Kalganova T. and I. Baradavka (2002) A probabilistic approach to analyse the evolutionary process in circuit design. Proc of the 9th LMPU Conference, France, pp. 689-696.

A probabilistic approach to analyse the evolutionary process in circuit design

Tatiana Kalganova
Brunel University, Uxbridge, UK
tatiana.kalganova@brunel.ac.uk

Igor Baradavka
Brunel University, Uxbridge, UK
igor.baradavka@brunel.ac.uk

Abstract

One of the actual problems in the evolvable hardware is the evolvability of logic circuits. In order to understand better the nature of existing problem, the probabilistic analysis can be used. This paper aims to investigate how the circuit layout evolution is carried out. This is interesting thing to do for two main reasons. Firstly, to investigate what type of genes mostly influence on the algorithm performance in evolvable hardware. Secondly, to see how effective an allocation of active logic gates might be in a digital circuit design task. In order to achieve this goal we investigate the genotypes of the best chromosomes which bring some improvements in evolutionary process. The logic circuits have been evolved using circuit layout evolution. [1]

Keywords: Evolutionary computation, Evolvable hardware.

1 Introduction

Evolvable Hardware approach is a recently developed technique to synthesise the electronic circuits using evolutionary algorithms [1], [2], [3], [4]. A central idea of this approach is to represent each possible electronic circuit as chromosome in an evolutionary process in which the standard genetic operators such as initialisation, recombination, selection are carried out. Currently, one of the main problem in this area is the evolvability of logic circuits. In order to overcome this problem a number of approaches has been proposed: (1) Divide-and-Conquer approach [5]; (2) Bi-directional incremental evolution [6]. Although, the circuit evolvability problem has been solved using additional techniques, the nature of this problem remains uninvestigated. Therefore, there is a demand to investigate the evolutionary process of logic circuits in depth. This paper attempts to have a look at the evolutionary design of logic circuits from the probabilistic point of view. This approach is applied to design of combinational logic circuits using extrinsic Evolvable Hardware (EHW).

The EHW generates logic circuits using circuit layout evolution that has been discussed in details in [7]. It has been shown that the Genetic Algorithm (GA) performance strongly depends on the set of logic gates used to produce the 100% functionally circuits. In [8] experiments were reported which revealed the dependence the GA performance with gate array dimensions and the degree of internal connectivity. Analysis of the evolvable hardware approach for both binary and multiple-valued functions shows us that the GA performance strongly depends on the number of rows and columns and the internal connectivity [9], [8]. In subsequent discussion we define the *circuit geometry* or *circuit layout* to mean the layout of the rectangu-

[1] This work is partially submitted by Nuffield foundation and BRIEF award (Brunel University).

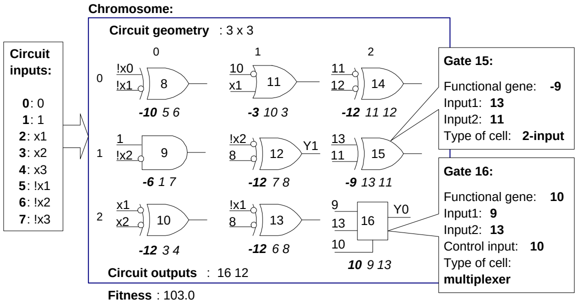


Figure 1: An example of the phenotype and corresponding genotype of a chromosome with 3x3 circuit layout

lar array of logic cells. It is characterised by just two numbers: the number of rows and columns in the cellular array. The degree of connectivity in the circuit called *levels-back* defines how many columns of cells to the left of current column can have their outputs connected to the inputs of the current cell, this also applies to the final circuit outputs.

The paper presents the probabilistic analysis of the method discussed above. A proposed probabilistic analysis approach allows us to define how different type of genes in chromosome representation and their location influence on GA performance. For this purpose a differential chromosome will be introduced. The differential chromosome shows the difference between genotypes of two ordinary chromosomes presented the circuit structure.

