Paper

Evolutionary Algorithm that Designs the DNA Synthesis Procedure

Maciej Michalak and Robert Nowak[a]

[a] *Faculty of Electronics and Information Technology, Warsaw University of Technology, Warsaw, Poland*

Abstract—Chemical synthesis of nucleotide chains is very erroneous for long sequences. Often a gene is constructed from short fragments joined with the use of complementary helper chains. The number of possible potential solutions for a long gene synthesis is very large, therefore a fast automated search is required. In the presented approach a modified method of long DNA construction is proposed. A computer program that searches for an optimal solution in the space of potential synthesis methods has been developed. This software uses an evolutionary algorithm for global optimization and a hill-climbing algorithm for local optimization. The long DNA construction method was tested on random sequences. The results are very promising. The next step is to perform experiments in a biotechnological wet laboratory involving DNA strand synthesis using the method designed by the presented software.

Keywords—bioinformatics, gene synthesis, optimum searching.

1. Introduction

The deoxiribonucleic acid (DNA) strand contains the genetic instructions of majority of living organisms. A need for artificial synthesis of these strands is important in biology and medicine.

A DNA molecule is formed from two separate DNA strands. Each strand may be viewed as a sequence of nucleotides or bases, in which nucleotides are as follows: adenine, cytosine, guanine, and thymine. Two strands are connected by hydrogen bond and this connection is selective – adenine bonds only with thymine and guanine bonds with cytosine. This specific interaction between base pairs is called *complementarity*, it is critical for all the functions of DNA and makes the information in the double-stranded DNA molecule duplicated on each strand. The nucleotide has a natural orientation denoted (according to chemical convention) as 5' and 3' end.

The DNA strand contains genetic information which encodes an amino acid sequence used for protein synthesis. There are 20 amino acids encoded by the sequence of three nucleotides ($4^3 = 64$ possible triplets), called codons. The genetic code is redundant because most amino acids can be encoded in several ways. This redundancy is essential in gene (protein-encoding molecule) synthesis since the DNA sequence can be modified for easier assembly without changing the genetic information. However it must be remembered that the frequencies of codons should satisfy particular preferences of the host (organism used for biosynthesis).

A DNA molecule of a given sequence could be chemically synthesized by repeatedly adding nucleotides [1]. To obtain a desired molecule, nucleotides that have protection groups are sequentially coupled to the growing chain in the order required by the sequence of the product. Despite the yield of this step is about 98%, the length of created strand is limited to 70 *base pairs* (bp) [2]. This restriction is significant for many biological and medical processes, e.g., for peptide biosynthesis because genes length is usually greater than 300 bp, typically 1000 bp.

Long DNA molecules synthesis techniques have been developed to obtain such molecules from smaller parts. A method of long DNA molecule construction from smaller DNA molecules is called *protocol*. Polymerase cycling assembly (PCA) is the most widely used technique [3], [4], [5] to produce long (e.g. 1000 bp) DNA chains. This technique uses the same reactions and reagents as polymerase chain reaction (PCR), but additionally the DNA polymerase [2] amplifies a complete sequence of DNA, as depicted in Fig. 1. The typical length of a fragment is 50 bp and the overlapping area is 20 bp. The correct syntheses of 800 bp strands have been reported [6].

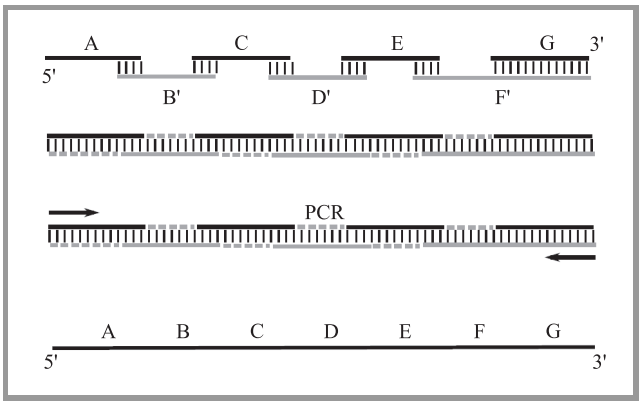

Fig. 1. The polymerase cycling assembly (PCA) used to create long DNA molecules from smaller parts. The shorter fragments are chemically synthesized, then create a longer molecule (hybridization), next DNA polymerase produces a complete sequence of DNA, then PCR with specific starters (primers) amplifies the long DNA strands. Finally, the isolation based on molecule length is performed. The fragments of complementary strand are denoted using a prime symbol.

