

Israel M. Martínez-Pérez, Zhang Gong, Zoya  
Ignatova, and Karl-Heinz Zimmermann

# Solving the Maximum Clique Problem via DNA Hairpin Formation



**Abstract.** The maximal clique problem is one of the famous hard combinatorial problems. We provide the first molecular-scale autonomous solution of the decision maximal clique problem. It is based on the formation of secondary structures of DNA molecules. Finally, we conject that every decision NP complete problem can be solved by DNA computing in linear time.

## 1 Introduction

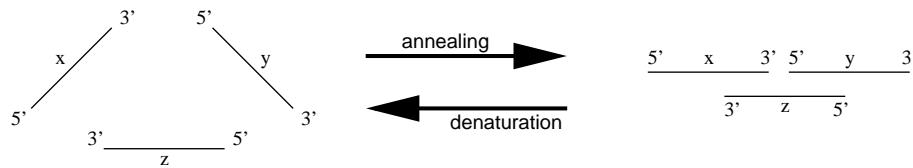
Feynman [1] was the first giving a visionary talk about the prospect of performing massively parallel computations in nanotechnology. His idea was first brought to laboratory by Adleman [2] solving a small instance of the Hamiltonian path problem by a laboratory-scale, human-operated DNA computation. The idea of computing at the molecular scale encouraged researchers to work on other biomolecular computing models. Most of them aimed to tackle complex computational problems [3] such as the satisfiability (SAT) problem [4] and graph problems [5], [6], [7], [8], and [9]. Moreover, DNA computing was shown to be computationally complete and universal [10], [11]. Algorithms at molecular scale exploit the high degree of parallelization and storage capacity inherent to DNA molecules. The objective was to outperform electronic computers when it comes to the computation of large instances of NP hard computational problems. But this vision was rapidly discarded when researchers realized some of the drawbacks of this incipient technology: growing number of error-prone laboratory operations and exponential growth of DNA volume according to problem size [12]. The ongoing research focussed to solve problems at molecular level which can hardly be tackled by electronic machines. Here, autonomous computations play a critical role.

Recently, biomolecular computer research focused on molecular-scale autonomous programmable models of computation. These models are based on the self-assembly of smaller DNA molecules modulated by DNA-manipulating enzymes, minimizing the external control or interference. Nano-structures in the form of large-scale periodic two-dimensional lattices were self-assembled from a diverse set of small branched DNA molecules [13]. Such a self-assembled lattice was used to design an autonomous universal Turing machine [14]. Several autonomous DNA devices were constructed that are able to move or walk [15], [16]. Moreover, autonomous two-input two-state finite automata were constructed [17], and later, applied to solve *in vitro* the logical control of gene expression [18]. Autonomous DNA devices are able also to tackle complex combinatorial problems as demonstrated by Sakamoto *et al* [19] proposing a solution of the SAT problem using hairpin formation of DNA molecules. A variant of this model was used by Martínez-Pérez *et al* [20] to solve the Hamiltonian path problem by using the hairpin formation of palindromic sequences - an unwanted sequence in most DNA computing models.

In this paper, we present a solution of the maximal clique problem by an autonomous DNA computation based on the principle of hairpin formation. The laboratory steps partly resemble the solution of the Hamiltonian path problem. However, the encoding of the information (graph) is much different. Before this, we provide the necessary background on DNA and DNA operations and review existing DNA algorithms for the maximal clique problem. Finally, we conject that every decision NP complete problem can be solved in linear time by DNA computing.

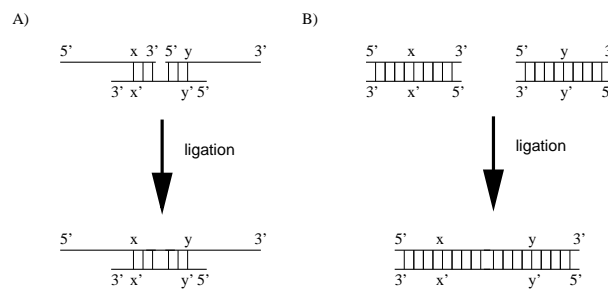
## 2 Background

The four bases are adenine (A), guanine (G), cytosine (C), and thymine (T) are covalently bonded end-to-end in 5' to 3' direction to form single stranded (ss) DNA molecules [21]. A ssDNA molecule has a polarity distinguishing it from its reverse strand. The unique complementation based on hydrogen bonds between (A) and (T), and (G) and (C) determine the rigid structure of double stranded DNA. At high temperatures both strands can be separated (*denaturation*) and re-anneal by cooling down (Fig. 1).



**Fig. 1.** Annealing and denaturation. At high temperature, the hydrogen bonds between complementary nucleotides break without changing the covalent bonds linking adjacent nucleotides.