2 Circuit layout evolution

2.1 Chromosome encoding

Let us consider the chromosome representation with the actual circuit structure using an example that is given in Fig. 1. Let us examine a possible circuit representing a full adder. This function has 3-inputs and 2-outputs and is implemented here on a combinational network with 3x3 circuit geometry ($N_{cols} \times N_{rows}$). The circuit inputs are labelled as follows: 0 and 1, which represent the logical constants 0 and 1 respectively, labels 2, 3 and 4 correspond to the input variables x_0, x_1 and x_2 respectively. The inverted inputs $\overline{x_0}$, $\overline{x_1}$, $\overline{x_2}$ are represented as 5, 6 and 7. In this example the cell type gene (shown in bold) represents one of the 13 possible gates (AND, OR, EXOR with primary and inverted inputs or multiplexer). The cell type gene may be positive or negative integer. If positive then the function is a multiplexer and the integer represents the control connection. If cell type gene is negative, we use an encoding table to define the type of gate.

The chromosome is represented by a 3-level structure: (1) Layout; (2) Circuit and (3) Gate (cell) structures. At the first level, the global characteristics of the circuit are defined. These are the connectivity parameter and the number of rows and columns. The circuit geometry can be changed at this level. At the second level, the array of cells are created and the circuit outputs are determined. Finally, the third level represents the structure of each cell in the circuit. This data consists of the number of inputs, the input connections and the cell type gene.

The output of each cell is assigned an individual address. Thus the output of logic cell in the 2^{nd} column and 2^{nd} row is labelled as 16. The number of circuit outputs is defined by the number of outputs in the logic function implemented. The logic cell label determines each of these outputs.

2.2 Dynamic Fitness Function

The evaluation process consists of the two main steps. First we are trying to find circuits with 100% functionality (F_1 criteria) and second we are trying to minimise the number of active gates in 100% functional circuits (F_2 criteria) as well as minimise the redundancy of logic gates in the circuit (F_3 criteria). Fitness function F_3 minimises the layout in the circuit. The active gate is the gate, which is proved to be not redundant. We use two evolutionary strategies: (1) F_1 strategy and (2) $F_1 + F_2 + F_3$ strategy.

In the first strategy, the chromosome is evaluated using F_1 criteria only and once the 100% functional circuit evolved, the evolutionary process is terminated. In the case of $F_1 + F_2 + F_3$ strategy, F_2 and F_3 criteria are activated as soon as $F_1 = 100.0$ and the number of inactive gates in circuit is estimated. When heterogeneous circuit geometry is employed, F_2 is calculated based on the maximum available circuit layout. For example, let the maximum circuit layout is 10×10. If the fully functional circuit has 3×8 circuit layout and contains 18 primitive active logic gates ($F'_2 = 18$), then $F_2 = \overline{F'_2} = 10 \times 10 - 18 = 82$.

Redundancy can be defined as number of redundant logic gates divided by total number of logic gates.

$$F'_3 = \frac{3 \times 8 - 18}{3 \times 8} = 0.25;$$
$$F_3 = \overline{F'_3} = 1 - F'_3 = 0.75.$$

