SYNTHESIS OF COMBINATIONAL BINARY CIRCUITS BASED ON GA WITH ENTROPY EVALUATION

Joanna Chimiak, Joanna Kolodziejczyk, Miroslaw Opoka
Technical University of Szczecin, Institute of Computer Science and Information Systems
Zolnierska 49, 70-210 Szczecin, Poland, e-mail: {jdv, pandora, mopoka}@dedal.man.szczecin.pl

Abstract

This paper addresses a problem of application of genetic algorithm (GA) to synthesize combinational circuits. The main objective of investigation was to test efficiency of GA, which is using entropy to evaluate fitness function and a case-study on using entropy as a criteria of algorithm's quality evaluation. We propose two methods of the fitness function calculation. The first one is based on functional conditional entropy and the other considers inverted truth (expected) vectors in a target function. The results presented below were obtained by testing circuits using AND, EXOR and NOT gates or any two-input binary gates.

1. Introduction

The genetic algorithms are one approaches used to solve optimizing problems (e.g. NP-hard). One of the GA applications mentioned in the overview by [Moraga94, Soldek95] is circuit design area. The GA was successfully used as a tool to minimize Boolean functions and to select the order in the Binary Decision Diagram (BDD) and the result of minimization was optimal Exclusive Sum of Product [Thoms96]. Other examples: the method was applied to find minimum number of product terms in fixed polarity Generalized Reed-Muller form [Chatt96, Mill94]. It is also useful for decomposition of multilevel logic net [Lanc97], optimizing interconnections in the network of gates [Fogar98] or PLAs with two bit decoder [Bent94]. The GA can effectively optimize multiple-valued circuits [Kalg98].

The algorithms of VLSI synthesis using GAs are known as Evolutionary Circuit Design or Evolvable Hardware. Application of evolution directly to a Field Programmable Gate Array (FPGA) device was described by [Thomp96, Harv96]. In the above-mentioned algorithm the GA changes internal configuration (given as a bit streams) of FPGA. Another application uses GA for Standard-cell Placement on the physical layout [Shah90, Shah94]. The papers [Sipper97, Iba96] include a brief survey of the Evolutionary Circuit Design algorithms.

This paper employs GA to synthesize switching circuits of 2-input gates. Two ways of fitness function interpretation are presented: (i) using the conditional functional entropy, (ii) considering inverted output vectors in the target function. The experimental results are compared through analysis of variation of the entropy's value between expected output vectors and response of the circuit coded as a chromosome. The methods of selection, crossing and parameters as population size, crossover probability and mutation, the number of selected individuals in the elitism selection have been investigated with respect to their influence on the GA efficiency. The GA evolves combinational networks employing a rectangular array of logic gates (to simplify possible realization by means of FPGAs).

2. How the GA is involved in the Circuit design problem

The distinguishing feature of the synthesis of logic circuits employing GA is viewing circuit as a black-box. The input and output signals are given as a truth table. The target function of the GA is the function to be implemented by the circuit. The topology of the circuit is coded by a chromosome and then – evolved (Fig. 1).

A regular net specifies size of a circuit where the intersection points correspond to the gates. From the set of all gates those have to be chosen, which are needed to realize a given function. The net is rectangular array with size of Nc x Nr (Nr – number of rows, Nc – number of columns). Each gate used in the net implements a two variable Boolean function. The number of inputs for the circuit is 2+2n, comprising variable and inverted variable values as well as the constants 0 and 1. Multi-output functions are allowed, too. The inputs, columns of gate matrix and outputs of the gates form levels. To eliminate feedback the inputs of a gate are only allowed to connect with outputs from lower levels. The *LB* (level back) variable limits the extent of connections. The inputs of gates from level *l* may be outputs of gates from level not lower than the *l-LB*.