In the presented solution we consider protocols different from PCA. Shorter fragments are separately chemically synthesized, as in the PCA, but the possibility of synthe-

sizing the parts of different length as well as carrying out the reaction at a decreasing temperature [7] is considered. Next, we accept changes in the resulting sequence if these changes do not affect the protein sequence, i.e., we include the possibility of codon substitution. Finally, the possibility of performing the reaction in separate tubes and then joining the results is considered. Our application generates a much larger number of protocols to synthesize a given strand than a typical PCA oriented algorithm. Each protocol is assigned a quality measure, which is optimized in the space of protocols using a combination of evolutionary algorithm for global optimization and a hill-climbing algorithm to perform fine tuning.

2. Synthesis Protocols

The base synthesis protocol consists of fragments of the desired DNA strand (molecules labeled A, C, E, G in Fig. 2) and the *helper chains* (molecules from complementary strands: B', D' and F' in Fig. 2). The helper chains are complementary to the two adjacent fragments and are used for joining.

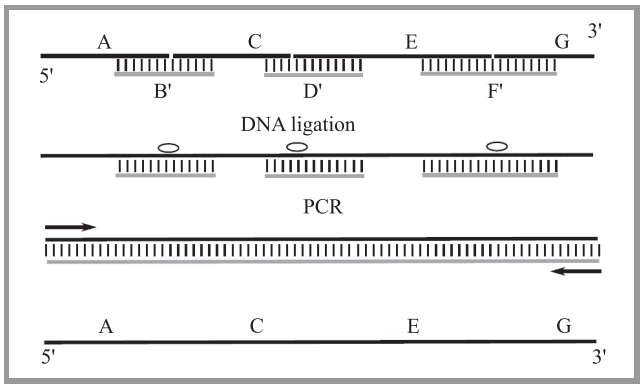

Fig. 2. Long DNA base synthesis protocol. Shorter fragments are chemically synthesized, then during hybridization and ligation the longer molecule is formed, finally PCR with specific starters (primers) amplifies the correct DNA strands. If a correct molecule cannot be created because the fragments fold in an incorrect way, the reaction is performed in separate tubes (complex protocol).

Every pair of DNA strands (also of a single strand) folds in the temperature dependent on its nucleotide sequences, therefore the idea is to start from the temperature high enough to unfold all the fragments and then gradually decrease it so that strands can join, as shown in Fig. 3.

Of course, there is no guarantee for every set of fragments that they will join in a desired order. We examine every possible pair of strand from the solution – both fragments and helper chains – including pairs consisting of two identical molecules (fragment with itself). These examinations use DNA secondary structure prediction algorithms, which tells us how the strands can fold and what temperature is needed to unfold them. The results ordered by

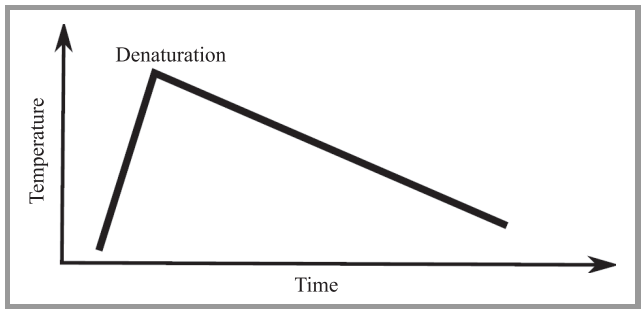

Fig. 3. Temperature in probe during synthesize reaction. Firstly, dilution is heated to denaturate, then cooled slowly.

the descending temperatures define the sequence of strands' foldings.