3 Differential chromosome

The difference between genes in chromosomes $\mathbb{C}_1$ and $\mathbb{C}_2$ can be defined using the *differential chromosome*, $\mathbb{D}_{\mathbb{C}_1\mathbb{C}_2}$. This chromosome contains 3-level structure as well as the ordinary one. The genes in the differential chromosome are calculated based on the difference in genes of $\mathbb{C}_1$ and $\mathbb{C}_2$ at the circuit and gate levels. The circuit level contains the rectangular array of logic gates and outputs. The circuit output gene represents the number of circuit genes which are different in $\mathbb{C}_1$ and $\mathbb{C}_2$. The gate level contains functionality and connectivity genes, which define the differential between the corresponding genes in $\mathbb{C}_1$ and $\mathbb{C}_2$. It is necessary to note that in chromosome interpretation the functional set represents any primitive logic gates AND, OR, EXOR, NOT with inverted and primary inputs. In order to receive more accurate analysis of circuit evolution, the "two-gene" interpretation of gate functionality has been used: $< d_{gt}\ d_{it} >$, where d_{gt} is the primitive gate type, $d_{gt} \in \{$AND, OR, NOT, EXOR$\}$ and the d_{it} defines the number of different inputs (primitive or inverted) used in logic cell. Thus each gate in $\mathbb{D}_{\mathbb{C}_1\mathbb{C}_2}$ can be described by three genes: $< d_{gt}\ d_{it}\ i_d >$, where d_{gt} defines if the primitive type of gate is the same or not for chromosomes $\mathbb{C}_1$ and $\mathbb{C}_2$; d_{it} determines the number of different gate inputs and i_d is the number of different uncommitted connections. We will refer to uncommitted connection if the logic function described the behaviour of logic cell does not depends on this connection. Note that if all genes of differential chromosome are 0, the genotypes of $\mathbb{C}_1$ and $\mathbb{C}_2$ are the same. The fitness function of differential chromosome is defined as follows: $F_d = F_{\mathbb{D}_{\mathbb{C}_1\mathbb{C}_2}} = F_{\mathbb{C}_1} - F_{\mathbb{C}_2}$. Note that if functionality $F_1 = 100$ for both chromosomes, then F_{1d} is to be also 100.0.

4 Probabilistic analysis

The probabilistic analysis is based on the analysis of differential chromosome genotypes. The differential chromosomes analysed have been calculated based on ordinary chromosomes involved in successful evolution. The issue of this work is to define how genes influence on evolutionary process. In order to do so the differential chromosomes with $F \neq 0$ have been considered.

The following notations have been adopted in order to explain the analysis process. Let define the outcomes of experiment ξ be the genotype and phenotype of differential chromosomes. The sample space Ω associated with an experiment ξ is the collection of all possible phenotypes and genotypes of differential chromosome of ξ. The intersection of events E_1 and E_2, written as $E_1 \cdot E_2$, is defined as the set of outcomes which belong to both E_1 and E_2. Given two subsets of Ω, say E_1, E_2, the union of E_1, E_2, written as $E_1 \cup E_2$, is defined as the set of outcomes which belong to either E_1 or E_2 or both. Let N_{dc} be the number of differential chromosomes generated using procedure described above. The circuit evolutionary process γ contains two evolution sub-processes: (1) Evolution of functional circuit, γ_{F_1}; (2) Improving the functional circuit evolved, $\gamma_{F_2+F_3}$. Note that these two processes can not be performed at the same time. The first process is carried out when the functionality of the best chromosomes is less then 100% and the second process is performed when the GA tends to improve the fully functional solution. There are 5 different types of genes which could influence on

GA performance: (1) cell type gene; (2) input cell type gene; (3) connection gene; (4) circuit output gene; (5) circuit layout gene. So, we can define the following events which could be associated with an experiment ξ:

- E_0 the fitness function of differential chromosome is greater that 0, $F_d > 0$;
- E_1 the functionality fitness function F_{1d} is less than 100%, $F_{1d} < 100.0$ (i.e. the functionality evolution γ_{F_1} is in question);
- E_2^j the cell type gene d_{gt} located in j-th cell is greater than 0;
- E_3^j the connection gene i_d located in j-th cell is greater than 0;
- E_4^k the circuit output gene o_d located in k-th circuit output is greater than 0;
- $E_5^{1,2}$ the circuit layout gene defined the number of columns or rows is greater than 0.

The event E_0 defines that the differential chromosome is calculated using two chromosomes with different fitness functions. Event E_1 shows that the compared chromosomes are not fully functional. It means that the compared chromosomes have been involved in evolutionary process γ_{F_1}. Note that $\overline{E_1}$ defines that the functionality fitness function F_{1d} is greater or equal to 100.0. So, the event $\overline{E_1}$ agrees with the case when the differential chromosome compare the functional circuits. In other words the compared chromosome have been participated in evolutionary process $\gamma_{F_2+F_3}$. The rest events E_2^j, E_3^j, E_4^k define how the compared chromosomes are different.