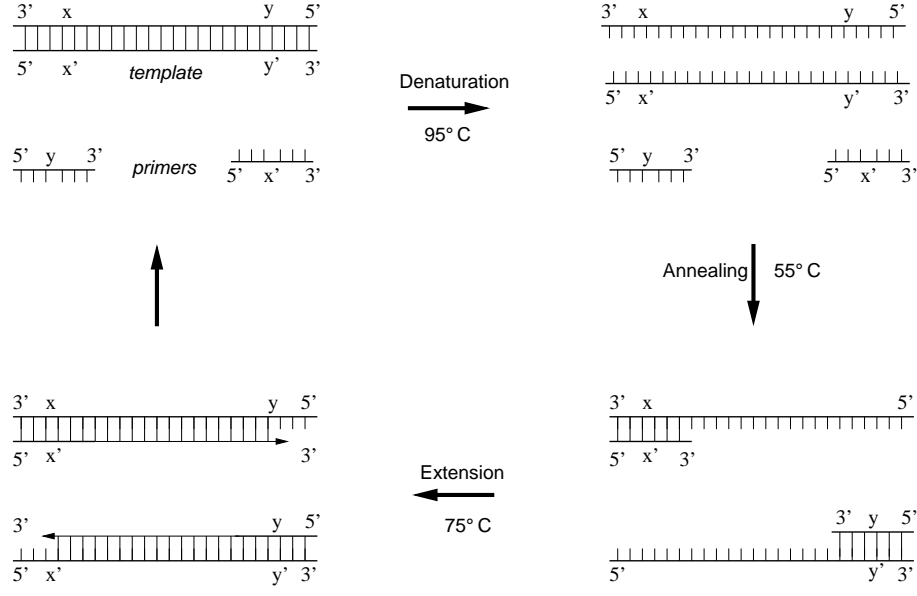
In DNA computing, DNA is utilized as a substrate for storing information. Depending on the model of DNA computation, information is stored in the form of ssDNA and/or dsDNA molecules. This stored information could be manipulated by enzymes. One class of enzymes, *restriction endonucleases*, recognize a specific short sequence of DNA, called *restriction site*, and cut the covalent bonds between the adjacent nucleotides (Fig. 8). *DNA ligase*, links the ends of ssDNA strands repairing backbone breaks (Fig. 2). The *exonucleases* are enzymes that hydrolyze phosphodiester bonds from either the 3' or 5' terminus of ssDNA or dsDNA molecules.



**Fig. 2.** Ligation. Ligase connects blunt or sticky ends of ssDNA either in a single strand (A) or within a double (B).

*DNA polymerase* performs several functions including replication of DNA. The replication reaction requires an annealing of ssDNA molecule, called *template*, with a short ssDNA molecule, called *primer*. DNA polymerase catalyzes DNA synthesis by successively adding nucleotides to one end of the primer. In this way,

the primer is extended in one direction until the desired strand that starts with the primer and is complementary to the template is provided. The template DNA can be amplified in a *polymerase chain reaction* (PCR) (Fig. 3). PCR is an iterative process, with each iteration consisting of the following steps: Annealing of the primers to the templates, extending of the primers by DNA polymerase, denaturing of the newly elongated dsDNA molecules to separate its strands, and cooling to allow re-annealing. Each iteration doubles the number of target DNA molecules.



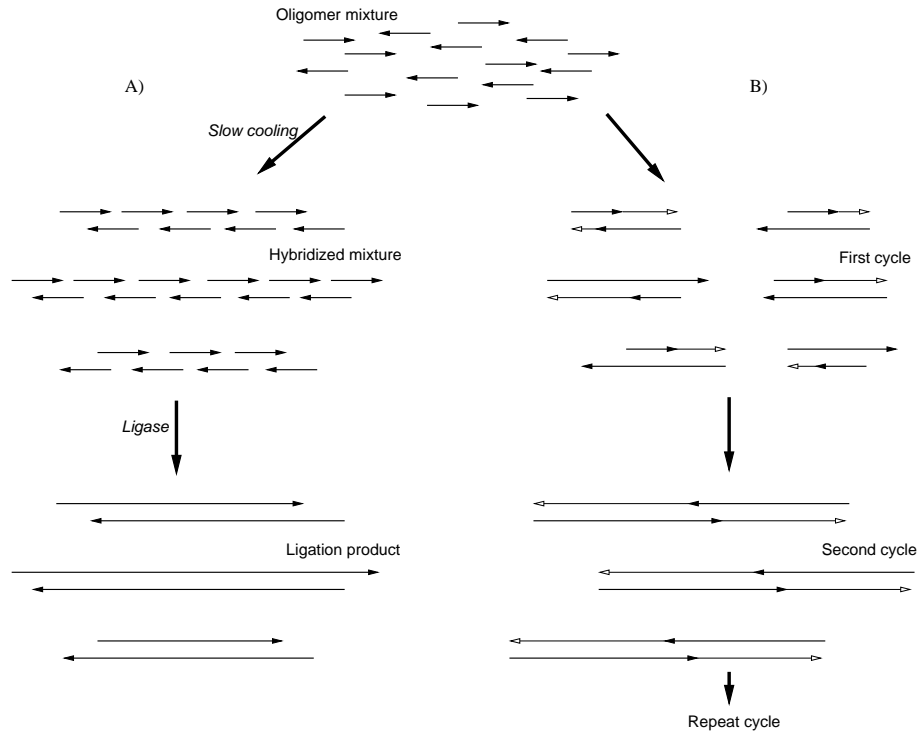
**Fig. 3.** One cycle of PCR.

