ITHEA

International Journal

INFORMATION TECHNOLOGIES & KNOWLEDGE

2016 Volume 10 Number 3

International Journal
INFORMATION TECHNOLOGIES & KNOWLEDGE

Volume 10 / 2016, Number 3

International Journal "INFORMATION TECHNOLOGIES & KNOWLEDGE" (IJ ITK)
is official publisher of the scientific papers of the members of
the ITHEA International Scientific Society

IJ ITK rules for preparing the manuscripts are compulsory.
The **rules for the papers** for IJ ITK are given on www.ithea.org

Responsibility for papers published in IJ ITK belongs to authors.

International Journal "INFORMATION TECHNOLOGIES & KNOWLEDGE" Volume 10, Number 3, 2016
Edited by the **Institute of Information Theories and Applications FOI ITHEA**, Bulgaria, in collaboration with:
Institute of Mathematics and Informatics, BAS, Bulgaria; V.M.Glushkov Institute of Cybernetics of NAS, Ukraine;
Universidad Politechnika de Madrid, Spain; Hasselt University, Belgium;
St. Petersburg Institute of Informatics, RAS, Russia; Institute for Informatics and Automation Problems, NAS of the Republic of Armenia.
Printed in Bulgaria
Publisher ITHEA®
Sofia, 1000, P.O.B. 775, Bulgaria. www.ithea.org, e-mail: info@foibg.com
Technical editor: Ina Markova

Издател: ИТЕА®, София 1000, ПК 775, България, www.ithea.org, e-mail: info@foibg.com

ISSN 1313-0455 (printed) **ISSN 1313-048X (online)**

EVOLUTIONARY SYNTHESIS OF QCA CIRCUITS: A CRITIQUE OF EVOLUTIONARY SEARCH METHODS BASED ON THE HAMMING ORACLE

R. Salas Machado, J. Castellanos, R. Lahoz-Beltra

***Abstract**: This paper introduces a discussion about evolutionary search methods based on Hamming oracle. In many optimization problems, the design of the fitness function includes the Hamming distance being referred this kind of functions as Hamming oracle. In this paper we adopt a critical look and ask ourselves to what extent genetic algorithms and other related evolutionary methods truly mimic evolution. We tested three evolutionary search methods taken as a case study the evolutionary synthesis of quantum-dot cellular automata circuits. Our main conclusion is that evolutionary search methods do not mimetic Darwinian evolution because knowledge is not obtained from the evolutionary surface exploration: evolution is the result of the 'knowledge' embedded by the researcher or human expert into the fitness function. Maybe a more appropriate denomination would be "combinatorial search algorithms" such as Minimax, Alpha-beta pruning, etc.*

***Keywords**: Evolutionary search methods, genetic algorithms, Dawkins weasel program, Hamming oracle*

***ACM Classification Keywords**: I.6 Simulation and Modeling*

Introduction

One of the key tasks in genetic and evolutionary algorithms is the evaluation of the quality, goodness or merit of a given solution, represented by an array, which is referred to chromosome. Generally, this evaluation is performed by an objective function or fitness that maps each chromosome or solution onto a real number, representing a measure of the optimality of a chromosome. In many optimization problems, the design of the fitness function includes the Hamming distance being referred this kind of functions as *Hamming oracle* (Figure 1). According to [Dembsky et al., 2007] an array or string, i.e. the chromosome, is then presented (*input*) to the Hamming oracle which assigns the string a rank, based on its proximity to a desired target (the *output* dependent on Hamming distance). In such cases, the desired target or optimal chromosome, could be defined by setting directly the optimal solution, e.g. in a theoretical study of the convergence of an evolutionary algorithm. In other cases, the desired target is

defined by setting the quantitative and qualitative features for a given optimal solution, e.g. in industrial design optimization we could define the most appropriate strength, weight, cost, and durability of a product. However, today one of the biggest criticisms received by genetic algorithms and related evolutionary search techniques is that in many practical applications the fitness function contains a lot of information about the optimal solution, such 'knowledge' being provided by a human expert.

In this paper we adopt a critical look and ask ourselves to what extent genetic algorithms (GAs) and other related evolutionary methods truly mimic the evolution in the search for an optimal or near-optimal solution. We tested three evolutionary search methods taken as a case study the evolutionary synthesis of quantum-dot cellular automata circuits. Our assumption is that the researcher or human expert feeds the Hamming oracle with an excess of knowledge, so the fitness landscape reduces its area significantly.