The probabilities of the events $E_1 \cdot E_0$ and $\overline{E_1} \cdot E_0$ can be defined as

$$p(E_1 \cdot E_0) = \frac{N_{E_0E_1}}{N_{gen} * N_{runs}}; \qquad (1)$$
$$p(\overline{E_1} \cdot E_0) = \frac{N_{E_0\overline{E_1}}}{N_{gen} * N_{runs}};$$

where $N_{E_0E_1}$ is the number of differential chromosomes with $F_d \neq 0$ and $F_{1d} \neq 100.0$ (i.e. defines the execution of process γ_{F_1}); $N_{E_0\overline{E_1}}$ is the number of differential chromosomes with $F_d \neq 0$ and $F_{1d} = 100.0$ (i.e. defines the execution of process $\gamma_{F_2+F_3}$). The conditional probability of E_2^j, given that E_1 has occurred, is defined as

$$p(E_2^j|(E_0 \cdot E_1)) = \frac{p(E_0 \cdot E_1 \cdot E_2^j)}{p(E_0 \cdot E_1)} = \frac{{}_jN^{gt}_{E_0E_1}}{N_{E_0E_1}} \qquad (2)$$

where ${}_jN^{gt}_{E_0E_1}$ is the number of times E_0, E_1 and E_2 occurred (i.e. the number of non-zero cell type genes d^j_{gt} in differential chromosomes with $F_{1d} < 100$ and $F_d \neq 0$). The conditional probabilities of E_3^j and E_4^k, given that $E_0 \cdot E_1$ has occurred, are calculated analogously to Eq. 2:

$$p(E_3^j|(E_0 \cdot E_1)) = \frac{{}_jN^{c}_{E_0E_1}}{N_{E_0E_1}}; \qquad (3)$$
$$p(E_4^k|(E_0 \cdot E_1)) = \frac{{}_kN^{o}_{E_0E_1}}{N_{E_0E_1}};$$
$$p(E_5^{1,2}|(E_0 \cdot E_1)) = \frac{N^{cl}_{E_0E_1}}{N_{E_0E_1}};$$

where ${}_jN^{it}_{E_0E_1}$ and ${}_jN^{c}_{E_0E_1}$ are the number of non-zero input cell and connection genes located in cell j in differential chromosomes with functionality fitness function $F_{1d} < 100$ and fitness function $F_d \neq 0$ respectively; ${}_kN^{o}_{E_0E_1}$ is the number of non-zero circuit output genes located in k position in differential chromosomes with $F_{1d} < 100$ and $F_d \neq 0$; $N^{cl}_{E_0E_1}$ is the number of non-zero circuit layout genes in differential chromosomes with $F_{1d} < 100$ and fitness function $F_d \neq 0$. The conditional probabilities calculated in Eq. 2 and Eq. 3 correspond to the evolutionary process γ_{F_1} such that the functionality of circuit is evolved.

The conditional probabilities shown below correspond to the evolutionary process $\gamma_{F_2+F_3}$ which forces to improve the functional circuit in terms of the number of active gates used and the circuit layout.

$$p(E_2^j|(E_0 \cdot \overline{E_1})) = \frac{{}_jN^{gt}_{F_1}}{N_{E_0\overline{E_1}}}; \qquad (4)$$
$$p(E_3^j|(E_0 \cdot \overline{E_1})) = \frac{{}_jN^{c}_{F_1}}{N_{E_0\overline{E_1}} * N^{max}_{in}};$$
$$p(E_4^k|(E_0 \cdot \overline{E_1})) = \frac{{}_kN^{o}_{F_1}}{N_{E_0\overline{E_1}} * N_{out}};$$
$$p(E_5^{1,2}|(E_0 \cdot \overline{E_1})) = \frac{N^{cl}_{F_1}}{2N_{E_0\overline{E_1}}}.$$

The conditional probabilities calculated above define the probability with the genes influence positively on evolutionary process. In other words, these genes belong to chromosome that have just changed fitness value and the whole evolutionary process has been successful.

5 Experimental results

In this section we will consider some experimental results obtained for two-bit multiplier (mult2.pla) and two-bit adder with carry (add2c.pla). The main idea of these experiments is to define how diverse types of genes located differently influence on successful GA performance.