It is desirable that each time the fragments in solution are either correctly folded or unfolded. The correctly folded fragment is created by a pair that folds into a coherent double-stranded structure with dangling single-stranded ends. If the pair contains a strand that is already folded, then it all combines into one structure, i.e., into a single fragment with two adjacent helper chains or two fragments connected by the helper chain. Ultimately, the whole long DNA strand is composed and can be subjected to further treatment i.e., to build bounds between the chains by DNA ligase, as shown in Fig. 2.

Simulations for the base protocol for random sequences of length 60 – 80 bp have shown that about 1% of possible synthesis protocols are proper. In these simulations, we assume that the random sequence is created from 3 strands, two fragments and one helper chain. The fragments were calculated by drawing, with uniform distribution, the position on the input sequence that divides it into two subsequences of length 20 – 40 bp. The helping chain was complementary to a region near this position. Most of base protocols are unsuitable for the synthesis due to incorrect foldings. Fortunately the number of good solutions can be increased by extending the base protocol to a complex one. Long strand assembling can be done in steps, as shown in Fig. 4. In each step, conflicting strands (i.e., strands forming improper pair) are separated into different probes

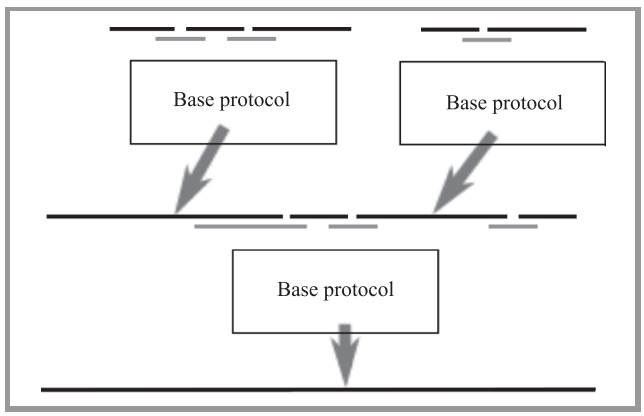

Fig. 4. Long DNA complex protocol, the multiple tubes are used.

so the base protocol can be applied for each probe. Chains formed by the base protocol become the input strands for the next and the procedure is repeated until a full long strand is obtained.

In order to determine the percentage of proper solutions five random sequences were generated. For each sequence four ranges of fragment length were considered: 6 – 12, 10 – 20, 14 – 28, 20 – 40. For every variant sequences were truncated to the length allowing them to be constructed from a given number of fragments. For each case (i.e., fragment length: 10 – 20, three fragments, first sequence truncated to 45 nucleotides) 10,000 solution candidates were randomized, simulated and searched for proper base protocols. The considered sample was relatively small, therefore the results for different variants of simulations were strongly dispersal, but some trends were visible. As it can be seen in Table 1, with the increase of the fragmentation size, the percentage of proper solutions strongly decreases. However, the contribution of complex protocols to all proper protocols grows.

Table 1
Percentage of proper solutions

Number of fragments	Base protocols [%]	Complex protocols [%]
2	5 – 20	0
3	0.2 – 5	0.02 – 0.5
4	0.05 – 0.3	0.02 – 0.25
5	0 – 0.1	0.01 – 0.13

The simulations revealed that the number of proper protocols grew insignificantly. The reason for this poor result lies in the length of helper chains in the consecutive steps. Probability of folding increases with the total length of corresponding strands, therefore two fragments are more likely to fold with each other than with the helper chain which is at least two times shorter. The solution to this problem is to replace helper chains with longer ones and, eventually, synthesize them in separate probes as well. As the next simulations indicate, this strategy induced the increase of the number of proper protocols by up to 50% for protocols with 3 fragments and more for a larger number of fragments.

Obviously, the complex synthesis protocol is more expensive than the base synthesis protocol. The execution of each step in the laboratory takes about 24 hours, multiple probes are needed and the number of required nucleotides is noticeably greater.

