Enzyme Optimization For Next Level Molecular Computing

Piotr Wąsiewicz[a], Michal Malinowski[a]
Andrzej Plucienniczak[b]

[a]Institute of Electronic Systems, Warsaw
[b]Institute of Biotechnology and Antibiotics, Warsaw

ABSTRACT

The main concept of molecular computing depends on DNA self-assembly abilities and on modifying DNA with the help of enzymes during genetic operations. In the typical DNA computing a sequence of operations executed on DNA strings in parallel is called an algorithm, which is also determined by a model of DNA strings. This methodology is similar to the soft hardware specialized architecture driven here by heating, cooling and enzymes, especially polymerases used for copying strings. As it is described in this paper the polymerase Taq properties are changed by modifying its DNA sequence in such a way that polymerase side activities together with peptide chains, responsible for destroying amplified strings, are cut off. Thus, it introduces the next level of molecular computing. The genetic operation execution succession and the given molecule model with designed nucleotide sequences produce computation results and additionally they modify enzymes, which directly influence on the computation process. The information flow begins to circulate. Additionally, such optimized enzymes are more suitable for nanoconstruction, because they have only desired characteristics. The experiment was proposed to confirm the possibilities of the suggested implementation.

Keywords: molecular computing, DNA computer, enzyme cloning, genetic engineering

1. INTRODUCTION

Recent research results rise a hope for very great miniaturization in the field of numerous molecular electronic devices. A single electronic gate would be like a chemical macromolecule. These devices could be interfaces between silicon circuits and molecular intelligent databases similar to human brains. The lab-on-chip technology is also very promising in sequencing, and analizing data on molecular level.

However, before implementing solid-state circuits it is easier to test molecular system potentials during liquid computing driven by heating, cooling, and chemical reactions - so called molecular computing introduced by Adleman [2]. Some improvements to Adleman's initial experiment were made [13] by using double stranded DNA molecules rather than the single stranded molecules. Double stranded molecules are less error prone than the single stranded. Like Adleman other scientists [15, 41] were very likely to chose one of graph theory problems for molecular computation. Paper [30] reports solving NP-complete problem by building a pool of DNA molecules corresponding to the total ensemble of six-vertex cliques, followed by a series of selection processes. Amos [4] found a solution of three colouring of graphs. P. Wąsiewicz [46] also proposed evolutionary programming of logic function graphs, which evaluation is based on PCR.

Solution to another NP-Hard problem of logic function satisfiability was invented by R.J. Lipton [21]. In [7] Dan Boneh and Christopher Dunworth and Richard J. Lipton presented a method of breaking DES using a molecular computer and in [40] Stevens reported DNA model of computation of a Sub-Set Sum. In his experiment objects were encoded into synthetic DNA samples with restriction enzymes specially encoded to each object. By cutting the strands progressively with the assigned restriction enzymes, the objects within the containers can be found.

Further author information:
[a]Institute of Electronic Systems, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland; (Send correspondence to pwasiewi@elka.pw.edu.pl)
[b]Institute of Biotechnology and Antibiotics, Staroscinska 5, 02-516 Warsaw, apl@iba.waw.pl

The splicing system concept was created in 1987 to allow in formal language theoretic terms the convenient representation of recombinant actions of certain sets of enzymes on double stranded DNA molecules. Head [14] provided a potential biochemical example, where the formal generative capacity of the restriction enzyme BpmI in the company of a ligase is discussed. Other publications about splicing systems are written by Laun, Reddy [20] and by Siromoney, Subramanian and Dare [36], which reports splicing systems based on circular DNA strings. A new method for separation of DNA according to its substrings is developed and demonstrated in [8]. In [5] theoretical models of DNA computation have been proposed. In this paper preliminary investigations into the laboratory implementation was described. The heteroduplex formation during the PCR was described in [16].