The initial data for the experiment are given in Table 1. Any type of genes in chromosome genotype allowed to be changed with constant gene mutation probability. The functional set of logic gates contains {AND, OR, EXOR, NOT}. The initial population is initialised randomly.

Table 1: Initial data, where ♯ is "the number of ..."

Circuit	mult2	add2c
max ♯ columns, N_{cols}	10	15
max ♯ rows, N_{rows}	10	15
Levels back, $N_{connect}$	10	15
Population size	5	5
♯ generations, N_{gen}	20000	25000
♯ GA runs, N_{runs}	1000	1000
Cell mutation rate, p_m	5%	5%
Circuit layout mutation rate, p_{cl}	10%	10%

In order to define how the different types of genes and their location influence on GA performance, the differential chromosomes have been generated and the conditional probabilities have been calculated according Eq. 3 and Eq. 4 as follows:

1. Consider the history of the best chromosomes if the final functional solution has been evolved during GA performance;

2. Select $\mathbb{C}_{t_1}$ created at generation t_1 such that the fitness of the best chromosome in question has been changed in comparison with previous one;

3. Chose $\mathbb{C}_{t_2}$ produced at generation t_2 such that the fitness of the best chromosome in question has been increased in comparison with chromosome $\mathbb{C}_{t_1}$, $t_2 > t_1$, $F_{t_1} < F_{t_2}$ and there is no improvement in terms of fitness function between generations t_1 and t_2;

4. Generate the differential chromosome $\mathbb{D}_{\mathbb{C}_{t_1}\mathbb{C}_{t_2}}$.

The selection procedure mentioned above provides that the differential chromosomes have been only generated from the chromosomes with improved fitness.

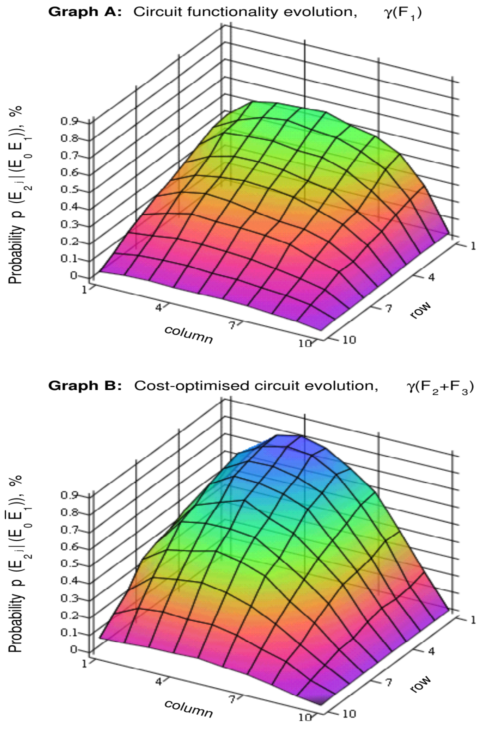
The conditional probabilities have been calculated for the following type of genes: (1) cell type gene; (2) connection genes; (3) circuit layout genes.

There are two main experiments that have been carried out. First, the conditional probabilities have been calculated for *all* logic gates in the chromosome. Second, the conditional probabilities have been calculated for the active logic gates in the circuit. At the same time, the conditional probabilities have been calculated separately for the following processes: (1) evolution of fully functional logic circuit, γ_{F_1}; (2) evolution of optimal logic circuit, $\gamma_{F_2+F_3}$.

Analysis of experimental results show that the obtained results are similar for both two-bit multiplier and two-bit adder. Therefore, in this paper we will consider in details the results obtained for two-bit multiplier only.

