

## A Study on DNA Based Computation

Swarnendu Mukherjee, Debashis Ganguly, Swarnendu Bhattacharya

Computer Science and Engineering Department, Heritage Institute of Technology, Kolkata, India,  
mukherjee.swarnendu@gmail.com, DebashisGanguly@gmail.com, b.swarnendu@gmail.com

### **Abstract**

*DNA computing is an alternative method of performing computations. DNA computing is fundamentally similar to parallel computing in that it takes advantage of the many different molecules of DNA to try many different possibilities at once. For certain specialized problems, DNA computers are faster and smaller than any other computer built so far. It is based on the observation that in general it is possible to design a series of biochemical experiments involving DNA molecules which is equivalent to processing information encoded in these molecules. In case of classical computing, we use electronic logic gates which allow for storing and transforming information. Designing of an appropriate sequence or a net of “store” and “transform” operations (in a sense of building a device or writing a program) is equivalent to preparing some computations. In DNA computing, the principle is quite same. But, the main difference is the type of computing devices, since in this new method of computing instead of electronic gates DNA molecules are used for storing and transforming information. From this follows that the set of basic operations is different in comparison to electronic devices but the results of using them may be similar. This paper presents a complete study on DNA computation in literature.*

**Keywords:** *Classical computation, DNA molecules, computation, parallel computing, advantages, applications*

### **INTRODUCTION**

The first person who spoke about the possibility of performing computations at the molecular level was probably Richard Feynman. His ideas had to wait over twenty years to be implemented. They evolved in two directions one of them being quantum computing and the other DNA computing.

The later approach was taken up by Leonard Adleman in 1994. He took a giant step towards a different kind of chemical or artificial biochemical computer. He used fragments of DNA to compute the solution to a complex graph theory problem. Adleman's method utilizes sequences of DNA's molecular subunits to represent vertices of a network or graph. Thus, combinations of these sequences formed randomly by the massively parallel action of biochemical reactions in test tubes described random paths through the graph. Using the tools of biochemistry, Adleman was able to extract the correct answer to the graph theory problem out of the many random paths represented by the product DNA strands [1].

Adleman's experiment took a revolutionary step in the world of computation of large and complex problems. In the NP-Complete problems, the space-time complexity of the program increases exponentially with the number of vertices in the graph or the network used. The DNA strands can store information in very large scale. One pound of DNA has the capability to store more information than all the electronic computers ever built. More precisely, one cubic cm of DNA can hold approximately 10 terabytes of data. DNA Computing is also used in different other fields, as a replacement of the silicon chips to biological computers [3, 6].

Of late the research effort in the area of DNA computing concentrates on four main problems: designing algorithms for some known combinatorial problems, designing new basic operations of

“DNA computers” (they are some biochemical procedures whose results may be interpreted as results of some computations), developing new ways of encoding information in DNA molecules and reduction of errors in DNA based computations [4, 2, 5, 8].

In this paper, we have tried to present a detail survey on DNA Computing, its relative advantages, applications and we hope that this work will definitely provide a concrete overview on all possible aspects in this field.

#### ORIGIN OF DNA COMPUTATION

The word ‘DNA Computing’ is based on the fundamental genetic material of all organisms. A DNA (Deoxyribonucleic Acid) is the genetic blueprint of all organisms in this planet. The structure of these molecules, discovered by Watson and Crick [9], is fundamental for the possibility of using them for encoding and processing information. The DNA is a double stranded helical structure made of protein nucleotides on a base of sugar phosphate.

There are 4-types of nucleotides namely, Adenine, Thymine, Cytosine, and Guanine. These nucleotides form bonds in pairs of Adenine-Thymine (A-T) and Cytosine-Guanine (G-C). This DNA holds the genetic information about the cell. This information is the code used within cells to form proteins and is the building block upon which life is formed. From computer science point of view a DNA strand is a word over alphabet  $\Sigma_{DNA} = \{A, C, G, T\}$ . In a DNA computer, computation takes place in test tubes or on a glass slide coated in 24K gold. The input and output are both strands of DNA, whose genetic sequences encode certain information. A program on a DNA computer is executed as a series of biochemical operations, which have the effect of synthesizing, extracting, modifying and cloning the DNA strands.