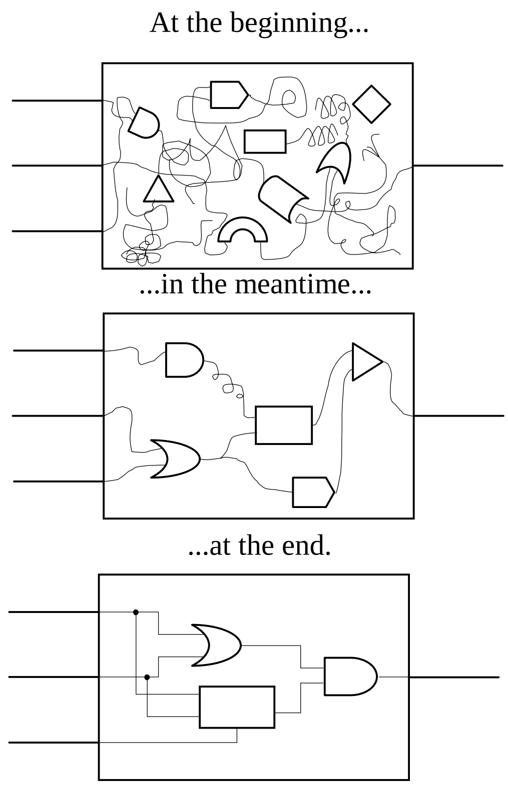

Fig. 1. The idea of using GA in the synthesize of circuit

The GA evolves the circuit by searching the spaces of gate types and input-output interconnections between them. These parameters are encoded in the chromosome which consists of Nr x Nc triples of integers (two of them are the input numbers and the third describes the gate type) and, additionally, m integers (m is the number of outputs). For example: Fig. 2 shows the coded chromosome. The list of negative numbers of gate types is the same as in the [Mill97]. The positive numbers indicate a multiplexer (MUX) with single control input.

2	3	-1	2	3	-12	6	7	2	8	1	-4	9

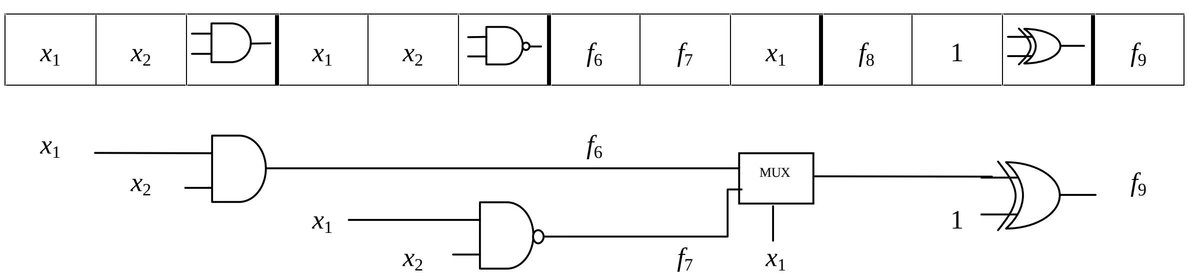

Fig. 2. The simple circuit coded in the chromosome

The GA operations: crossover, mutation and selection improve circuit randomly selected at the first step of algorithm. The realization of Boolean function from the truth table by the circuit is the stop condition of the algorithm.

3. Entropy evaluation

The method employs conditional functional entropy in two ways: it is used to evaluate the fitness function inside of the GA and also to compare the efficiency of the examined algorithms. The fitness function is calculated as conditional entropy of a current output vector and the target output vector. This information measure allows estimation of the process' convergence, which means that changes of the entropy indicate changes of the vectors' independence while approaching the circuit to realize the proper function. The entropy is calculated for each chromosome in the population.

Definition 1. **[**Simov97**]** For given variable x_i, the *conditional entropy* $H(f|x_i)$ of logic function f is a measure of average uncertainty of logic function values with respect to the variable x_i:

$$H(f|x_i) = -\sum_{\beta=0}^{1}\sum_{t_i=0}^{1} p_{\left| \substack{f=\beta \\ x_i=t_i} \right.} \log_2 \frac{p_{\left| \substack{f=\beta \\ x_i=t_i} \right.}}{p_{|x_i=t_i}} \tag{3.1}$$

where $p_{\left| \substack{f=\beta \\ x_i=t_i} \right.}$ denotes the probability that $x_i = t_i$ and $f = \beta$.

Definition 2. For given variable x_i, the *conditional functional entropy* $h(f|x_i)$ of logic function f is calculated as follows:

$$h(f|x_i) = -\sum_{\beta=0}^{1}\sum_{t_i=0}^{1} r(\beta,t_i)\log_2 \frac{r(\beta,t_i)}{r(t_i)} \tag{3.2}$$

where $r(\beta,t_i)$ denotes the number of combinations of variable x_i taking value t_i and logic function f taking value β, and $r(t_i)$ denotes the number of combinations of variable x_i taking value t_i**.** Formula (3.2) has been yielded from the dependence of *functional entropy* [Simov97] on Shannon entropy.