Other researchers used DNA to solve different computational tasks e.g. [3, 6, 9–11, 17, 19, 24–29, 31–33, 35, 37, 38, 43]. Those described in the next point works were arbitrarily chosen from molecular computing bibliography [1] and present experiments, which results can influence on the next level biomachines.

2. MORE ADVANCED TECHNIQUES OF COMPUTING ON MOLECULES

Molecules could be used as memory. Kashiwamura, Yamamoto, Kameda, Shiba, and Ohuchi described the memory based on nasted PCR in [17]. The papers [28] shows a possibility of building associative memory based on DNA strands. The method for suppressing DNA fragment amplification during PCR was used. Experimental construction of a very large scale DNA database is presented in [32]. Mills, Yurke, Platzman in [26] described a particular set of DNA operations to effect the interconversion of electrical and DNA data and to represent the Hopfield associative memory. This new type of DNA computing has the possible advantage of being fault tolerant and thus more immune to DNA hybridization errors than a Boolean DNA computer.

Molecules can add integer numbers. Wąsiewicz, Rudnicki, Mulawka and Lesyng [43] presented new algorithm of DNA computing for adding binary integer numbers, which add numbers at the same quantity of elementary operations, regardless of a number of bits used for representation.

Molecules can exchange messages with each other. In paper [31] a new technique of sending data between molecular processors is presented. Its computation results have to be sent to other units in the form of addressed messages - tokens.

The molecule flow in mixtures can be automated within general-purpose DNA computers. McCaskill papers e.g. [27] described the microreactors implemented "in silico" with very small reaction nano chambers and input, output nanochannels. They could do make graph optimization on molecular level. Amenyo [3] showed that all proposed DNA computing algorithms can be run on parallel computer architectures configured from trellis/lattice banks, filter banks and switching banks. Thus, DNA computation can be re-interpreted as dataflow (or signal flow) networks and subject to conventional treatment. A micromachined chemical amplifier was successfully used to perform the polymerase chain reaction(PCR) in continuous flow at high speed [19].

Molecules can play and have intelligent behaviour. The first game-playing DNA computer was revealed a year ago. An enzyme-powered tic-tac-toe machine that cannot be beaten was developed by Milan Stojanovic, and Darko Stefanovic [37]. Faulhammer, Lipton, Landweber [10] developed RNA-based computing approach to a 9-bit instance of the "Knight Problem". Wood, Bi, Kimbrough, Wu and Chen made DNA learn Poker by using independent populations of strategies that adapt and learn from their experiences in competition. Deaton and Garzon implemented fuzzy logic with biomolecules [9]. Mihalache [24] made first steps towards molecular Prolog machine.

Molecules can be utilised in developing of a nanomachine molecular automaton and molecular state transitions. Design of nanomechanical devices based on DNA was proposed by Reif [38]. Simmel, and Yurke prepared experiments with DNA nanoactuators [35]. DNA nanomachine "scissors" were invented in Bell laboratories by Yurke and Mitchell [25]. Nowak, and Plucienniczak in [29] described the finite state automaton based on PCR. This method is used for analyze DNA molecules whether they are described by specified regular expression. Presented ideas are confirmed by experiment performed in genetic engineering laboratory. In [33] Sakamoto showed that combined with parallel overlap assembly, a single series of successive transitions can solve NP-complete problems. Benenson, Paz-Elizur, Adar, Keinan, Livneh, and Shapiro [6] reported development of a programmable 'finite automaton', using a restriction nucleaseand ligase as hardware and software consisting of transition rules encoded by DNA. Komiya, Sakamoto, Gouzo, Yokoyama, Arita, Nishikawa, and Hagiya [18]

showed that a single-stranded DNA can serve as an independent machine by using a solid support technique in three experimental achievements in computation model based on 'whiplash' reactions, while Garzon, Gao, Rose, Murphy, Deaton, Franceschetti, and Stevens [12] used a ligation-based approach for in-vitro implementation of finite-state machines, which requires sequential input feed and different molecules for different machines. In their second implementation not based on ligation transitions are represented by reusable molecules and the input, coded as a molecule, can be introduced at once.