The idea of DNA computing is similar (to some extent) to the non-deterministic Turing machine. Indeed, in many DNA based algorithms at the beginning there are created DNA sequences encoding all feasible (not necessarily optimal) solutions to a given problem and a lot of other sequences which do not encode any feasible solutions. In the next steps the algorithm eliminates from this input set of sequences those which are not solutions to the considered problem and also those ones which do not encode optimal solutions. This elimination is performed successively and the power of DNA computers lies in their massive parallelism (almost) all encoding sequences are processed simultaneously. At the end of the computations in the set of DNA molecules there remain only those which encode the optimal solution.

The elimination of “bad” sequences is usually done in such a way that the algorithm checks successively if the sequences possess some properties necessary for every string which potentially could encode the problem solution. If they do not have these properties, they are eliminated from the set.

DNA computing is utilizing the property of DNA for massively parallel computation. As DNA threads provide a large number of combinations of the codes and they are repeated large number of times hence they provide an easy way to generate the parallel computation of the system provided that there is an appropriate setup and enough supply of DNA to process the large programs.

The utilization of DNA is at times much faster than the conventional silicon chip processors. The multitude of the possible combinations of the nucleotides can generate huge number of sequences and hence can speed up the processing of complex data of large programs.

#### BASIC OPERATIONS IN DNA COMPUTATION

The basic operations of DNA algorithms are usually designed for selecting sequences which satisfy some particular conditions. On the other hand, there may be different sets of such basic

operations. In fact, any biochemical procedure which may be interpreted as a transformation (or storing) information encoded in DNA molecules may be treated as a basic operation of algorithms based on this technology. One of the possible set of such operations is described below [7]:

- Merge: Given two test tubes  $N1$  and  $N2$  create a new tube  $N$  containing all strands from  $N1$  and  $N2$ .
- Amplify (Copying): Given tube  $N$ , create a copy of them by using the Polymerase Chain Reaction (PCR).
- Detect: For a tube  $N$ , it returns true if  $N$  contains at least one DNA strand, otherwise it returns false.
- Separate: For a tube  $N$  and a word  $w$  over alphabet  $\Sigma_{DNA}$ , create two tubes  $+(N, w)$  and  $!(N, w)$ , where  $+(N, w)$  consists of all strands from  $N$  containing  $w$  as a substring and  $!(N, w)$  consists of the remaining strands.
- Length-Separate: For a tube  $N$  and a positive integer  $n$  create tube  $(N, \leq n)$  containing all strands from  $N$  which are of length  $n$  or less.
- Position-Separate: For a given tube  $N$  and a word  $w$  over alphabet  $\Sigma_{DNA}$  create tube  $B(N, w)$  containing all strands from  $N$  which have  $w$  as a prefix and tube  $E(N, w)$  containing all strands from  $N$  which have  $w$  as a suffix.

Each of the above operations is a result of some standard biochemical procedure.

#### ENCODING TECHNIQUE OF DNA INSTANCES

A significant difference between classical computation and DNA computation is a very close relationship between the algorithm and the molecules encoding the problem instance in the case of molecular computations.

In electronic computers, the input data are stored in standard memory device or in registers and the programmer must not take care of their structure. Again, at the time of designing algorithms which will be implemented on a classical computer, one also need not consider the structure of the registers for storing problem data.

But, the situation is completely different in the case of DNA computing. First, there is no clear distinction between an algorithm and a program. Second, for each problem its own encoding scheme must be developed, which is very closely connected with the algorithm.

The reason could be the lack of some standard architecture of the “DNA computer”. Indeed, in some sense, for each problem being solved by DNA molecules a new “hardware architecture” is developed.

So, the designing of DNA molecules for encoding problem instances is an important part of the process of algorithms developing and usually these two issues cannot be separated.

The molecules have to be designed in such a way that when poured into one test tube in a certain physical conditions they will create double stranded DNA molecules encoding potential solutions to the considered problem.

#### APPLICATION OF DNA COMPUTATION

**Solution of NP-Complete Problems:** DNA computation technique is precisely used to find the solution of NP-Complete problems in an efficient manner. The NP-Complete problems are an interesting class of problems whose status is unknown. The properties of NP-Complete problems are:

- No polynomial-time algorithm has been discovered for solving these problems.
- No polynomial lower bound in respect of time overhead has been proven for these problems.