Example. Assume that a function of four variables is given by its truth vector [1111001100000010]. Conditional entropy (3.1) of a function f given the variable x_0:

$$H(f|x_0) = -\left(2/16\log_2\frac{2/16}{8/16} + 7/16\log_2\frac{7/16}{8/16} + 6/16\log_2\frac{6/16}{8/16} + 1/16\log_2\frac{1/16}{8/16}\right) \approx 0{,}677$$

Conditional functional entropy (3.2) of a function f given the variable x_0:

$$h(f|x_0) = -\left(2\log_2\frac{2}{8} + 7\log_2\frac{7}{8} + 6\log_2\frac{6}{8} + 1\log_2\frac{1}{8}\right) \approx 10{,}839$$

For all function's variables we obtain the conditional entropy values as below:

x_i:	x_0	x_1	x_2	x_3
$h(f\|x_i)$	10.839	15.635	14.126	15.635
$H(f\|x_i)$	0.677	0.977	0.883	0.977

In the algorithm we used the conditional functional entropy because of its lower computation cost. The interpretation of entropy values calculated according to formulas (3.1) and (3.2) is the same.

4. The GA's characteristics

The authors propose new algorithm of the fitness function calculation. The goal of GA is to find circuit realizing the Boolean function given as a truth table.

The fitness function represents the percentage of sameness of the truth vectors and the current output vector obtained at every step of the GA. If the vectors are equal (100 %

matching) the fitness function is equal to 1 and the circuit represented by the corresponding chromosome is the solution [Mill97].

4.1. The fitness functions based on the conditional functional entropy (GENT)

In this paper it is proposed for the first time to compute the conditional functional entropy between the truth vector of the required function v_t=[v_{t1},v_{t2},...,v_{tm}], (where m is the number of specified values) and the current output vector (the response of the circuit described by a chromosome to the input pattern) v_c=[v_{c1},v_{c2},...,v_{cm}]. The entropy is minimal (0) when the vectors are equal ($v_t = v_c$) or each of them is the complement of the other ($v_t = \sim v_c$).

For instance, the conditional functional entropy between the vector v_t=[10101110] and the current output vector (the best in current population) v_c=[01110111] is equal to $h(y|c)$=6.

If, for instance, v_t =[10101110] and v_c=[01010001] then $h(y|c)$=0. So, the vector v_c is the (complemented) solution.

From now on, the algorithm using the method described above will be called GENT.

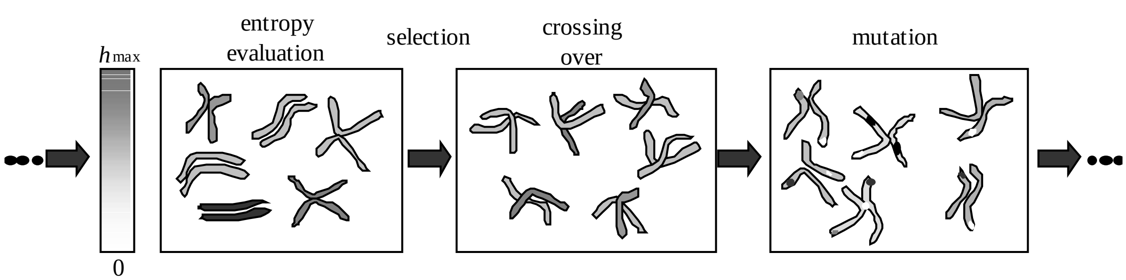

Fig. 3. The idea of GA based on entropy calculation.

4.2. The target function taking into account the vector which is a complementation of the desired (truth table) vector (GINV)

We propose to calculate the fitness function as the percentage of coincidence of the current output vector described by a chromosome and truth vector or its complementation. In other words, the circuit whose output values are inverted desired values is a solution too. In this case, a NOT gate has to be added at the circuit's output.

From now on the algorithm using the method described above will be called GINV.