During the last decade there has been an attempt to apply GAs to the efficient and optimal design of QCA circuits. In all the following examples, the fitness is the result of comparing the output of the evaluated circuit with a predefined truth table or desired target: evolution is guided by a Hamming oracle. For instance, [Kamrani et al. 2012] applied GAs to design QCA circuits optimizing the number of gates and clock cycles by utilization of NAND and inverter gates. In this approach a tree structure is used to represent a chromosome, i.e. a candidate QCA circuit. The fitness function evaluates as positive effects a size reduction of the tree, and therefore a small number of QCA cells. Consequently, the negative effects are a result of any enlargement of the tree, and thus a greater number of QCA cells. A similar approach is used by [Bonyadi et al., 2007] and [Houshmand et al., 2009, 2011] representing a chromosome as a tree structure. However, the tree is constructed with majority and inverter gates and the leaves can be either logical value 1 or Boolean variables. Once again, the fitness function minimize the number of gates through promoting the rule that 'fewer nodes is a better solution'. Applying a more sophisticated methodology [Vilela Neto et al., 2007] introduced a coevolutionary model of GA optimizing the desired logic by the evolution of circuit topology, type cell and clock zone. The fitness is calculated by comparing the output of the candidate circuit with a truth table. However, the 'oracle needs some extra help' including the fitness function some terms –depending on the specific problem - to prevent local trapping problems.

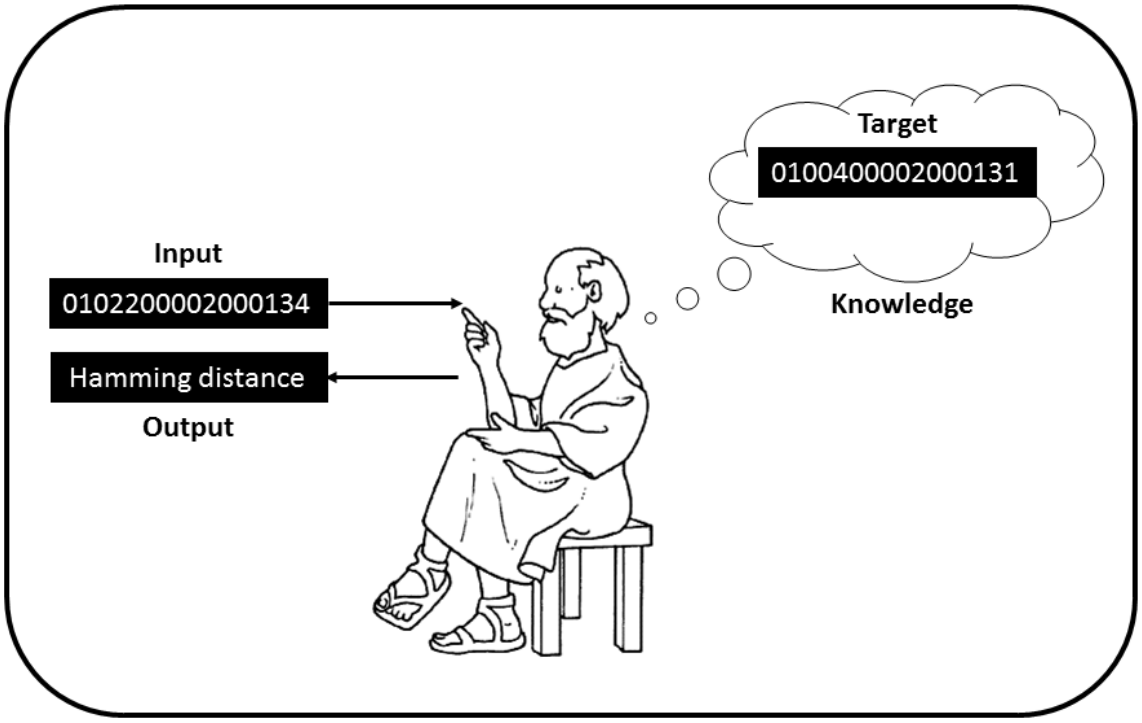


Figure 1.- Hamming oracle (for explanation see text)

Simulation experiments

Quantum-dot cellular automata (QCA) are a nanoscale technology based on Coulombic forces instead of current (e.g. CMOS) technology. A QCA cell (Figure 2) is the basic element being composed of four dots and two electrons. Electrons due to tunnel and the Coulomb repulsion effects occupy the dots, leading to two stable arrangements or polarizations which are encoded as 0 or 1. Thus, electrons arrange diagonally in order to be at a maximum distance from each other (Figure 2). QCA circuits are combinational logic circuits constructed from the binding of QCA cells, being the fundamental QCA logic devices: QCA wire, QCA majority gate and QCA inverter. Once a QCA circuit is designed the logical functionality of the circuit is tested through simulation. At present one of the most used tools to create and conduct the simulation of a novel QCA circuit is QCADesigner [Walus et al, 2004] [Walus Group, 2009].