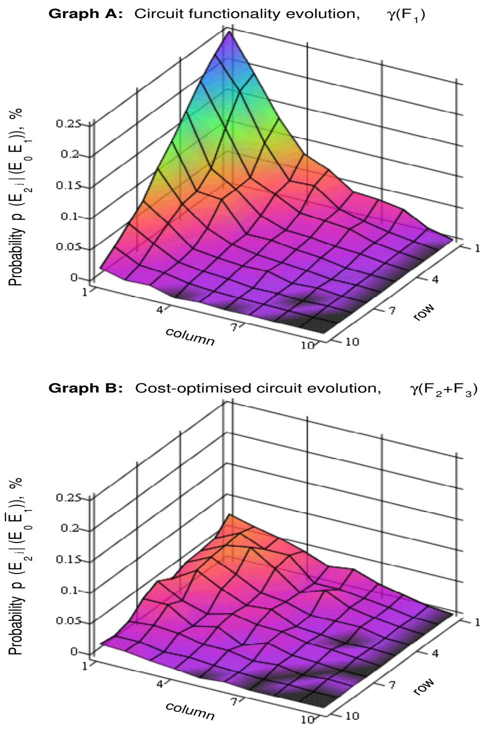
5.1 Behaviour of cell type genes

Graphs A and B (Fig.2) show the functional dependance of evolved circuit layout and the conditional probability calculated for evolutionary processes γ_{F_1} and $\gamma_{F_2+F_3}$ respectively. The results have been obtained for both active and redundant logic gates. It can be seen clearly that the cell type genes influence more on the second evolutionary process. The experimental results reveal the following dependence: the lower number of rows, the higher conditional probability. This means that the


Graph A: Circuit functionality evolution, $\gamma(F_1)$

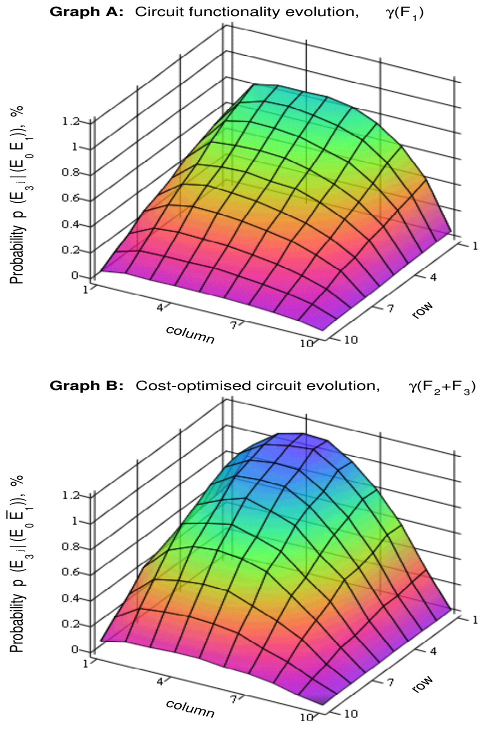
Graph B: Cost-optimised circuit evolution, $\gamma(F_2+F_3)$

Figure 2: Conditional probabilities for cell type genes. Axes named column and row define the position of logic gate in the circuit layout.


Graph A: Circuit functionality evolution, $\gamma(F_1)$

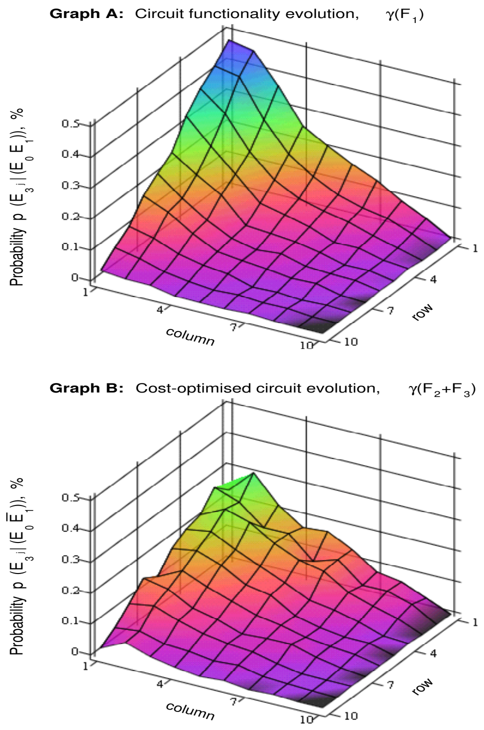
Graph B: Cost-optimised circuit evolution, $\gamma(F_2+F_3)$

Figure 3: Conditional probabilities for cell type genes calculated for active logic gates. Axes named column and row define the position of logic gate in the circuit layout.