5. Experiments and results

The goal of the research work was:

(i) to compare the algorithms [Mill97], GENT, GINV and their use to create circuits built from either AND, EXOR gates or all types of 2-input gates,

(ii) to test methods of selection (elitist and non elitist) and crossover (1point, 2points and uniform),

(iii) to investigate the influence of the parameters (the size of population, the probability of crossover and mutation, the number of copied individuals in the elitist selection) on the efficiency of the GA,

(iv) to analyze the quality of optimization of every algorithm, the influence of the level back variable (*LB*) and size of gate matrix on number and type of gates in the evolved solution.

The algorithm has been implemented in C++, compiled with full optimization using gcc 2.7.2.3 and tested on a Pentium II@350 Mhz with 512 MB of RAM., running Debian GNU/Linux 2.1 and kernel 2.2.4.

5.1 The comparison of the algorithms using entropy criteria

First, we compare (Table 1) the original GA proposed in [Mill97] and two modifications of the proposed GA – GENT and GINV. The entropy calculated for truth vectors and current output vectors gives an information about how much the output vector has converged to the desired vector. Lower entropy means better mutual coincidence. The average entropy is the off-line value, i.e. it is the average from the best entropies obtained from all iterations.

All algorithms were examined for the benchmark add0* (8 inputs, 1 output).

Table 1. The efficiency of the original GA [Mill97], the entropy based algorithm GENT and the one considering inverted truth vector GINV (the number of generations was 5000 in 50 runs, the number of populations was 70)

	GENT			GINV			[Mill97]		
	A 4x3	**A-E 4x3**	**A-E 5x5**	**A 4x3**	**A-E 4x3**	**A-E 5x5**	**A 4x3**	**A-E 4x3**	**A-E 5x5**
Exact solutions	36%	6%	8%	36%	6%	6%	30%	2%	0%
Mean Entropy	0.216	0.426	0.452	0.234	0.384	0.455	0.266	0.611	0.680
Time (min)	15:20	19:30	34:45	15:40	19:17	34:47	15:21	19:59	36:13
Time of FES(min)**	00:51	06:30	08:41	00:52	06:26	11:36	01:01	19:59	0
Min. number of gates	4	4	4	4	4	4	4	6	0
Mean amount of gates	6.500	4.333	6.000	6.944	5.333	6.667	6.467	6.000	0

A 4x3- circuits include all types of gates with the size of gate matrix 4x3, A-E 4x3 circuits include AND, EXOR and NOT types of gates with the size of gate matrix 4x3, A-E 5x5 circuits include AND, EXOR and NOT types of gates with the size of gate matrix 5x5

*** Mean time of searching for the exact solution.*

Analysis of the entropy, the percentage of exact solutions and mean time of searching for exact solutions show that both GENT and GINV perform similarly (Table 1, Fig. 4), because their target functions are identical.

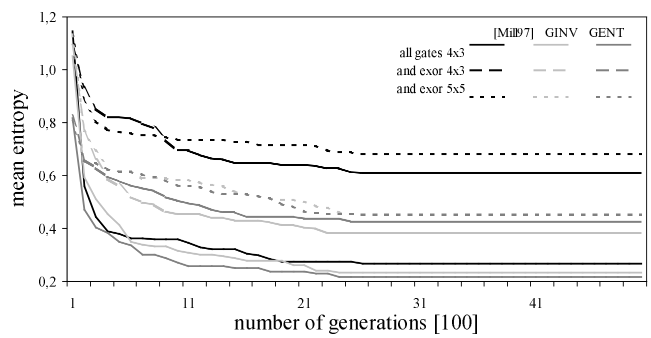

Fig. 4. Relation between mean entropy and number of generation

The target function does not take the optimizing goal into account - the aim is to implement function. The authors analyze mean amount of gates in the exact solution. The result was that the method based on entropy fitness function calculation found minimal number of gates more frequently than other algorithms.

The circuits built from AND, EXOR and NOT gates are generally less effective, mostly due to the fact that MUX type gates are not present. All-gates type circuits consist of MUX gates in roughly 50%.