There are various NP-Complete problems like Knights Problem, Traveling Salesman, etc. In the Traveling Salesman problem, as the number of cities grows, the number of possible path combinations soars. For example, if there are nine cities, there are 180,000 possible paths. Eleven cities would have 19.8 million paths, 13 cities would have about 3 billion paths, and 17 cities would have about 200 trillion paths. For larger and larger numbers of cities, brute force attempts to calculate all paths would quickly overwhelm even a supercomputer. But, with the help of DNA computing this huge complexity can be reduced to a large extent. Adleman had first implemented the solution of this problem using the concepts of DNA.

The algorithm of Traveling Salesman can be states as:

1. Generate all possible routes.
2. Select routes that start with the initial city and end with the destination city.
3. Select itineraries with the correct number of cities.
4. Select itineraries that contain each city once.

Adleman, created randomly sequenced DNA strands 20 bases long to chemically represent each city and a complementary 20 base strand that overlaps each city's strand halfway to represent each street. By placing a few grams of every DNA city and street in a test tube and allowing the natural bonding tendencies of the DNA building blocks to occur, the DNA bonding created over  $10^9$  answers in few hours. This is the most time consuming process in the method of DNA Computing. It took about a month for Adleman to create the DNA sequences individually for each city and then leave them for natural bonding.

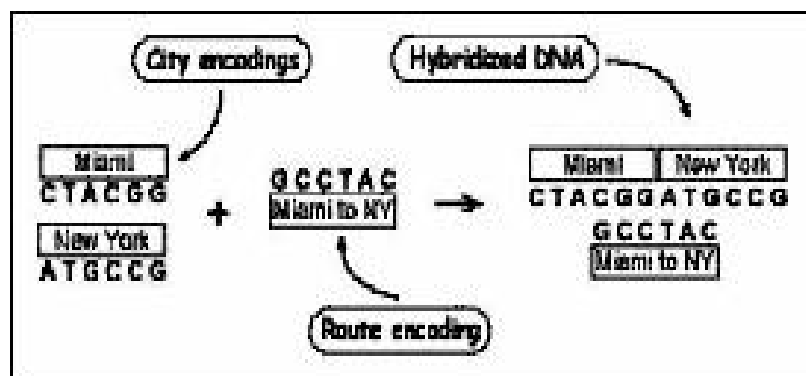
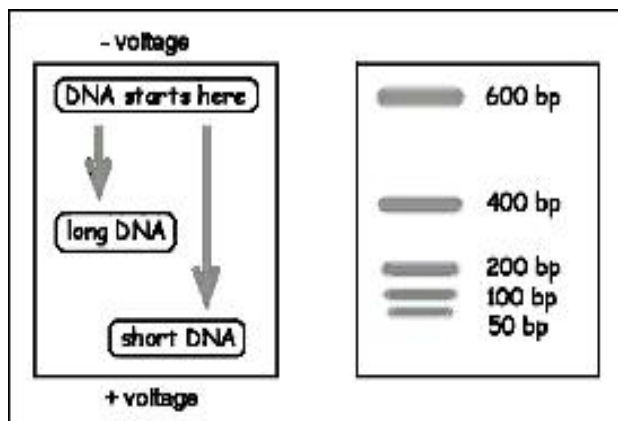


FIGURE – 1

After having all the possible routes, the DNA strands which has the code with the starting city and ends with the destination city are selected. This process is accomplished by a biochemical process called Polymerase Chain Reaction. In this reaction, the required DNA strands are extracted from the complete DNA thread by repeating 3 basic steps – denaturation, annealing and extension. After, the extraction of the DNA strand, the strands are exponentially copied for parallel processing. In each step of the expansion, the DNA strands are copied in the power of 2. So, if the process is repeated for 36 cycles which consumes a time of about an hour, the total number of replicated DNA strands will be  $2^{36}$  i.e. there will be approximately 68-billion copies of the DNA.