At the end molecules can even conduct electrical current. Some researchers treat DNA computing as methodology similar to classical methods and study electrical conduction [11]. This studies showed that the resistivity values derived from these measurements are comparable to those of conducting polymers and indicate that DNA transports electrical current as efficiently as a good semiconductor [34].

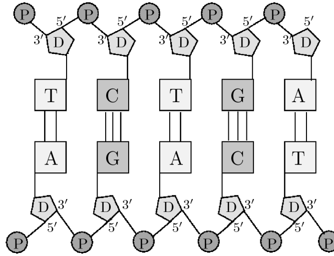

Figure 1. The chemical structure and bonds of DNA double molecule, a joint between single strings

3. DNA DESCRIPTION

A single-stranded DNA string has a phospho-sugar backbone with two different, 5' and 3' ends and four bases Adenine, Thymine, Cytosine, Guanine denoted by the symbols: A, T, C, and G, respectively. A double-stranded DNA string is in the form of oriented in opposite directions two single strings due to hybridization or in other words annealing reaction, because A is complementary with T, and C is complementary with G as is seen in Fig. 1. Operations on DNA oligos [22] may be described in the following way:

1. *Hybridization or renaturation* means connecting or annealing of single complementary DNA strings to single standard DNA strings and forming double stranded molecules. This operation is caused by cooling down the test tube reaction solution.

2. *Denaturation* means changing double stranded DNA molecules into single complementary and standard strings. Heating the test tube reaction solution causes it.

3. *Cutting* of a double DNA string into two parts is performed in DNA computing with the help of digesting restriction enzymes. The resulting double strings can have sticky ends (single stranded DNA) or blunt ends. Methylation of DNA usually has the effect of protecting the DNA from the related restriction endonuclease.

4. *Concatenation* of two strings is a string formed by placing the second string after the first string without any gap. In DNA computing joining of two strings is done during hybridization and ligation. They form together a longer single string. In order to concatenate two oligos a and b, the hybridized and complementary to them in the place of joint *third string* is needed.

5. *Amplification (PCR)*. In the cycle of amplification a single strings called primers are lengthened from its 3' end up to its complementary longer string 5' end. Thus, the primer has to be hybridized e.g. far away from longer string 5' end. Then a special enzyme called a polymerase lengthens the primer adding to the primer 3' end nucleotides complementary to the longer string which this primer is attached to. The amplification result is in the double stranded form and its length is equal to the previous longer string one.

A signal is represented by a DNA string called an oligo, an oligonucleotide, a strand, a DNA fragment, a DNA chain. A sequence of operations on DNA strings is called an algorithm. Together the genetic operations, driven by enzymes, heating and cooling, DNA sequence, and the model make computation possible.

Table 1. Properties of different polymerases

	E.coli DNA polymerase	E. coli DNA - Klenow fragment	T4 DNA poly-merase	T7 DNA polymerase	Taq DNA polymerase
$5' \to 3'$ exonuclease activity	exists				exists
$3' \to 5'$ exonuclease activity	exists	exists	exists	exists	
Error rate $\times 10^{-6}$	9	40	< 1	15	285
Strand replacement		exists			
Heat inactivation	exists	exists	exists	exists	

4. POLYMERASE PROPERTIES AND THEIR OPTIMIZATION PROCESS

Utilized in PCR process all DNA polymerases share two general characteristics [22]:

1. They add nucleotides to the $3' - OH$ end of a primer. The order of the nucleotides in the nascent polynucleotide is complementary to a oligo, which is connected with the primer.
2. In addition to the $5' \to 3'$ polymerase activity, polymerases can contain exonuclease activity. This exonuclease activity can proceed either in the $5' \to 3'$ direction, or in the $3' \to 5'$ direction.
   - Exonuclease activity in the $3' \to 5'$ direction allows the polymerase to correct a mistake if it incorporates an incorrect nucleotide (so called "error correction activity"). It can also slowly degrade the 3' end of the primer.
   - Exonuclease activity in the $5' \to 3'$ direction will allow to degrade any other hybridized primer it may encounter. Without $5' \to 3'$ exonuclease activity, obstructing primers may or may not be physically deplaced, depending on the polymerase being used.