*Parallel overlap assembly* (POA) is a method to generate a pool of DNA molecules (combinatorial library) [22]. Short ssDNA molecules overlap after annealing and their sticky ends are extended by DNA polymerase. Repeatedly denaturation, annealing, and extension increase the length of the strands. Unlike the PCR, where the target DNA strands doubles in every cycle, in POA, the number of DNA strands does not change, only the length increases with cycle progression (Fig. 4).

Short ssDNA molecules (oligomers) can be designed by using available software, e.g., DNASequencer [23]. The CG contents can be specified as input affecting the melting temperature of the sequences. Oligomers can be synthesized *in vitro* using PCR. Each DNA algorithm requires to encode the basic data, e.g., vertices and edges of a graph, by oligomers. We assume that this *preparation step* requires linear time.

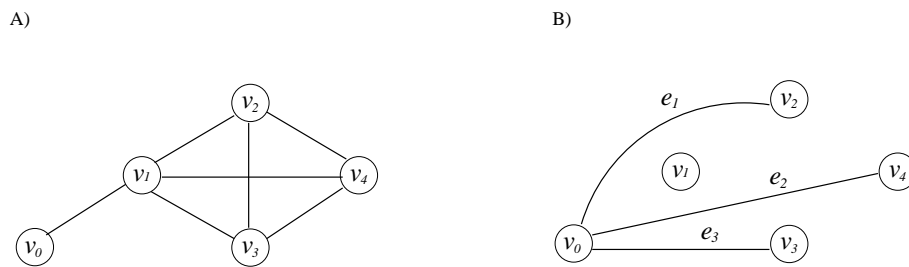
### 3 DNA Solutions of Maximal Clique Problem

A *complete graph* is a graph in which each pair of vertices is connected by an edge. A *clique* of a graph  $G$  is a complete subgraph of  $G$ . A *maximal clique* of  $G$  is a clique with maximal number of vertices. For instance, the graph in Fig. 5.A has the maximal clique given by the vertex set  $\{v_2, v_3, v_4, v_5\}$ . The problem of finding



**Fig. 4.** Synthesis methods for combinatorial libraries: A) Annealing/ligation: The arrow heads indicate the 3' end. B) POA: The thick arrow represents the synthesized oligomers which are the input of the computation. The thin arrows represent the elongated part by the polymerase. The arrow head indicates the 3' end.

a maximal clique of a graph is NP hard. The *complement* of a graph  $G$  is a graph  $G^c$  with the same vertex set but whose edge set consists of the edges not present in  $G$ . An *independent vertex set* of  $G$  is a subset of the vertices such that no two vertices in the subset represent an edge of  $G$ . A *maximal independent vertex set* of  $G$  is an independent vertex set of  $G$  with maximal cardinality. Notice that the cliques of  $G$  correspond one-to-one with the independent vertex sets of  $G^c$ . In particular, the maximal cliques of  $G$  correspond to maximal independent vertex sets of  $G^c$ .



**Fig. 5.** A) Graph  $G$  with five vertices. B) Complementary graph  $G^c$ .

Ouyang *et al* [24] proposed a quadratic time DNA algorithm to solve the maximal clique problem. To specify the algorithm, let  $G$  be a graph with vertex set  $V = \{v_1, \dots, v_n\}$ . Each subset of  $V$  is represented by a binary number of length  $n$ , i.e., a subset contains  $v_i$  iff the  $i$ th bit is set. These binary numbers are represented by ssDNA molecules of the form  $P_1V_1P_2V_2P_3 \dots P_nV_nP_{n+1}$ , where  $P_i$  has length 20 bp and  $V_i$  has length 0 bp (bit 1) or 10 bp (bit 0). These molecules are synthesized via POA from short ssDNA molecules:  $P_iV_iP_{i+1}$  for  $i$  odd and the complement of  $P_iV_iP_{i+1}$  for  $i$  even. Moreover, each ssDNA molecule with  $V_i$  of length 0 bp contains a restriction site between  $P_i$  and  $P_{i+1}$ . These sites are different for different  $i$  and require a distinguished restriction enzyme. The algorithm works as follows:

1. Generate random subsets of  $V$  (combinatorial library).
2. Identify cliques in the combinatorial library.
3. Find the maximal cliques.
4. Read out maximal cliques.