5.1 Methods of selection, crossover and probability of mutation, probability of crossover, size of population and their influence on efficiency of GA

There are many operators to use in GAs. Specifically, the algorithms were tested for elitist and non-elitist selection, one point, two points and uniform crossover. Conducted research shows that it is highly unlikely to find an exact solution in less than 5000 generations

* 10% of specified values (source: directories of Portland State University)

without using elitist selection. It could be speculated that choosing more then one of the bests individuals for further reproduction should lead to better results, but testing indicates that it is not the case. The number of the copied individuals being equal to 1 gave the minimal entropy and the best percentage of solutions. The research shows that increasing of the parameter is ineffective. However the difference in efficiency of tested crossover methods was negligible (*Table 2.*).

Table 2. Influence of type of crossover on efficiency of the GA [Mill97], GENT, GINV

	GENT			**GINV**			**[Mill97]**		
Type of crossover	**1-p.***	**2-p.***	**uf***	**1-p.***	**2-p.***	**uf***	**1-p.***	**2-p.***	**uf***
Exact Solution	86%	78%	84%	74%	86%	68%	78%	86%	84%
Mean Entropy	0.05	0.07	0.05	0.09	0.05	0.11	0.07	0.05	0.05
Time of FES(min)**	00:53	00:54	01:00	01:08	00:51	01:24	00:50	00:38	00:46
Mean amount of gates	8.53	8.59	9.24	9.08	8.77	8.29	8.44	8.63	8.33

**1-p. – one point, 2-p. – two points, uf – uniform; ** Mean time of searching for the exact solution.*

For values of mutation probability ranging from 1 to the inverse of chromosome length the algorithms effectiveness is highest.

As the value of probability grows, the number of exact solutions decreases. When it exceeds 10%, the algorithm no longer finds solutions in reasonable number of generations.

For probabilities of crossover near ½ the solutions are optimal (minimal entropy and maximal percentage of exact solutions).

For population size ranging from 70 to 90 the value of mean entropy is minimal in all tested algorithms.

Observe that curves depicting relation of mean entropy and percentage of exact solutions to GA parameters (Fig. 5) are symmetrical. The meaning of this is that is possible to replace quality assessment based on the percentage of exact solutions with functional conditional entropy. Entropy is a new proposed measure of the algorithms' comparison.

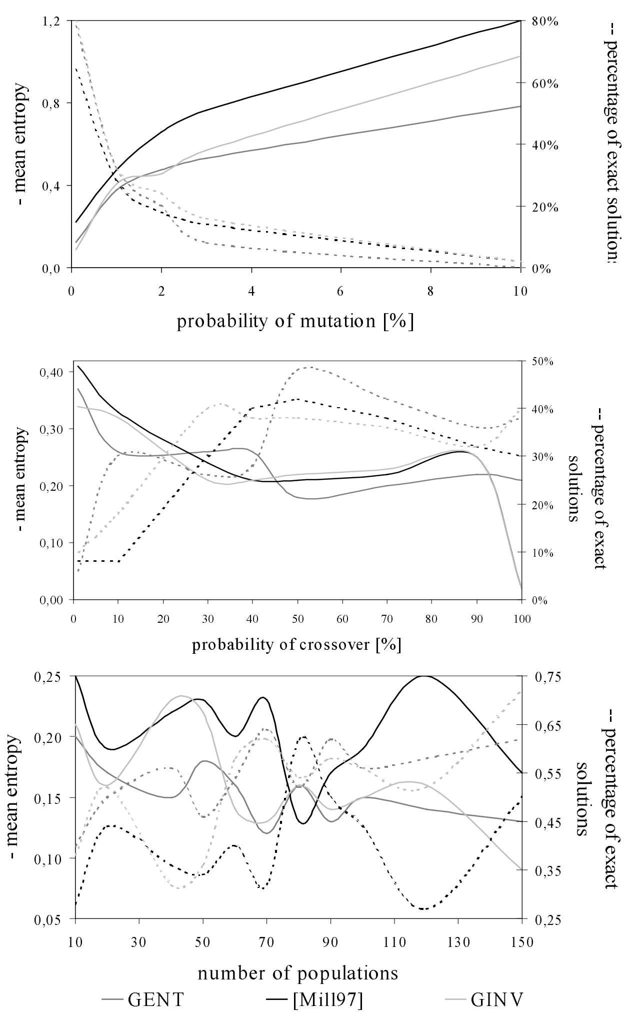

Fig. 5. Relationship between probability of mutation, probability of crossover, size of population and mean entropy, percentage of exact solutions