Different polymerases have different characteristics, rates of polymerization as is seen in Table I. Thus, after adding all needed nucleotides in primer lengthening process polymerases usually begin to disassemble finished products, so it is very difficult with use of natural enzymes to construct nanodevices. Good synthesized enzymes are still not available and it is even impossible to synthesize new kinds of them. The only solution is to change properties of the existing ones.

In our experiment the Taq polymerase is modified in the following steps:

- DNA sequence modification. A DNA fragment responsible for $5' \to 3'$ exonuclease activity is cut off by PCR reaction from the polymerase DNA sequence in order to remove this polymerase ability, which destroys 5' single oligo ends.
- DNA structure evaluation. The amplified product is put in plasmid vector (circular DNA molecule) and cloned within bacteria colonies inside cells in order to reconstruct DNA structure, which is more suitable for living cell environment e.g. with all methylase DNA protection against restriction enzymes.

- Polymerase generation. After retrieving this new DNA from bacteria plasmid clones, it is placed in an expression vector. The expressed protein obtained from next bacteria colonies is just a modified Taq polymerase.

Every retrieval operation is executed with the help of amplification operation and electrophoresis, which can resolve DNA molecules in gel. In the electrical field smaller molecules pass through gel matrix faster and larger ones get caught up in this matrix of long gel chains. The group of strings with the same lengths can be cut out of gel.

Insertion of DNA fragments into vectors is performed with a help of cutting and concatenation operations. Vector is an agent that can carry a DNA fragment into a host cell. If it is used for reproducing the DNA fragment, it is called a cloning vector. If it is used for expressing certain gene in the DNA fragment, it is called an expression vector. One kind of vectors is called plasmid. Plasmids are circular, double-stranded DNA molecules that exist in bacteria. They can replicate independently of the host cell, which is destroyed at the end of the process. The size of plasmids ranges from a few kb (kb=1000 nucleotides) to near 100 kb. A plasmid vector is made from natural plasmids by removing unnecessary segments and adding essential sequences. To clone a DNA sample, the same restriction enzyme must be used to cut both the vector and the DNA sample. Therefore, a vector usually contains a sequence (polylinker) which can recognize several restriction enzymes so that the vector can be used for cloning a variety of DNA samples.

A plasmid vector must also contain a drug-resistance gene for selective amplification. After the vector enters into a host cell, it may proliferate with the host cell. However, since the transformation efficiency of plasmids in E. coli is very low, most E. coli cells that proliferate in the medium would not contain the plasmids. But it is a way to allow only the transformed E. coli to proliferate. Typically, antibiotics are used to kill E. coli cells which do not contain the vectors.

Many proteins which may be used for research are normally expressed at very low concentrations. Through recombinant DNA technology, a large quantity of proteins can be produced. This involves the cloning of the gene encoding the desired protein into an expression vector which must contain a promoter so that the protein can be expressed [22].

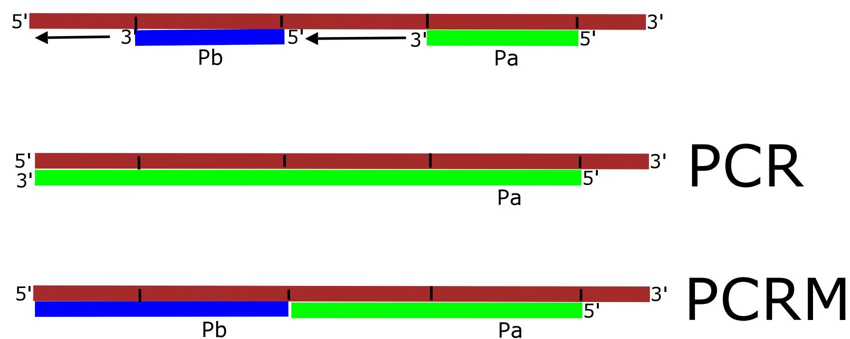

Figure 2. Amplifications: PCR with original polymerase, PCRM with optimized polymerase