3. Implementation Issues

The application searches the synthesis protocol in the space of possible basic and compound protocols using a combination of evolutionary algorithm and a hill-climbing algorithm.

3.1. Evolutionary Algorithm

Intuitively, the synthesis protocol is a solution candidate (individual) for the evolutionary algorithm [8]. The individual is represented by the target DNA strand (in a form of sequence of nucleotides), the collection of positions describing the places of fragments separation and the collection of helper chains. This set of data should meet the constraints on minimum and maximum strand length. Therefore, the initial population is a set of protocols which differ in three aspects:

– codon sequence (nucleotide sequence encoding given peptide),

– chain fragmentation,

– helper chain selection.

The initiation process first creates a the target DNA strand. The codons were chosen randomly with uniform distribution. Next, places of fragments separation are chosen randomly, with uniform distribution, having regard to the minimum and maximum length of the fragment. Finally, the helper chains are calculated, their length is also random (uniform distribution). There is an option not to randomize a codon sequence, but to optimize it earlier with the use of local search – the decision is left for the user.

All individuals are chosen for reproduction. Every parent produces a single mutated child by choosing, with the equal probability, one of the following transformations:

– replacement of the random nucleotide triplet from the sequence (change a codon),

– increment or decrement of a random position describing the separation into fragments (one fragment is shortened and the other is extended),

– one helper is shortened or extended or moving left or right.

The fitness of the individual depends on:

– the root mean square of the differences between requested and obtained codon frequencies,

– the number of nucleotides needed,

– the number of protocol steps,

and can be represented by the formula:

$$F = \frac{1}{(1 + F_C + F_N + F_L)},$$

$$F_C = w_C \sum_{j=0}^{n} \frac{(C_{rj} - C_{ij})^2}{n},$$

$$F_N = w_N \frac{N - N_{\min}}{N_{\max} - N_{\min}},$$

$$F_L = w_L \frac{L-1}{2},$$

where: C_r – required codon frequency, each organism has optimal codon frequency, available at [9], C_i – individual's codon frequency, N – number of used nucleotides, L – number of protocol's steps, w – weight of the partial evaluation.
In every generation parents and their offspring are selected for succession. There are three methods of selection to choose: ranking, tournament ($k = 2$) and proportional.

3.2. Implementation

In order to compute individual's fitness, our application is able to simulate the process of synthesis, check for eventual conflicting pairs and switch from the base to the complex protocol. Due to a good trade-off between the availability of optimization techniques, portability and extensibility, C++ language for implementation was chosen. The application is set up on the Django server which provides a graphic user interface via HTTP.
The proposed algorithm for protocol calculation is quite complex so its execution is expensive. Single individual's fitness calculation time for real-life input needs seconds or even minutes, e.g., on a typical 3 GHz CPU core, it takes an average of 90 s to compute 1000 bp gene synthesis simulation. We decided to use distributed calculations (many computers) and take advantage of GPU power. Distributed calculations use CORBA (Common Object Request Broker Architecture), individuals' fitness is considered independently, so the calculations can be delegated to different cores, processors and computers.
The DNA strands' joining order is examined using the Zuker algorithm [10] for the secondary structure prediction. It was implemented in a bottom-up dynamic programming manner. The idea is to calculate – with the use of the delivered thermodynamic data – the free energy of the smallest sub-chains and iteratively find optimal solutions to increasingly longer sub-chains. An optimal solution is characterized by the minimum free energy which corresponds to the set of base pairs forming a secondary structure.
The complexity of Zuker algorithm is $\Theta(n^4)$ and with the use of heuristics can be simplified to $\Theta(n^3 + 30^2 n^2)$. This is a highly expensive operation and indeed, the profiling tool has shown that it takes over 99% of the whole application execution time. It should be noticed that the mutation can affect changes only on 1–3 strands in a probe, therefore the best effort was made in order to minimize the number of such operation calls by storing the results and making them inheritable by mutated individuals. As a result, when the 1000 bp gene synthesis protocol is once evaluated, its derivatives are calculated 10 times faster. The improvement for different gene lengths is shown in Fig. 5.
To speed up the DNA secondary structure prediction calculations, the module was implemented in CUDA (Compute Unified Device Architecture) in order to take advantage of the GPU power. As it was explained, the algorithm computes the solution for sub-chains starting from the shortest ones. For each length every sub-chain is considered independently, therefore it is possible to make these calculations

