigms, New York: Springer.

[14] Amos, M. (1997). DNA computation, PhD Thesis, UK: The University of Warwick.

[15] Udo, F., Sam, S., Wolfgang, B. and Hilmar, R. (2001). DNASequenceGenerator: a program for the construction of DNA sequences, *Proceedings of the Seventh International Workshop on DNA Based Computers*, 23–32.

[16] Sugimoto, N., Nakano, S., Yoneyama, M. and Honda, K. (1996). Improved thermodynamic parameters and helix initiation factor to predict stability of DNA duplexes, *Nucleic Acid Research, 24*, 4501–4505.

[17] Lee, J. Y., Lim, H. W., Yoo, S. I., Zhang, B. T. and Park, T. H. (2004). Efficient initial pool generation for weighted graph problems using parallel overlap assembly, *Preliminary Proceedings of the Tenth International Meeting on DNA Based Computers*, 357–364.

[18] Kaplan, P. D., Ouyang, Q., Thaler, D. S. and Libchaber, A. (1997). Parallel overlap assembly for the construction of computational DNA libraries, *Journal of Theoretical Biology, 188*(3), 333–341.

[19] Ho, S. N., Hunt, H. D., Horton, R. M., Pullen, J. K. and Pease, L. R. (1989). Site-directed mutagenesis by overlap extension using the polymerase chain reaction, *Gene, 77*, 51–59.

[20] Jayaraman, K., Fingar, S. A. and Fyles, J. (1991). Polymerase chain reaction-mediated gene synthesis: synthesis of a gene coding for isozymec of horseradish peroxidase, *Proc. Natl. Acad. Sci. U.S.A., 88*, 4084–4088.

[21] Stemmer, W. P., Crameri, A., Ha, K. D., Brennan, T. M. and Heyneker, H. L. (1995). Single-step assembly of a gene and entire plasmid from large numbers of oligodeoxyribonucleotides, *Gene, 164*, 49–53.

[22] DeSalle, R., Barcia, M. and Wray, C. (1993). PCR jumping in clones of 30-million-year-old DNA fragments from amber preserved termites, *Experientia, 49*, 906–909.

[23] Stemmer, W. P. (1994). DNA shuffling by random fragmentation and reassembly: in vitro re-combination for molecular evolution, *Proc. Natl. Acad. Sci. U.S.A., 91*, 10747.

[24] Zucca, M. (2000). DNA based computational models, Ph.D. Thesis, Italy: Politecnico di Torino.

[25] Noort, D., Gast, F. U. and McCaskill, J. S. (2001). DNA computing in microreactors, *Lecture Notes in Computer Science, 2340*, 33–45.

[26] Ibrahim, Z., Tsuboi, Y., Ono, O. and Khalid, M. (2004). A study on lower bound of direct-proportional length-based DNA computing for shortest path problem, *Lecture Notes in Computer Science, 3314*, 71–76.