5. EXPECTED RESULTS

During PCR with not modified polymerase, only one primer amplifies destroying others on its way to the 5' end of complementary string as is seen in Fig. 2. With optimized polymerase (PCRM) all primers lengthen themselves without removing others. After the process end all lengthened primers can be concatenated in the process of ligase.

In this way more sophisticated nanoconstructions can be self-assembled level by level with the help of modified enzymes. First new primers are attached to the previous level molecules. This process is repeated several times

and at the end all single string gaps are filled with a help of the new polymerase without destroying the whole construction.

6. SUMMARY

In this paper the process of changing characteristics of the enzyme during biochemical reactions was proposed. In near future it will be verified in genetic engineering laboratory.

Modified polymerase can be applied in nanoconstructions and nanocomputations. First, in one reaction chamber this highly thermostable polymerase can lengthen appropriate primers for many cycles and nanodevices can be created one layer over another one without unwanted polymerase side effects. Second, enzymes drive molecular computation processes and these processes can change their properties. Information data can circulate much longer. Of course, quicker process should be achieved in vivo or in synthesized living cells, but it is beyond present technological limits.

Molecular computing in vitro is a rapidly developing research area with unknown to an end opportunities and challenges. However, it is sure that despite its embrionic state this methodology enables on molecular level creation of new alternative computing architectures from which in future may emerge very powerful, massively parallel supercomputers or at least specialized lab-on-chip devices made for quick sequencing or cancer gene detection.

From fundamental self-assembly of basic elements, new logic systems could emerge through implementing information techniques among others inference systems, neural structures, binary operations on nano scale. A gap between implementation and theory of computer science should be omitted in the mentioned way leading to new architectures of based on polimers or peptides specialized processors computing huge data amounts.

With latest achievements it would be possible to invent new paradigms, more effective algorithms and with a help of all previous methods to make autonomous biomachines, which can contain even living cells as processing units with implemented selfmodification.

In further research original nanoconstruction systems with the help of own modified enzymes with optimized properties should be proposed.

References

[1] *The Bibliography of Molecular Computation and Splicing Systems*, at the site: http://www.dcs.ex.ac.uk/˜ pf201/dna.html.

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

[3] J. Amenyo. Mesoscopic computer engineering: Automating DNA-based molecular computing via traditional practices of parallel computer architecture design. In *Proceedings of the Second Annual Meeting on DNA Based Computers, held at Princeton University*, 1996.

[4] M. Amos. DNA computation. PhD Thesis. University of Warwick, 1997.

[5] M. Amos, S. Wilson, D. A. Hodgson, G. Owenson, and A. Gibbons. Practical implementation of DNA computations. In *Unconventional Models of Computation*, 1998:1–18.

[6] Y. Benenson, T. Paz-Elizur, R. Adar, E. Keinan, Z. Livneh, and E. Shapiro. Programmable and autonomous computing machine made of biomolecules. *Nature*, 414, 2001.

[7] D. Boneh, C. Dunworth, and R. J. Lipton. Breaking DES using a molecular computer. In *DNA Based Computers*, 1996.

[8] J. Chen and D. H. Wood. A new DNA separation technique with low error rate. In *Proceedings of the 3rd DIMACS Workshop on DNA Based Computers, held at the University of Pennsylvania*, 1997.

[9] R. Deaton, M. Garzon Fuzzy logic with biomolecules. Soft Computing 5, 2001:2-9.