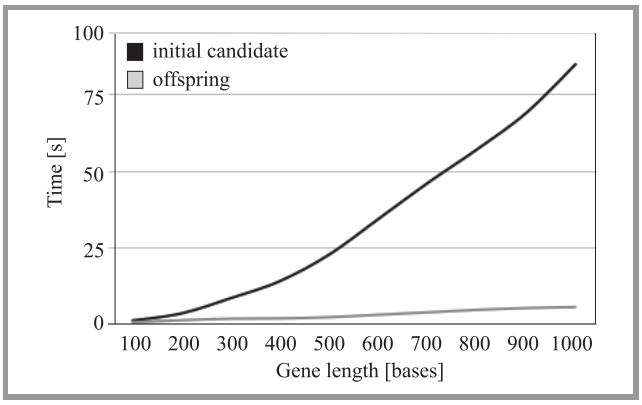

Fig. 5. Synthesis protocol evaluation time versus DNA length.

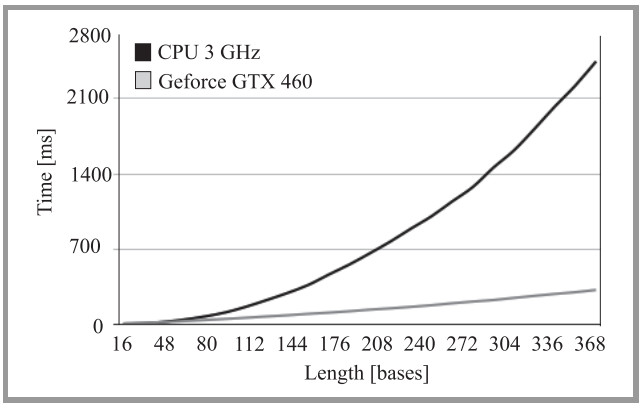

Fig. 6. Zuker algorithm's execution time for 3 GHz CPU an nVidia Geforce GTX 460 versus DNA length

parallel. As the Fig. 6. indicates, the longer strand (pair of strands) the greater speed increase – i.e., the 384 bp sequence is calculated 12 times faster on Geforce GTX 460 than on 3 GHz CPU core. It was measured that 1000 bp gene synthesis simulation with the GPU use takes 40 s on average and is 2.5 faster than CPU.

4. Results

The application was tested using a random sequence of the length 50 as an input gene, the minimum and maximum fragment length equal to 10 and 20, respectively. The search process was performed by a trial and error method and by the evolutionary algorithm in three variants: with ranking, proportional and tournament selection. For the purpose of comparison, all optimizations were set to evaluate 10,000 solution candidates, therefore, each variant was conducted for 10, 100 and 1000 generations with population size of 1000, 100 and 10 respectively. Table 2 shows the results of these experiments.
As it can be seen, when one simply draws 10,000 solution candidates (Monte Carlo method), they get only 150 correct solutions. The use of evolutionary algorithm increases this number significantly, which may indicate similarity of correct solutions.

The results of these experiments are promising, but one must remember that with the increase of the requested sequence length the time complexity grows. Due to the greater fragmentation the percentage of correct protocols also decreases. Search for the optimal synthesis protocol for real genes is greatly time consuming and that is what motivated the authors to create implementation which maximally takes advantage of all available resources.