The second step is implemented by the restriction enzymes. If the vertices  $v_i$  and  $v_j$  are adjacent in  $G^c$ , the combinatorial library is divided into two test tubes  $T_1$  and  $T_2$ . The tube  $T_1$  and  $T_2$  are digested by the enzyme corresponding to  $v_i$  and  $v_j$ , respectively. The resulting solutions are mixed into a new tube  $T$ . This operation is sequentially repeated for each edge of  $G^c$ . The nondigested dsDNA molecules correspond to cliques of  $G$ . These molecules are amplified by PCR (with primers  $P_1$  and the complement of  $P_{n+1}$ ), while the broken molecules are not amplified. In the third step, the dsDNA molecules with shortest length are detected by gel electrophoresis, as these molecules correspond to maximal cliques of  $G$  thanks to the encoding of the subsets of  $V$ . Notice that the DNA algorithm solves the decision maximal clique problem. i.e., preparation step and algorithm steps 1-3, in quadratic time  $O(n^2)$ , as the preparation step is assumed to be linear and the steps 1-3 are linear in the number of edges of  $G^c$ .

## 4 DNA Hairpin Model

Originally proposed by Sakamoto *et al* [19], the DNA Hairpin model was used to solve a particular case of the SAT problem. The goal of the SAT problem is to decide whether a given Boolean formula in conjunctive normal form (CNF) has an assignment that makes the formula "true". A CNF consists of a conjunction (and) of one or more conjuncts, each of which is a disjunction (or) of one or more literals (variables or complements of variables). A literal string is defined as a conjunction of literals such that one literal per conjunct is selected. A Boolean formula is satisfiable if there is a literal string that does not involve a variable and its complement. In Sakamoto's work, each literal encodes a DNA strand, whereas the negation complements in Watson-Crick mode. As a consequence, literal strings with at least one pair of complementary literals may form a hairpin structure. Hairpin structures can be distinguished from non-hairpin structures by biochemical means. Sakamoto *et al.* suggested an alternative in which each literal contains the Bst NI recognition site so that the joint region of the hairpin structure is susceptible by digestion. Consequently, those literals strings that remain complete after enzyme digestion satisfy the Boolean formula.

Recently, the DNA hairpin model was used to solve the Hamiltonian path problem [20]. For this, the graph  $G$  in question is encoded as in Adleman's first experiment [2], i.e., each vertex  $v$  is encoded by a ssDNA

molecule and each edge connecting vertex  $v_i$  to vertex  $v_j$  is encoded by a ssDNA molecule that consists of the complement of the second 3' half-mer of  $v_i$  and the complement of the first 3' half-mer of  $v_j$ . A starting vertex in the graph is distinguished and the ssDNA molecules encoding the starting vertex are dephosphorylated. Firstly, random paths in the graph are generated (combinatorial library) by annealing and ligation. Secondly, those paths are kept that begin with the starting vertex. This is achieved by lambda exonuclease which selectively digests all strands with 5' phosphate group, i.e., strands of edges and strands of vertices with invalid starting vertex. Thirdly, those paths are kept that contain each vertex at most once. To this end, paths with at least one repeated vertex form a hairpin structure thanks to the palindromic encoding of the vertices. Those ssDNA molecules contain the Hae III recognition site and are digested by Hae III (Fig. 8). Fourthly, those paths are kept that have  $n$  vertices, where  $n$  is the number of vertices of  $G$ . This is achieved by gel electrophoresis. The corresponding fragment provides Hamiltonian paths (if any).

Both DNA algorithms work in an autonomous manner, i.e., constant time. If we assume that the preparation step requires linear time, these algorithms solve decision SAT problem and the decision Hamiltonian path problem in linear time. A linear time DNA algorithm for the maximal clique problem is provided next.

## 5 DNA Hairpin Model for Maximum Clique Problem

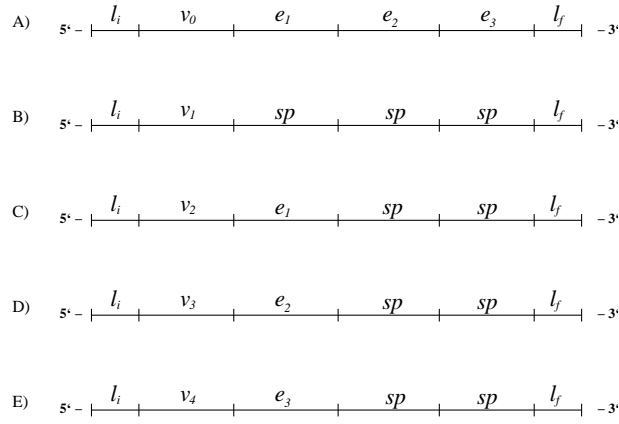
The maximal clique problem will be solved by using the DNA hairpin model. Let  $G$  be a graph. A data structure called *vertex template* is used which basically contains information about the edges incident with vertices in  $G^c$ . The template for the vertex  $v_j$  is encoded by a ssDNA molecule consisting of initial linker  $l_i$ , encoding of  $v_j$ , encodings of all edges of  $G^c$ , and final linker  $l_f$  (Fig. 6). The linkers are used to construct a combinatorial library. For this, there is a ssDNA molecule called *bridge* that consists of the complement of the second 3' half-mer of  $l_i$  and the complement of the first 5' half-mer of  $l_f$  (Fig. 7.a). Moreover, all vertices and edges of  $G^c$  are encoded in palindromic form containing in the centre the Hae III recognition site GGCC (Table 1). Furthermore, all vertex templates have the same length although the vertices of  $G^c$  may have varying degrees, i.e., number of incident edges. For this, there is a constant non-palindromic encoding sequence called *spacer* (sp). Finally, for each vertex template, part (say half) of its concentration is treated with calf intestinal alkaline phosphatase (CIAP) which dephosphorylates these strands (i.e., removes phosphate group at 5' terminus). A dephosphorylated vertex template cannot be a substrate of lambda exonuclease, which only recognizes phosphorylated 5' termini.