The third step deals with finding out those paths that has the correct number of cities. With the code of the starting and ending cities at the two end of the DNA strands, the strands with the correct number of cities will be the shortest among the rest. To separate the shortest DNA strands from other long strands, another biochemical process is used. This process is known as Gel-Electrophoresis. In this process, a potential difference is implied between the two ends of a gel. The DNA strands are then raced through the gel starting from the region of negative polarity. The DNA strands race towards the region of positive polarity. The speed of this movement of the DNA strands varies with their sizes. The DNA strands with smaller or shorter size races towards the positive polarity much faster than that of the longer or larger DNA strands. With the aid of this

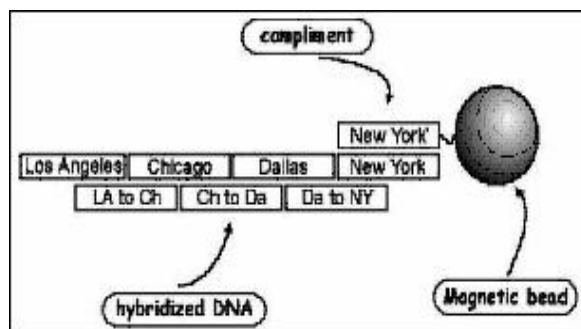
process, the shorter DNA strands can be easily separated from the the larger DNA strands or in other words the DNA strands with the correct number of cities can be separated from the rest of the unwanted DNA strands which have huge repetition of the different cities.



Determination of Path from source to destination

FIGURE – 2

The fourth and the final step is to find out those cities which have each city only once. The DNA nucleotides have a high affinity towards their counterparts or their complements. As a result of which, it is often found that a complement of a particular code gets attached to the hybridized DNA strand. This strand affects the computation process. To remove this unwanted DNA strand from the created hybrid DNA strand a strong magnetic bead is used which pulls out the unwanted DNA strand from the created hybrid DNA strand to acquire the desired result from the DNA strands. Now, by counting the total number of DNA strands results in the number of paths from the starting city to the ending city while travelling to each intermediate city once. The study of the DNA strands reveals the actual path from the starting city to the ending city traveled by the s



Determination of Optimum Path

FIGURE – 3

Thus we see in a graph with  $n$  vertices, there are a possible  $(n-1)!$  permutations of the vertices between beginning and ending vertex. To explore each permutation, a traditional computer must perform  $O(n!)$  operations to explore all possible cycles. However, the DNA computing model only requires the representative oligos. Once placed in solution, those oligos will anneal in parallel, providing all possible paths in the graph at roughly the same time. That is equivalent to  $O(1)$  operations, or constant time. In addition, no more space than what was originally provided is needed to contain the constructed paths.

**DNA Computers:** Sooner or later, the competition of creating faster and smaller microprocessor is bound to hit a wall. Microprocessors made of silicon will eventually reach their limits of speed and miniaturization. Chip makers need a new material to produce faster computing speeds. Millions of natural supercomputers exist inside living organisms, including your body. DNA (deoxyribonucleic acid) molecules, the material our genes are made of, have the potential to perform calculations many times faster than the world's most powerful human-built computers. DNA might one day be integrated into a computer chip to create a so-called biochip that will push computers even faster the other computing devices.

The Rochester team's DNA logic gates are the first step toward creating a computer that has a structure similar to that of an electronic PC. Instead of using electrical signals to perform logical operations, these DNA logic gates rely on DNA code. They detect fragments of genetic material as input, splice together these fragments and form a single output. For instance, a genetic gate called the "And gate" links two DNA inputs by chemically binding them so they're locked in an end-to-end structure, similar to the way two Legos might be fastened by a third Lego between them. The researchers believe that these logic gates might be combined with DNA microchips to create a breakthrough in DNA computing.

#### ADVANTAGES AND DISADVANTAGES OF DNA COMPUTATION

Like other field of computational techniques, DNA Computing has also got several advantages as well as disadvantages. The relative advantages and disadvantages of this technology are discussed below.

**Advantages:** The advantages of DNA based computation method are:

- **Parallel Computing:** The process followed in DNA computing creates massive parallel processing. The DNA strands are duplicated very rapidly in a short period of time and these duplicated strands are used to generate the environment for parallel processing of the data.

- **Light Weight:** The DNA strands can hold very high amount of information in relatively very small space. Only 1 cubic centimeter of DNA can store information equivalent to 1 Tera Byte of data and 1 pound of DNA possess more computation power than the entire computer ever created.

- **Low Power:** The only power that is required by the process it to protect the DNA strands from denaturing. So, the consumption of power decreases rapidly with the use of the DNA computing.

- **Fast Computation:** The DNA computers are much faster than the silicon chip power computers. The processes that take days to complete their execution can be successfully executed in hours.