Table 2
Properties of the solution space coverage

Method	Uniqueness [%]	Correctness [%]	Correct uniqueness [%]	Unique correct
Trial and error	100	1.5	100	150
Ranking				
1000×10	86.6	24.2	70.7	1711
100×100	51.4	55.6	45.3	2519
10×1000	20.7	33.1	12.4	410
Proportional				
1000×10	64.6	57.0	51.0	2907
100×100	67.4	80.6	64.6	5207
10×1000	68.1	69.2	65.1	4505
Tournament				
1000×10	84.3	20.4	72.5	1479
100×100	68.9	55.8	59.8	3337
10×1000	54.4	50.1	43.8	2194

5. Conclusions

The application would be confirmed by larger studies involving comparison with the existing solutions [11], and experimental work on DNA molecule synthesis. Especially, an experiment with the use of the proposed method in a genetic laboratory is required. Realization of the calculated synthesis protocol of a known gene and an examination of the product of this realization would confirm the credibility and the usefulness of the proposed method.

References

[1] S. L. Beaucage and R. P. Iyer, "Advances in the synthesis of oligonucleotides by the phosphoramidite approach, *Tetrahedron*, vol. 48, pp. 2223–2311, 1992.

[2] M. Stryer, *Biochemistry*. Freeman, 1995.

[3] W. Stemmer *et al.*, US Patent No. 6,368,861, 2002.

[4] W. P. C. Stemmer, A. Crameri, K. D. Ha, T. M. Brennan, and H. L. Heyneker, "Single-step assembly of a gene and entire plasmid from large numbers of oligodeoxyribonucleotides", *Gene*, vol. 164, no. 1, pp. 49–53, 1995.

[5] S. Huntsman, "Towards the batch synthesis of long DNA", Tech. Rep. ADA409078, Institute for Defense Analyses, Alexandria, US, 2002.

[6] S. J. Kodumal *et al.*, "Total synthesis of long DNA sequences: synthesis of a contiguous 32-kb polyketide synthase gene cluster", in *Proc. Nat. Academy Sci. United States of America*, vol. 101, no. 44, p. 15573, 2004.

[7] R. Nowak and M. Romaniuk, "Long DNA strands synthesis optimizing", *Prace Naukowe Politechnnniki Warszawskiej, z. 165, Evolutionary Computation and Global Optimization*, 2008, pp. 157–164.

[8] G. Morawski and R. Nowak, "Porównanie algorytmów optymalizujących sekwencje DNA kodujące bialka ze szczególnym uwzględnieniem algorytmu ewolucyjnego", Tech. Rep., Warsaw University of Technology, Institute of Electronic Systems, Warsaw, 2008.

[9] "Codon usage database" [Online]. Available: http://www.kazusa.or.jp/codon/

[10] M. Zuker and P. Stiegler, "Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information", *Nucleic Acids Res.*, vol. 9, no. 1, p. 133, 1981.

[11] D. Hoover and J. Lubkowski, "DNAworks: an automated method for designing oligonucleotides for PCR-based gene synthesis", *Nucleic Acids Res.*, no. 30, 2002.

Robert Nowak received the M.Sc. (1999) and Ph.D. (2004) in Computer Science from Warsaw University of Technology. He is Assistant Professor at Artificial Intelligence Division, Institute Electronic Systems, WUT. He participated and coordinated in a number of research and commercial projects in the military, biology, medicine and energy field. His interests are artificial intelligence, computational biology, data fusion, risk management and computer software development. He is the author of 40 papers and one book.
E-mail: r.m.nowak@elka.pw.edu.pl
Faculty of Electronics and Information Technology
Warsaw University of Technology
Nowowiejska st 15/19
00-665 Warsaw, Poland

Maciej Michalak was born in 1984. He studied information technology at Warsaw High School of Economy and Information Technologies and on the Faculty of Electronics and Information Technology at Warsaw University of Technology. In 2009 he wrote his B.Sc. thesis devoted to gene synthesis. He has been continuing this topic in his M.Sc. thesis. In September 2011 he took part in the XIII National Conference of Evolutionary Algorithms and Global Optimization where he introduced some results of his work on gene synthesis optimization. Simultaneously he works at Horus s.c. where he develops business management systems for telecommunication companies.
E-mail: M.Michalak@stud.elka.pw.edu.pl
Faculty of Electronics and Information Technology
Warsaw University of Technology
Nowowiejska st 15/19
00-665 Warsaw, Poland