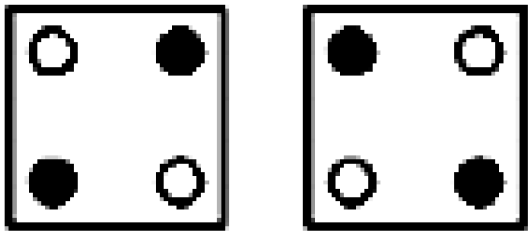


Figure 2.- QCA cell showing (left) polarization *P*=+1 ('bit state 1')
and (right) polarization *P*=-1 ('bit state 0')

Simulations experiments were conducting studying the synthesis of four QCA circuits (Figure 3): XOR gate, inverted XOR gate, AND gate, 5-input majority gate [Angizi et al., 2015]. After choosing these simple circuits, the circuits were designed with QCADesigner verifying their logical functionality with a bistable simulation engine [Walus et al., 2004], i.e. assuming that each cell is a simple two-state system. The two-state model assumes the following Hamiltonian:

$$H_i = \sum_j \begin{bmatrix} -\frac{1}{2} P_j E_{i,j}^k & -\gamma_i \\ -\gamma_i & \frac{1}{2} P_j E_{i,j}^k \end{bmatrix} \quad (1)$$

where $E_{i,j}^k$ is the kink energy between cell *i* and *j*, P_j is the cell polarization and γ is the tunneling energy. Using the Schrodinger equation the simulation engine obtains the state of each cell with respect to other cells that are close in the circuit, thus:

$$P_i = \frac{\frac{E_{i,j}^k}{2\gamma} \sum_j P_j}{\sqrt{1 + \left(\frac{E_{i,j}^k}{2\gamma} \sum_j P_j \right)}} \quad (2)$$

where P_i is the polarization state of cell *i* and P_j the polarization state of the cells in the neighborhood.

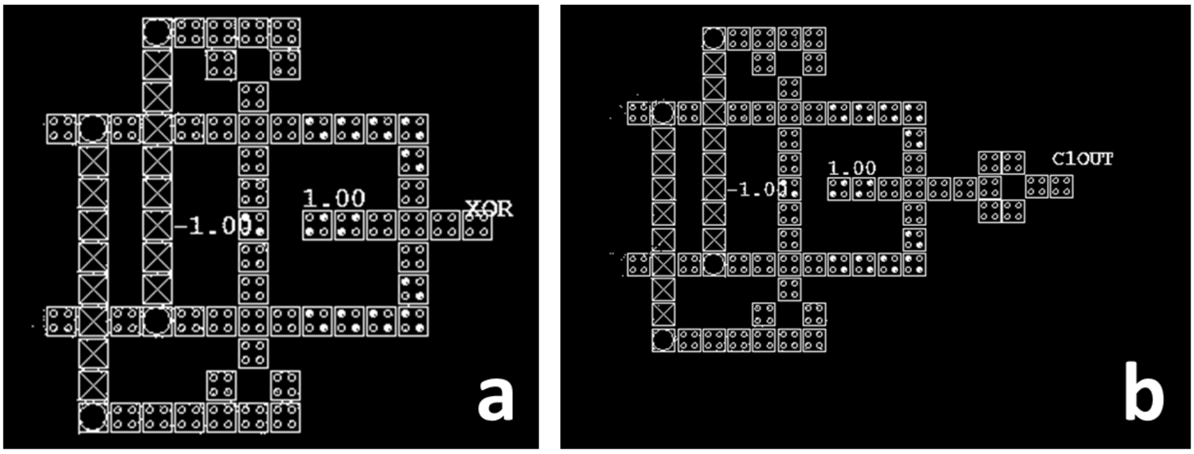


Figure 3.- QCA circuits designed and tested with QCADesigner: (a) XOR gate (b) inverted XOR.