**Disadvantages:** The disadvantages of DNA based computation method are:

- **Time:** The initials steps of DNA computing are very time consuming. The method of extraction of the DNA strands to generate the hybridized DNA strands takes weeks or even months of work in the labs.

- **Occasionally Slower:** For simpler problems, the method of DNA computing produces slower results whereas the silicon chip powered computer can generate the result in much short time.

- **Retrieve Answer from Solution:** By the method of DNA computing the solutions of a particular problem can be generated in much shorter time. But, to find the actual answers from the solution set result in a very much time consuming and complicated task. The solutions are all in codes that are sequences of DNA strands. Hence extraction of the answer from the solution is another troublesome job.

- **Reliability:** The process of DNA computing falls into unreliable situations due to the unwanted pairing of the DNA nucleotides with other DNA nucleotides resulting in unwanted DNA

strands. Moreover the extraction process is not always 100% efficient as a result of which there is the high chance of inconsistent solutions.

#### CURRENT RESEARCH ON DNA COMPUTATION

The DNA computers are the upcoming technology in the modern world. The DNA computers are a probable replacement of the silicon chip computers. There are a lot of research are running on this topic for further enhancement.

**Test Tube Programming Language:** The different methods of the DNA computing can be expressed using this language. It has mainly four parts – merge, amplify, detect and separate. With the help of the test tube programming language all problems can be expressed. For example, the Hamiltonian problem,

```
test-tube = make-tube (edge-strands, vertex-strands)
start-selected = filter-sequence (test-tube, start-vertex)
end-selected = filter-sequence (start-selected, end-vertex)
nvertices = filter-length (end-selected, path-length)
all-vertices = filter-vertices (nvertices, list-of-rest-vertices)
finalresult = if (test-tube-empty (all-vertices) )then
  "There is no Hamilton Path"
  else "There is Hamilton Path"
```

**Soft Molecular Computing:** The DNA computing utilizes the complex reaction of the molecules and the molecular biology to effect the computation. So, these computations can only be performed in the laboratories with sophisticated instruments for performing such complex tasks. But laboratory works are not accurate and hence they are inefficient, unreliable and inconsistent. The major problem that is faced in DNA computing is to test the protocols. As a result of which a new platform is created to address this problem. This platform is called EDNA.

EDNA, integrated software platform is created to address the problem of reliability, efficiency and scalability. EDNA is object oriented so that is can be easily modified and it could evolve as the field progresses. EDNA allows having the advantage of digital computers to gain realistic insights on actual test tube performance of a protocol before they are carried out in the laboratory. Thus it reduces the cost of performing experiments for verifying expensive protocols in the laboratory unless it is verified by the software. EDNA also provides a research tool that makes it possible to use the advantages of conventional computing to make DNA computing reliable.

**DNA Steganography:** Steganography is the art and science of writing hidden messages in such a way that no one apart from the intended recipient knows of the existence of the message. In this method, the message usually encrypted is hidden behind some other text, image, audio or video.

In DNA Steganography, the message is first coded and then stored in the DNA. Initially, the alphabets in the message are converted into codes of DNA sequences. Then these DNA sequences are placed in a hybridized DNA strand with markers at their ends. Now, the hybridized DNA strands are placed into any animal DNA and mixed with other DNA strands. This mixture is placed on a piece of paper and dried to form microdots.

The microdots contain billions of DNA strands. The messages present in the DNA strands are not only very tough to detect but also the microdots contains billions of such DNA strands among which only one has the message stored into it. To decrypt the message, the key is the marker present at the two ends of the message. Without knowing the marker it is almost impossible to find the message from billions of DNA strands [10].

Hence, DNA Steganography shall soon become most secure technique of data encryption in the modern world.

**DNA Authentication:** The first DNA Authentication chip was designed by Taiwan. Inside the chip there is a synthesized DNA which can be recognized by a device which is similar to an ID card reader. The synthesized DNA that is present inside the chip can be enough large in size, as a result the reading device and the chip will have a tough bonding nature that could not be easily faked. The DNA Authentication device takes almost 2 seconds to verify the chip for the purpose of authentication.

The DNA chip can also be used on passports, credit cards, debit cards, membership cards, driver's licenses, automobile license plates, CD s, VCD s, DVD s, notebooks, PDA s, and computer software. Surprisingly, though the DNA authentication system is much advanced but the price of such a chip is comparable to an IC chip.