The maximal clique problem for a graph  $G$  with vertex set  $V$  can be solved by the following DNA hairpin algorithm:

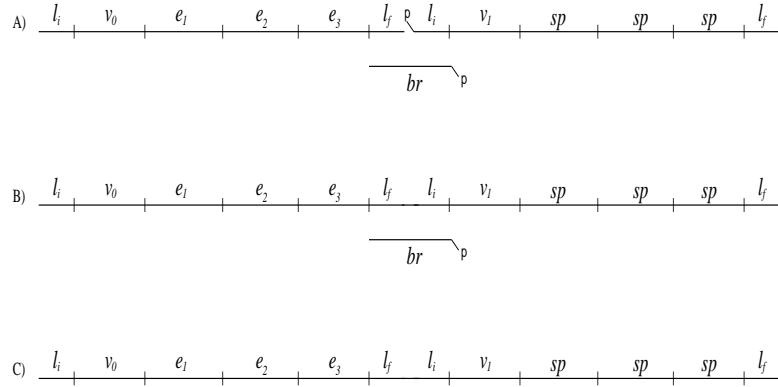
1. Generate random multi-subsets of  $V$  (combinatorial library).
2. Find cliques of  $G$ .
3. Detect maximal cliques.
4. Readout maximal cliques.

A combinatorial path library is created by annealing and linking vertex templates and bridges (Fig. 7.B). These partially dsDNA molecules correspond to *multi-subsets* of  $V$ , i.e., subsets with repeated vertices. Notice that a dephosphorylated vertex template can only occur at the beginning of such a partially dsDNA





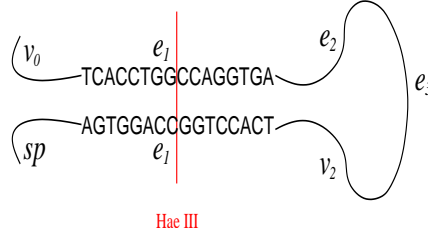
**Fig. 6.** Templates for the vertices of the graph  $G$ : A)  $v_0$  template, B)  $v_1$  template, C)  $v_2$  template, D)  $v_3$  template, and E)  $v_4$  template.



**Fig. 7.** Generation of combinatorial library: A) Annealing of two vertex templates and a bridge by the linkers. The 5'-phosphate group of the first vertex template was removed. B) Ligation bonds together vertex templates. C) Lambda exonuclease selectively digests all phosphorylated DNA strands leaving only those ssDNA molecules which have no 5'-phosphate group.

molecule. Thereafter, lambda exonuclease is used to selectively digest the phosphorylated strand of dsDNA molecules (Fig. 7.C). In this stage of the experiment, the remaining non-digested molecules are ssDNA which correspond to multi-subsets of  $V$  beginning with a non-phosphorylated vertex template. Due to the palindromic nature of the encoding vertices and edges, after lambda exonuclease treatment, those ssDNA corresponding to multi-subsets of  $V$  with two or more repeated vertices or edges form a hairpin structure. The partially double stranded region of such a hairpin structure contains the Hae III recognition site. All ssDNA with a hairpin structure will be digested by Hae III (Fig. 8). Therefore, the non-digested ssDNA will correspond to subsets of  $V$  which contain each vertex or each edge of  $G^c$  only once. These subsets correspond to cliques of  $G$ . Finally, the non-digested ssDNA of longest length correspond to maximal cliques of  $G$  and are detected by gel electrophoresis.

This procedure was successfully implemented in the laboratory for the graph shown in Fig. 5 [25]. The sequence of the vertex templates is given in Table 1. For instance, for dephosphorylated vertex template  $v_0$ , the largest visible band is about 160 bp corresponding to the clique  $\{v_0, v_1\}$  (Fig. 9, lane 3). For de-



**Fig. 8.** DNA hairpin formation cleavage: The DNA strand contains the vertex templates for  $v_0$  and  $v_2$ . Both vertices are incident with edge  $e_1$  and the strand forms a hairpin structure by intramolecular annealing to the palindromic  $e_1$  sequence. The hairpin is cleaved by the Hae III restriction enzyme.

phosphorylated vertex template  $v_2$ , the largest visible band is 320 bp corresponding to the maximal clique  $\{v_1, v_2, v_3, v_4\}$  (Fig. 9, lane 5).