Graph A: Circuit functionality evolution, $\gamma(F_1)$

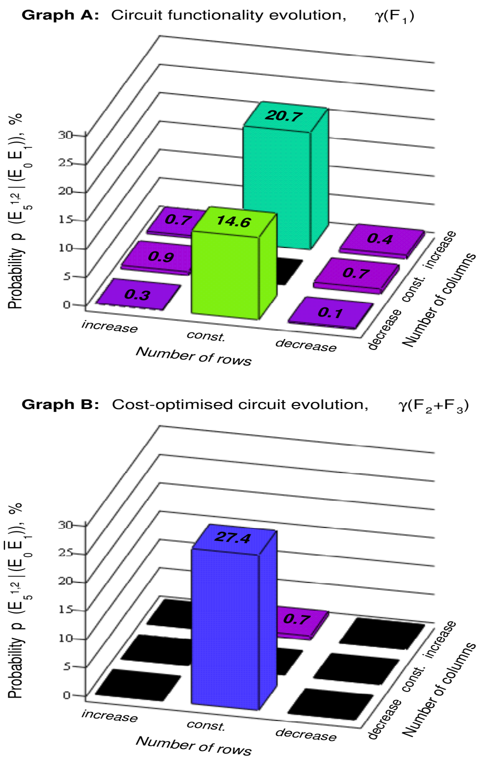
Graph B: Cost-optimised circuit evolution, $\gamma(F_2+F_3)$

Figure 4: Conditional probabilities for connection genes. Axes named column and row define the position of logic gate in the circuit layout.


Graph A: Circuit functionality evolution, $\gamma(F_1)$

Graph B: Cost-optimised circuit evolution, $\gamma(F_2+F_3)$

Figure 5: Conditional probabilities for connection genes calculated for active logic gates. Axes named column and row define the position of logic gate in the circuit layout.



Graph A: Circuit functionality evolution, $\gamma(F_1)$

Graph B: Cost-optimised circuit evolution, $\gamma(F_2+F_3)$

Figure 6: Conditional probabilities for circuit layout genes. The probabilities are decomposed by the direction of circuit layout modification.

cell type genes located in the rectangular array in low rows influence stronger on the evolutionary process rather then other one. Fig.3 shows the similar results that have been described above, but the conditional probabilities have been calculated only for active logic gates. The cell type genes participate in the first evolutionary process more actively, then in the second one. There is clearly defined the area of the most influenced cell type genes that is located in smaller index of rows and columns. Such effect is not appeared for the second evolutionary process. The cell type gene almost does not influence on the second evolutionary process. Comparison of sharpness of Graphs A at Fig.2 and Fig.3 shows that the second shape is sharper because the redundant genes are not taken into account. This proves the importance of the redundancy.

5.2 Behaviour of connection genes

The experimental results similar to cell type genes have been obtained for connection genes (Fig.4 - Fig.5). The shape of graphs are sharper for the case when only active logic gates are taken into account. This means that the connection genes influence relatively strongly on the second evolutionary process. (Fig.5 Graph B). Comparing the results obtained for cell type and connection genes (Fig. 3 and Fig. 5) one can notice that the connection genes play more important role in the second evolutionary process.

5.3 Behaviour of circuit layout genes

Some interesting results have been obtained, when the analysis of circuit layout gene behaviour has been carried out. It is noticeable, that the number of rows has not been changed during the second evolutionary process. The number of columns has been decreased significantly. One can conclude that in the given particular case the optimisation is carried out for the number of columns rather then the number of rows. The algorithm defines that the optimal circuit layout has low number of rows and moderate number of columns.

5.4 Summary analysis

Analysing Fig. 2 - Fig. 6, we can conclude:

1. The conditional probabilities for cell type and connection genes are higher for processes $\gamma_{F_2+F_3}$, i.e. changing these genes during $\gamma_{F_2+F_3}$ brings more positive solutions.