[10] D. Faulhammer, A.R. Cukras, R.J. Lipton, L.F. Landweber. When the Knight Falls: On Constructing an RNA Computer. In *Proceedings of the 5rd DIMACS Workshop on DNA Based Computers, held at MIT*, 1999.

[11] H. Fink and C. Schoenenberg. Electrical conduction through DNA molecules. *Nature*, 398, 1999.

[12] M. Garzon, Y. Gao, J.A. Rose, R.C. Murphy, R. Deaton, D.R. Franceschetti, and S.E. Stevens, Jr. In-vitro implementation of finite-state machines. In *Proc. 2nd Int. Workshop on Implementing Automata*, 1998.

[13] T. Head. Hamiltonian paths and double stranded DNA. In *Computing with Bio-Molecules. Theory and Experiments*, 1998.

[14] T. Head. Splicing languages generated with one sided context. In *Computing with Bio-Molecules. Theory and Experiments*, 1998.

[15] N. Jonoska and S. A. Karl. A molecular computation of the road coloring problem. In *Proceedings of the Second Annual Meeting on DNA Based Computers, held at Princeton University*, 1996.

[16] P. D. Kaplan, G. Cecchi, and A. Libchaber. DNA based molecular computation: template-template interactions in PCR. In *Proceedings of the Second Annual Meeting on DNA Based Computers, held at Princeton University*, 1996.

[17] S. Kashiwamura, M. Yamamoto, A. Kameda, T. Shiba, and A. Ohuchi. Hierarchical DNA memory based on nested PCR. Poster paper at 8th International Workshop on DNA-Based Computers, DNA 2002, Sapporo, Japan, 2002.

[18] K. Komiya, K. Sakamoto, H. Gouzo, S. Yokoyama, M. Arita, A. Nishikawa, and M. Hagiya. Successive state transitions with i/o interface by molecules. In *DNA Computing, 6th international Workshop on DNA-Based Computers, Leiden, The Netherlands*, 2000.

[19] M. U. Kopp, A. J. de Mello, and A. Manz. Chemical amplification: Continuous-flow PCR on a chip. *Science*, 280, 1998.

[20] E. Laun and K. J. Reddy. Wet splicing systems. In *Proceedings of the 3rd DIMACS Workshop on DNA Based Computers, held at the University of Pennsylvania*, 1997.

[21] R.J. Lipton. Solution of Hard Computational Problems. Science 268, 1995:542–545.

[22] T.Maniatis, E.F.Fritsch, and J.Sambrook. Molecular cloning: a Lab. Manual. Cold Spring Harbor Lab. Press, Cold Spring Harbor, NY, 1982.

[23] D. Matsuda, M. Yamamura. Cascading Whiplash PCR with a Nicking Enzyme. LNCS 2568, 38-46.

[24] V. Mihalache. Prolog approach to DNA computing. Proc. of IEEE Int. Conf. on Evolutionary Comp. ICEC97, 1997.

[25] J.C. Michell, B. Yurke. DNA Scissors LNCS 2340, 258-268.

[26] A. P. Mills Jr. and B. Yurke and P. M. Platzman. DNA Analog Vector Algebra and Physical Constraints on Large-Scale DNA-Based Neural Network Computation. In *Proceedings 5th DIMACS Workshop on DNA Based Computers, held at the Massachusetts Institute of Technology*, 1999.

[27] D. van Noort, F. Gast, J. S. McCaskill. DNA Computing in Microreactors. LNCS 2340, 33-45.

[28] R. Nowak, A. Plucienniczak, J. Mulawka. Molecular associative memory built on DNA. *http://staff.elka.pw.edu.pl/˜rnowak2/memory.ps.*

[29] R. Nowak, A. Plucienniczak. Non-deterministic finite state automata built on DNA. *http://staff.elka.pw.edu.pl/˜rnowak2/automaton.ps.*

[30] Q. Ouyang, P. D. Kaplan, S. Liu, and A. Libchaber. DNA solution of the maximal clique problem. *Science*, 278, 1997.