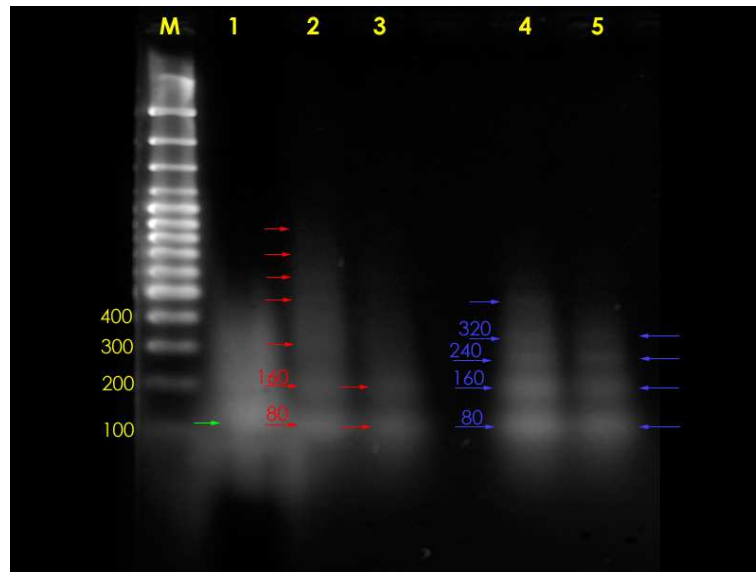
**Table 1.** Sequence of vertex templates.

Region	ssDNA molecule
$v_0$	5' - ACTGACGGCCGTCAGT - 3'
$v_1$	5' - TACGATGGCCATCGTA - 3'
$v_2$	5' - GTGAGAGGCCTCTCAC - 3'
$v_3$	5' - CGTTCAGGCCTGAACG - 3'
$v_4$	5' - AGCTTCGGCCGAAGCT - 3'
$e_1$	5' - TCACCTGGCCAGGTGA - 3'
$e_2$	5' - GATCTGGGCCAGATC - 3'
$e_3$	5' - CTGTAAGGCCTTACAG - 3'
$l_i$	5' - CGCAATTC - 3'
$l_f$	5' - TCTACGCT - 3'
$br$	5' - GAATTGCGAGCGTAGA - 3'
$sp$	5' - TAAATAAATAAATAAA - 3'

## 6 Discussion

The above DNA hairpin algorithm solves the decision maximal clique problem of a graph  $G$  with  $n$  vertices in linear time  $O(n)$ . To prove this, the vertex templates can be treated with CIAP in linear time and we assume that the remaining preparation step requires linear time. Moreover, the algorithm steps 1-3 work in an autonomous manner in constant time. This method could be scaled up and automated to efficiently solve a case of the maximal clique problem for graphs with up to 20 vertices. This estimate is in accordance with the largest instance (20 variables) of the SAT problem solved by DNA computing [4].

Our approach increases the spectrum of application of the DNA hairpin model, not only to solve the decision version of the SAT problem, but also the Hamiltonian path problem, and the maximal clique problem in linear time. Therefore, we postulate the following hypothesis:



**Fig. 9.** Experimental solution of maximal clique problem for the graph in Fig. 5. The lanes are marked as follows in the DNA-agarose gel: Lane M: DNA ladder. Lane 1: vertex template  $v_3$  as reference (80 bp) marked with green arrow. Lane 2: partially ds DNA molecules after annealing and linking, starting with dephosphorylated vertex template  $v_0$ . Lane 3: ssDNA molecules starting with vertex template  $v_0$ , after Hae III digestion. Lane 4: partially ds DNA molecules after annealing and linking, starting with dephosphorylated vertex template  $v_2$ . Lane 5: ssDNA molecules starting with vertex template  $v_2$ , after Hae III digestion. The size of the bands is highlighted by an arrow.

*Given a decision NP complete problem  $P$ . There is an encoding of the basic data of  $P$  (preparation step) so that each instance of  $P$  can be solved by an autonomous DNA computation in constant time. The preparation step takes linear time and so the overall computation time is linear, too.*

The polynomial time reduction of a decision NP complete problem into another is not useful. So in order to validate the conjecture, the basic data of each decision NP complete problem need to be individually encoded as is the case for the three prototype problems.

The palindromic sequences encoding the vertices and edges trigger two additional reactions on ssDNA molecules: inter- and intrastrand annealing. Clearly, the goal is to maximize intrastrand annealing (hairpin formation) and minimize interstrand annealing. As a biomolecular reaction, the intramolecular annealing can be reduced by decreasing the concentration of ssDNA molecules [25].