#### **FUTURE POSSIBILITIES**

A year ago, researchers from the Weizmann Institute of Science in Rehovot, Israel, unveiled a programmable molecular computing machine composed of enzymes and DNA molecules instead of silicon microchips. "This re-designed device uses its DNA input as its source of fuel," said Ehud Shapiro, who led the Israeli research team. This computer can perform 330 trillion operations per second, more than 100,000 times the speed of the fastest PC.

While a desktop PC is designed to perform one calculation very fast, DNA strands produce billions of potential answers simultaneously. This makes the DNA computer suitable for solving "fuzzy logic" problems that have many possible solutions rather than the either/or logic of binary computers. In the future, some speculate, there may be hybrid machines that use traditional silicon for normal processing tasks but have DNA co-processors that can take over specific tasks they would be more suitable for.

#### **CONCLUSION**

In this paper, basic concepts of DNA based computations have been presented and illustrated by some algorithms known from the literature. This new model of computing requires some new paradigms and offer, at least in principle, new possibilities for solving hard problems. One of the most important properties of DNA computing is its real massive parallelism (in "DNA computers" billions of DNA molecules play the role of elementary processors). The model of DNA computing is similar, to some extent, to the non-deterministic Turing machine. Obviously, the sizes of instances possible to solve by the methods of this type are limited by the amount of DNA molecules necessary for encoding all potential solutions to the problem under consideration.

Of late, we see a hike in the field of parallel computing. The probable reasons for this could be its high throughputs, performance etc. So, the trend of computation should move in this way for the better utilization of resources. DNA computation is one of the leading solutions in this regard. Proper use of this concept can provide us a low power and fast computing system which is mostly required for today's world.

One of the open questions concerning DNA computing is its universality. It is still not clear if it will be possible to develop a kind of universal "DNA computer" able to solve any algorithmic problem. Most of the biochemical procedures solving some algorithmic problems proposed until now can be used to solve only one particular problem and can be seen as some simple specialized "DNA computers". It should be also noted that it is relatively easy to develop DNA based algorithms solving some set or graph theoretical problems, while it is much more difficult to design such methods solving arithmetic problems. It follows from the general framework of DNA computing where the operations on set of molecules are the most natural ones. On the other hand, there are attempts to model by DNA molecules deterministic Turing machines which may eventually lead to a construction of universal molecular computer.



So, lastly, we can conclude that in spite of having several advantages, this field has not reached its ultimate goal. Research should be carried out to make this field a developed from the developing one. It is hoped that this emerging technology will provide us a new definition of computing in the near future.

#### REFERENCES

- [1] L. Adleman, "Molecular computations of solutions to combinatorial problems", *Science* 266 (1994), pages 1021-1024.
- [2] L. F. Landweber, E. B. Baum (eds.), "DNA Based Computers II", American Mathematical Society, 1999.
- [3] R. J. Lipton, "DNA solution of hard computational problems", *Science* 268 (1995), pages 542-545.
- [4] R. J. Lipton, E. B. Baum (eds.), "DNA Based Computers", American Mathematical Society, 1996.
- [5] Marathe, A. E. Condon and R. M. Corn, "On combinatorial DNA word design", *Journal of Computational Biology*, Vol. 8 (2001), pages 201-219.
- [6] Q. Ouyang, P. D. Kaplan, S. Liu and A. Libchaber, "DNA solution of the maximal clique problem", *Science* 278 (1997), pages 446 - 449.
- [7] G. Paun, G. Rozenberg and A. Salomaa, "DNA Computing". *New Computing Paradigms*, Springer-Verlag, Berlin 1998.
- [8] S. Roweis and E. Winfree, "On the reduction of errors in DNA computation", *Journal of Computational Biology*, Vol. 6 (1999), pages 65-75.
- [9] J. D. Watson and F. H. C. Crick, "A structure of deoxiribose nucleic acid", *Nature*, Vol. 171 (1953), pages 737-738.
- [10] L. M. Adleman, P. W. K. Rothmund, S. Roweis, and E. Winfree. On applying molecular computation to the data encryption standard. In *Proceedings of the Second Annual Meeting on DNA based Computers*, 1996, pages 31 - 44.

---

Article received: 2009-01-06