5.3 The analysis of size of gate matrix, depth of interconnection variable (LB) and the average number of gates in the exact solution

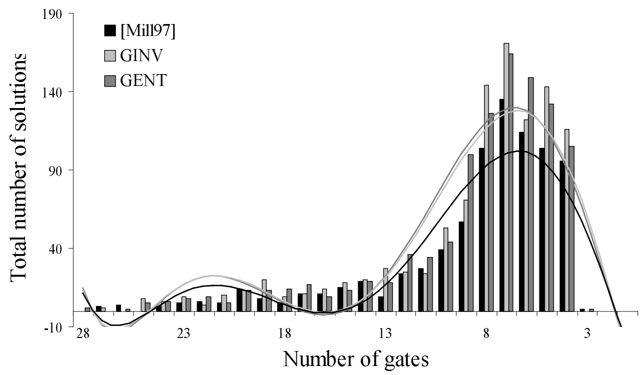

Fig. 6. Histogram of number of gates in exact solutions

The frequentness of occurrence of the specified number of gates in each of exact solutions was analyzed. The graph showed on Fig. 6 was obtained by compiling all results from all tests. We discovered that all examined algorithms found the circuit built from 5, 6, 7 gates most often. The GENT and [Mill97] algorithms found the minimum circuit which was built from 3 gates (realizing benchmark add0, 10% of specified values, when the size of matrix was 4x3, number of generations equal to 5000 and 50 runs). The trend curves achieve maximum for circuit with 5 to 7 gates. The maximum value is highest in GENT. Consequently, the algorithm found the most optimal solutions regardless of GA parameters.

The size of gate matrix and the LB value have been tested, too. More freedom in circuit, i.e. when the number of gates is higher than the number of inputs and specified output values gives more efficiency (number of exact solution) but optimization is worse. However, for large number of inputs, using large size of the array increases the chromosome length and time consumption.

Increase in the number of generations is the reason of increased efficiency of GA (Table 3). It has no influence on mean amount of gates in solution.

Table 3. Influence of type of crossover on efficiency of the GA [Mill97], GENT, GINV.

	GENT				GINV				[Mill97]			
Number of generation	**100**	**1000**	**3000**	**5000**	**100**	**1000**	**3000**	**5000**	**100**	**1000**	**3000**	**5000**
Exact Solution	2%	16%	30%	44%	0%	14%	24%	36%	0%	12%	32%	44%
Mean Entropy	0.44	0.31	0.24	0.20	0.62	0.34	0.28	0.23	0.51	0.34	0.26	0.20
Time of FES(min)**	00:37	00:42	00:58	01:01	---	00:48	01:16	01:20	---	00:56	00:55	01:01
Mean amount of gates	6.00	6.25	5.60	6.59	---	6.00	6.33	6.56	---	5.17	6.00	6.32

*** Mean time of searching for the exact solution.*

6. Conclusions

We propose a new approach to evaluate the fitness function as well as target function for the GA applied to synthesis of circuits of two-input gates. The main results obtained were:

- First, a new algorithm to find the fitness function has been developed, which opened the direction of research to the bipolar convergence; it allows to define the target function taking into account also the complemented output vectors and improve the GA efficiency.
- Second, the method of assessment in GA using conditional functional entropy for the truth vector of the required function and the current output vector is presented. The entropy describes the process of fitting the vectors during the iterations and is equivalent to a measure of the quality characteristics (efficiency).
- Third, analysis of the influence of the proposed parameters on the GA efficiency has been conducted.

We can conclude that the proposed GA is efficient for functions with small number of inputs and specified values. For larger functions the size of a chromosome and computing time sharply rise. In further research we plan to use decomposition and a different way to code a chromosome.

Acknowledgements

We are thankful Prof. S. Yanushkevich and Prof. V. Shmerko for support and M. Smyk for moral support.