Finally, the techniques applied in this work to tackle the maximal clique problem (i.e., destroying molecules with repeated information by using intramolecular hybridization followed by digestion, and making use of palindromic sequences and hairpin formation) may open new avenues in other promising fields of DNA computing such as controlling living cells, building patterns, and controlling nano-machines.

## References

1. R.P. Feynman, Miniaturization, in: D.H. Gilbert (Ed.), Rheinhold, New York (1961) 282-296.
2. L. Adleman, Molecular computation of solutions to combinatorial problems, Science 266 (1994) 1021-1024.

3. C. Papadimitriou, K. Steiglitz, *Combinatorial Optimization*, Prentice Hall, Englewood Cliffs, NJ, 1982.
4. B. Ravinderjit, N. Chelyapov, C. Johnson, P. Rothmund, L. Adleman, Solution of a 20 variable 3-SAT problem on a molecular computer, *Science* 296 (2002) 499-502.
5. E. Bach, A. Condon, E. Glaser, C. Tanguay, DNA models and algorithms for NP complete problems, *Proc. 11th Ann. IEEE Conf. Comp. Complexity*, IEEE Computer Society Press (1996) 290-299.
6. T. Eng, D. Faulhammer, A surface-based DNA algorithm for minimal set cover, *Proc. 3rd DIMACS Meeting on DNA Based Computers* (1997) 74-82.
7. N. Jonoska, S. Karl, A molecular computation of the road color problem, *Proc. 2nd Annual Meeting on DNA Based Computing* (1996) 148-158.
8. R.J. Lipton, DNA solution of hard combinatorial problems, *Science* 268 (1995) 542-548.
9. K.-H. Zimmermann, Efficient DNA sticker algorithms for graph theoretic problems, *Computer Physics Communications* 144 (2002) 297-309.
10. L. Kari, G. Paun, G. Rozenberg, A. Salomaa, S. Yu, DNA computing, sticker systems, and universality, *Acta Informatica* 35 (1998) 401-420.
11. L. Kari, DNA computing: arrival of biological mathematics, *The Mathematical Intelligencer* 19 (1997) 9-22.
12. J. Hartmanis, On the weight of computations, *Bull. Europ. Assoc. Theoret. Comput. Sci.* 55 (1995) 136-138.
13. E. Winfree, *Algorithmic self-assembly of DNA*, PhD Thesis, CalTech, 1998.
14. J.H. Reif, The design of autonomous DNA nanomechanical devices, *LNCS* 2568 (2003) 22-37.
15. Y. Chen, M. Wang, C. Mao, An autonomous DNA motor powered by a DNA enzyme, *Angew. Int. Ed.* 43 (2004) 2-5.
16. P. Yin, A. Turberfield, S. Sahu, J.H. Reif, Design of an autonomous DNA nanomechanical device capable of universal computation and universal translational motion, *Proc. 10th Int. Meeting on DNA Computing* (2004) 426-444.
17. Y. Benenson, T. Paz-Elizur, R. Adar, E. Keinan, Z. Livneh, E. Shapiro, Programmable and autonomous computing machine made of biomolecules, *Nature* 414 (2001) 430-434.
18. Y. Benenson, B. Gil, U. Ben-Dor, R. Adar, E. Shapiro, An autonomous molecular computer for logical control of gene expression, *Nature* 429 (2004) 423-429.
19. K. Sakamoto, H. Gouzu, K. Komiyama, D. Kiga, S. Yokoyama, T. Yokomori, M. Hagiya, Molecular computation by DNA hairpin formation, *Science* 288 (2000) 1223-1226.
20. I.M. Martínez-Pérez, G. Zhang, Z. Ignatova, K.-H. Zimmermann, Biomolecular autonomous solution of the Hamiltonian path problems via DNA hairpin formation, *Int. J. Bioinformatics Res. Applic.* (2005) in print.
21. J. Kendrew (Ed.), *Encyclopedia of Molecular Biology*, Blackwell Pub., 1994.
22. J.Y. Lee, H.W. Lim, S.I. Yoo, B.T. Zhang, T.H. Park, Efficient initial pool generation for weighted graph problems using parallel overlap assembly, *Preprint* (2004).
23. U. Feldkamp, W. Banzhaf, H. Rauhe, DNA sequence generator: A program for the construction of DNA sequences, *Proc. 7th DIMACS Meeting on DNA Based Computers* (2001) 23-32.
24. Q. Ouyang, P.D. Kaplan, S. Liu, A. Libchaber, DNA solution of the maximal clique problem, *Science* 278 (1997) 446-449.
25. G. Zhang, DNA computing: Using hairpin method to solve graph problems, *Project Work*, TUHH, 2005.