[31] J.J. Mulawka R. Nowak, P. Wąsiewicz and A. Plucienniczak. Processing DNA tokens in parallel computing. In *Proc. International Parallel and Distributed Processing Symposium*, 2001.

[32] J. H. Reif, T. H. LaBean, M. Pirrug, V. S. Rana, B. Guo, C. Kingsford, and G. S. Wickham. Experimental construction of a very large scale DNA database with associatice search capability. In *DNA Computing, 7th international Workshop on DNA-Based Computers, Tampa, U.S.A.*, 2001.

[33] K. Sakamoto, D. Kiga, K. Komiya, H. Gouzu, S. Yokoyama, S. Ikeda, H. Sugiyama, and M. Hagiya. State transitions by molecules. In *Proceedings 4th DIMACS Workshop on DNA Based Computers, held at the University of Pennsylvania*, 1998.

[34] D. Sen and R. P. Fahkman. DNA conformational switches as sensitive electronic sensors of analytes and as components for logic gates. In *DNA Computing, 8th international Workshop on DNA-Based Computers, Sapporo, Japan*, 2002.

[35] F.C. Simmel and B. Yurke. Operation of a Purified DNA Nanoactuator. LNCS 2340, 248-257.

[36] R. Siromoney, K.G. Subramanian, and V. Rajkumar Dare. Circular DNA and splicing systems. In A. Nakamura, M. Nivat, A. Saoudi, P.S.P. Wang, and K. Inoue, editors, *Proceedings of Parallel Image Analysis, 2nd International Conference ICPIA '92, Ube, Japan*, 2002.

[37] M. N. Stojanovic and D. Stefanovic. A Deoxyribozyme-Based Molecular Automaton. Nature Biotechnology 21, 2003:1069-1074.

[38] J. H. Reif. The Design of Autonomous DNA Nanomechanical Devices: Walking and Rolling DNA. LNCS 2568, 22-37.

[39] P. Sa-Ardyen, N. Jonoska, N.S. Seeman. Self-assembling DNA Graphs LNCS 2568, 1-9.

[40] J. Stevens. DNA model of computation of a sub-set sum. *http://www.lcc.net/ dmjs/Dna.htm.*

[41] G. Tomczuk, P. Wąsiewicz. Molecular binary trees. In *Proc. of the VI Polish Conference on Evolutionary Algorithms and Global Optimization, Korbielow*, 249-254.

[42] P. Wąsiewicz, G. Tomczuk, A. Plucienniczak. Molecular Neuron Network Experimental Approximation. In *Proc. 7th WSEAS Int. Conf. Automatic Control, Modelling and Simul., Prague, Czech*, 2005, 489-493.

[43] P. Wąsiewicz, R. Rudnicki, J.J. Mulawka, and B. Lesyng. Adding numbers with DNA. In *Proceedings 2000 IEEE International Conference on Systems*, 2000.

[44] P. Wąsiewicz, A. Malinowski, R. Nowak, J.J Mulawka, P. Borsuk, P. Weglenski, and A. Plucienniczak. DNA computing: Implementation of data flow logical operations. *Future Generation Computer Systems*, 17/4, 2001, 361-378.

[45] P. Wąsiewicz, A. Dydynski, G. Tomczuk, J.J. Mulawka, A. Plucienniczak. Molecular Neuron Realization. WSEAS Transactions Journal on Biology and Biomedicine, 1(1):73-75.

[46] P. Wąsiewicz, J.J. Mulawka. Molecular Genetic Programming. Soft Computing, Springer-Verlag 5(2), 2001:106-113.

[47] D. H. Wood, H. Bi, S. O. Kimbrough, D. Wu, and J. Chen. DNA starts to learn poker. In *DNA Computing, 7th international Workshop on DNA-Based Computers, Tampa, U.S.A.*, 2001.