References

[Bent94] Benten M. S. T., Sait S. M., GAP: a Genetic Algorithm to Otimize Two-Bit decoder PLAs, *Int. J. Electronics*, Vol. 76, No. 1, 1994, pp. 99-106,

[Chatt96] Chattopadhyay S., Roy S., Chudhuri P. Pal, Synthesis of Highly Testable Fixed-Polarity AND-EXOR Canonical Networks-A Genetic Algorithm-Based Approach, *IEEE Transactions on Computers*, vol. 45, No. 4, 1996, pp. 487-490

[Fogar98] Fogarty T. C., Miller J. F., Thomson P., Evolving Digital Logic Circuits on Xilinx 6000 Family FPGAs, *Soft Computingin Engineering Design and Manufacturing*, Springer-Verlag, London 1998, pp. 299-305

[Gold87] Goldberg E., *Genetic Algorithm In Search, Optimisation and Machine Learning*, Addison-Wesley, 1987

[Harv96] Harvey A., Thompson A., Through the Labyrinth Evolution Finds a Way: A Silicon Ridge, *Proc. of The 1st Int. Conf. on Evolvable Systems: from Biology to Hardware(ICES96)*, 1996, Springer-Verlag, pp. 406-422

[Iba96] Iba H., Iwata M., Higuchi T., Machine Learning Approach to Gate Level Evolvable Hardware, *Proc. of The 1st Int. Conf. on Evolvable Systems: from Biology to Hardware(ICES96)*, 1996, Springer-Verlag, pp. 327-343

[Kalg98] Kalganova T., Miller J. F., Lipnitskaya N., Multiple-Valued Combinational Circuits Synthesised Using Evolvable Hardware Approach, *7th Int. Workshop on Post-Binary ULSI Systems*, 1998, Japan, pp. 52-54

[Lanc97] Lanchares J., Hidalgo J. I., Sanchez J. M., Boolean Networks Decomposition Using Genetic Algorithm, *Microelectronics Journal 28*, 1997, pp. 551-560

[Mill94] Miller J. F. , Luchian H., Brandbeer P. V. G., Barclay P. J., Using a Genetic Algorithm for Optimizing for Fixed Polarity Reed-Muller Expansion of Boolean Functions, *Int. J. Electronics*, 1994, vol. 76, no. 4, pp. 601-609

[Mill97] Miller J. F., Thomson P., Fogarty T., Designing Electronic Circuits Using Evolutionary Algorithms. Aritmetic Circuits: A Case Study, *Genetic Algorithms in Engineering and Computer Science*, 1997 John Wiley & Sons Ltd., pp. 105-131

[Moraga94] Moraga C., Spectral Techniques and Soft Computing, *Proc.5th Int. Workshop on Spectral Techniques*, China, 1994, pp. 1-7

[Shah94] Shahookar K., Khamisani W., Mazumder P., Reddy S. M., Genetic Beam Search for Gate Matrix Layout, *IEE-Proc.- Comput. Digital Technology*, Vol. 141, No.2 1994, pp.123-128

[Shah90] Shahookar K., Mazumder P., A Genetic Approach to Standard Cell Placement Using Meta-Genetic Parameter Otimization, *IEEE Transactions on Computer-Aided Design*, Vol. 9, No. 5, 1990, pp. 500 -511

[Simov97] Simovici D., Shmerko V., Cheushev V., Yanushkevich S., Information Estimation for Logic Function, *Proc. of the Fourth Int. Conf. Applications of Computer Systems*, Poland 1997, pp. 287-301

[Sipper97] Sipper M., Sanchez E., Mange D., Tomassini M., Perez-Uribe A., Stauffer A., A Phylogenic, Ontogenic, and Epigenic View of Bio-Inspired Hardware Systems, *IEEE T. Evolutionary Computation*, 1997, pp.1-14

[Soldek95] Soldek J., Yanushkevich S., Genetic Algorithms in Logic Design, *Proc. of the Int. Conf on Computer-Aided Design of Discrete Device*, Belarus, 1995 , Vol 2., pp.17-25

[Thomp96] Thompson A., An Evolved Circuit, Intrinsic in Silicon, Entwined with Physics, *Proc. of The 1st Int. Conf. on Evolvable Systems: from Biology to Hardware (ICES96)*, 1996, Springer-Verlag, pp. 390-405

[Thoms96] Thomson P., Miller J. F., Symbolic Methods for Simplifying AND-EXOR Representations of Boolean Functions Using a Binary-Decision Technique and a Genetic Algorithm, *IEE Proc.-Comput. Digit. Tech.*, Vol. 143, No. 2, 1996, pp. 151-155