2. There is an area of logic gates in the circuit layout that participate more actively in evolutionary processes. For two-bit multiplier this area is located within approximately rows 1-4 and columns 2-6. Therefore, these genes are more important during both evolutionary processes. It should be mentioned that the behaviour of the evolutionary process depends on the type of logic function aimed to be implemented.

3. Different types of genes and its location influence with various effectiveness during the

evolutionary process. The both stages of GA (γ_{F_1} and $\gamma_{F_2+F_3}$) can be improved by redistribution of mutation rate for different types of genes and their position according Fig.2 and Fig.4. The certain mutation rate could be chosen for active logic gates according Fig.3 and Fig.5 in order to achieve even better performance of the GA.

6 Conclusion

In this paper the probabilistic analysis has been performed in order to define how different types of genes and its location influence on GA performance. It was also expected that the behaviour of the evolutionary process depends on the fitness function chosen. It has been shown that the different types of genes influence differently on GA performance and are involved in evolutionary process differently. GA performance can be significantly improved by choosing the mutation rate according to the stage of evolutionary process, the type of gene and its position.

References

[1] Higuchi T., Iwata M., Kajitani I., Murakawa M., Yoshizawa S., and Furuya T. Hardware evolution at gate and function level. In *Proc. of the Int. Conf. on Biologically Inspired Autonomous Systems: Computation, Cognition and Action*. Durham, NC, USA, 1996.

[2] Thompson A. Evolving electronic robot controllers that exploit hardware resources. In *Proc. of the Third European Conference on Artificial Life (ECAL95)*, pages 640–656. Springer-Verlag, 1995.

[3] Layzell P. A new research tool for intrinsic hardware evolution. In Sipper M., Mange D., and Perez-Uribe A., editors, *Proc. Of the 2nd Int. Conf. on Evolvable Systems: From Biology to Hardware (ICES'98)*, volume 1478 of *Lecture Notes in Computer Science*, pages 47–56, Lausanne, Switzerland, 1998. Springer-Verlag, Heidelberg.

[4] Koza J.R., Benett III F.H., Andre D., and Keane M.A. Four problems for which a computer program evolved by genetic programming is competitive with human performance. In *Proc. of 1996 IEEE Int. Conference on Evolutionary Computation.*, pages 1–10. IEEE Press, 1996.

[5] Torresen J. A divide-and-conquer approach to evolvable hardware. In Sipper M., Mange D., and Perez-Uribe A., editors, *Proc. Of the 2nd Int. Conf. on Evolvable Systems: From Biology to Hardware (ICES'98)*, volume 1478 of *Lecture Notes in Computer Science*, pages 57–65, Lausanne, Switzerland, 1998. Springer-Verlag, Heidelberg.

[6] Kalganova T. Bidirectional incremental evolution in ehw. In *Proc. of the Second NASA/DoD Workshop on Evolvable Hardware*. IEEE Computer Society, July 2000.

[7] Kalganova T. and Miller J. Evolving more efficient digital circuits by allowing circuit layout evolution and multi-objective fitness. In Stoica A., Keymeulen D., and Lohn J., editors, *Proc. of the First NASA/DoD Workshop on Evolvable Hardware*, pages 54–63. IEEE Computer Society, July 1999.

[8] Kalganova T., Miller J., and Fogarty T. Some aspects of an evolvable hardware approach for multiple-valued combinational circuit design. In Sipper M., Mange D., and Perez-Uribe A., editors, *Proc. Of the 2nd Int. Conf. on Evolvable Systems: From Biology to Hardware (ICES'98)*, volume 1478 of *Lecture Notes in Computer Science*, pages 78–89, Lausanne, Switzerland, 1998. Springer-Verlag, Heidelberg.

[9] Miller J. and Thomson P. Aspects of digital evolution: Geometry and learning. In Sipper M., Mange D., and Perez-Uribe A., editors, *Proc. Of the 2nd Int. Conf. on Evolvable Systems: From Biology to Hardware (ICES'98)*, volume 1478 of *Lecture Notes in Computer Science*, pages 25–35, Lausanne, Switzerland, 1998. Springer-Verlag, Heidelberg.