## Technical Reports in this Series:

Technical Report 93.6 (November 1993)  
O. Knüppel: A Multiple Precision Arithmetic for PROFIL

Technical Report 93.7 (December 1993)  
S. M. Rump: Verification Methods for Dense and Sparse Systems of Equations

Technical Report 94.1 (February 1994)  
C. Jansson, O. Knüppel: Numerical Results for a Self-Validating Global Optimization Method

Technical Report 94.2 (June 1994)  
C. Jansson: On Self-Validating Methods for Optimization Problems

Technical Report 94.3 (December 1994)  
C. Jansson: Some Properties of Linear Interval Systems and Applications

Technical Report 95.1 (March 1995)  
S. M. Rump: Improved Iteration Schemes for Validation Algorithms for Dense and Sparse Nonlinear Systems

Technical Report 95.2 (March 1995)  
S. M. Rump: Expansion and Estimation of the Range of Nonlinear Functions

Technical Report 95.3 (May 1995)  
S. M. Rump: Bounds for the Componentwise Distance to the Nearest Singular Matrix

Technical Report 95.4 (October 1995)  
O. Knüppel: A PROFIL/BIAS Implementation of a Global Minimization Algorithm

Technical Report 95.5 (October 1995)  
S. M. Rump: Perron-Frobenius like Theorems for not Sign-Restricted Matrices

Technical Report 96.1 (July 1996)  
S. M. Rump: Almost sharp bounds on the componentwise distance to the nearest singular matrix

Technical Report 96.2 (August 1996)  
S. M. Rump: Theorems of Perron-Frobenius type for matrices without sign restrictions

Technical Report 96.3 (November 1996)  
K.-H. Zimmermann: Integral Hecke Modules, Integral Generalized Reed-Muller Codes, and Linear Codes

Technical Report 96.4 (December 1996)  
C. Jansson, J. Rohn: An Algorithm for Checking Regularity of Interval Matrices

Technical Report 97.1 (February 1997)  
K.-H. Zimmermann, G. Ehlers, W. Brandt, F. A. M. Bouchard, J. Diedrichsen, T. Möller: Das ELLA 2000 Mikrocontroller-Projekt

Technical Report 97.2 (June 1997)  
K.-H. Zimmermann: A Class of Double Coset Codes

Technical Report 97.3 (November 1997)  
C. Jansson: An NP-hardness result for nonlinear systems

Technical Report 98.1 (March 1998)  
C. Jansson: Construction of convex lower and concave upper bound functions

Technical Report 98.2 (July 1998)  
G. Ehlers, K.-H. Zimmermann: Einführung in den Hardware-Entwurf mit ELLA

Technical Report 98.3 (October 1998)  
S.M. Rump: Fast and Parallel Interval Arithmetic

Technical Report 98.4 (October 1998)  
S.M. Rump: INTLAB – Interval Laboratory

Technical Report 99.1 (March 1999)  
O. Knüppel: PROFIL/BIAS V 2.0

Technical Report 99.2 (March 1999)  
J.-P. M. Zemke: b4m – A free interval arithmetic toolbox based on BIAS

Technical Report 99.3 (October 1999)  
N. Suresh Babu, Karl-Heinz Zimmermann: A Short Introduction to Quaternary Codes

Technical Report 99.4 (October 1999)  
C. Jansson: Quasiconvex Relaxations Based on Interval Arithmetic

Technical Report 00.1 (October 2000)  
Karl-Heinz Zimmermann: DNA Algorithms for Binary Linear Codes

Technical Report 01.1 (January 2001)  
Karl-Heinz Zimmermann: The Patch Selection Problem and Discrete Stochastic Dynamic Programming

Technical Report 01.2 (March 2001)  
Tangha Tina Lai, Karl-Heinz Zimmermann: A Software Platform for the Sticker Model

Technical Report 01.3 (June 2001)  
Prashant Batra: Omitted Values and Real Robust Schur Stability

Technical Report 02.1 (June 2002)  
C. Jansson: Rigorous Lower and Upper Bounds in Linear Programming

Technical Report 03.1 (January 2003)  
C. Jansson: A rigorous lower bound for the optimal value of convex optimization problems

Technical Report 04.1 (January 2004)  
T. Ogita, S.M. Rump, and S. Oishi: Accurate Sum and Dot Product

Technical Report 04.2 (June 2004)  
Prashant Batra: On Small Circles Containing Zeros and Ones of Analytic Functions

Technical Report 05.1 (February 2005)  
Israel M. Martínez-Pérez, Zoya Ignatova, Zhang Gong, Ming-Woei Lau, Karl-Heinz Zimmermann: Self-Assembly of DNA Molecules

Technical Report 05.2 (April 2005)  
Israel M. Martínez-Pérez, Zoya Ignatova, Zhang Gong, Ming-Woei Lau and Karl-Heinz Zimmermann: A Programmable Computing Machine Operating Autonomously at the Molecular Scale

Technical Report 06.1 (May 2006)  
Israel M. Martínez-Pérez, Zoya Ignatova, and Karl-Heinz Zimmermann: An Autonomous DNA Model for Finite State Automata

Technical Report 06.2 (May 2006)  
Israel M. Martínez-Pérez, Zoya Ignatova, and Karl-Heinz Zimmermann: An Autonomous DNA Model for Stochastic Finite State Automata