Since QCA circuits are clocked, different clock zones were assigned to QCA cells, pumping the information through the circuit according to QCADesigner signal clocking. QCA cells perform computations using a four-phase clock representing such cells in different color according to its clock zone. Following, each circuit was transformed into an array of 13 rows and 19 columns (Figure 4), assigning to each cell a value: 0 (empty cell or layout without QCA cell), 1 (QCA cell with clock zone 0), 2 (QCA cell with clock zone 1), 3 (QCA cell with clock zone 2) or 4 (QCA cell with clock zone 3). According to Figure 4 each of arrays would be representing the desired or target QCA circuit.

```
#XOR

0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 1, 1, 1, 1, 1, 2, 2, 2, 3, 3, 3, 3, 0, 0, 0, 0, 0, 0,
0, 0, 1, 0, 1, 0, 0, 2, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0,
0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0,
0, 0, 1, 0, 1, 0, 0, 2, 0, 3, 3, 4, 4, 4, 4, 0, 0, 0, 0,
0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0,
0, 0, 1, 0, 1, 0, 0, 2, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0,
1, 1, 1, 1, 1, 1, 2, 2, 2, 3, 3, 3, 3, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
```

```
#INVERTED XOR

0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 1, 1, 1, 1, 1, 2, 2, 2, 3, 3, 3, 3, 0, 0, 0, 0, 0, 0,
0, 0, 1, 0, 1, 0, 0, 2, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0,
0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 4, 0, 0, 4, 4, 0, 0,
0, 0, 1, 0, 1, 0, 0, 2, 0, 3, 3, 4, 4, 4, 4, 4, 0, 4, 4,
0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 4, 0, 0, 4, 4, 0, 0,
0, 0, 1, 0, 1, 0, 0, 2, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0,
1, 1, 1, 1, 1, 1, 2, 2, 2, 3, 3, 3, 3, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
```

```
#AND

0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
```

```
#5 INPUT MAJORITY

0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 1, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 2, 2, 2, 2, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 1, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
```

Figure 4.- QCA circuits represented by an array (for explanation see text)

Concluded the circuit design step, we studied the evolutionary synthesis of QCA circuits by different evolutionary algorithms which details are described below.

Evolutionary search was studied testing three evolutionary search and optimization methods:

- Random search (RS).- RS is an optimization algorithm that does not use the gradient of the problem. Thus, the algorithm allows all QCA cells to mutate in any generation, but with the possibility that correct letters can be mutated again. The algorithm consists of the following steps: (1) obtains a random initial string of same length as desired or target circuit, (2) apply the evolutionary feedback loop: (2.1) obtains the mutated offspring, (2.2) selects the best circuit and pass it to the next generation and (2.3) stop the loop when a target circuit is reached.
- Partitioned search (PS).- PS is the name given by [Dembsky et al., 2007] to denote the Dawkins' weasel program: an experiment with computer described in the book by Richard Dawkins entitled *The Blind Watchmaker* [Dawkins, 1986]. Based on 'infinite monkey theorem' the PS algorithm simulates the Darwin's cumulative selection principle: Darwin's theory of evolution by natural selection in organisms with asexual reproduction (individuals reproduce by dividing –bipartition- in two 'children'). In this case, QCA cells mutate in any generation, but correct positions are locked (i.e. preserved) as soon as they appear being impossible that correct letters can be mutated again. The algorithm consists of the following steps: (1) obtains a random initial string of same length as desired or target circuit, (2) apply the evolutionary feedback loop: (2.1) obtains the mutated offspring, (2.2) selects the best circuit and pass it to the next generation and (2.3) lock correct string positions preventing further mutation (2.4) stop the loop when a target circuit is reached.
- Darwinian selection pressure (DSP).- DSP is the name we have chosen to refer to a genetic algorithm without recombination or mutation [Alajmi and Wright, 2014]. Therefore, we study the time or number of generations required for the population to be composed of the best solutions (or circuits) found in the initial generation.

In this paper, the fitness of strings, i.e. QCA circuits, is calculated in RS and PS algorithms as follows:

$$f(s) = \frac{L - d_{s,\text{target}}^{\min}}{L} \tag{3}$$

Regarding the above expression s is a label that identifies a given circuit, L the length of string depicting the circuit and $d_{s,\text{target}}^{\min}$ the distance of the circuit closest to the target circuit. Note that for $d_{s,\text{target}}^{\min} = 0$, i.e. when the target circuit is reached, the fitness $f(s)$ takes the maximum value being equal to 1.

In the case of DSP algorithm the fitness is given by the following expression:

$$f(s) = L - H \tag{4}$$

where H is the Hamming distance between the circuit s and the target circuit. Note that when the target circuit is reached then H is zero, and therefore the circuit is ranked with a maximum fitness value (in our simulation experiments, 247). Just as it was done in the expression (3) we could also have standardized (4) measure (maximum fitness value equal to 1) dividing by L:

$$f(s) = \frac{L - H}{L} \tag{5}$$

Applying probability theory [Dembsky et al., 2007] obtained the expressions to estimate the median number of queries required for success - the optimum circuit or target has been found - in RS and PS search algorithms (Table 1). In this study we defined such values like Q_1 and Q_2 for RS and PS algorithms, respectively. Details and mathematical model of these two algorithms are described in [Dembsky et al., 2007]. We also include 'takeover time' (Q_3, Table 1): a DSP performance measure introduced by [Bäck, 1996]. $Q3$ is defined as the number of generations or queries for the offspring (or population) to be filled with the best solutions (or circuits) found in the initial generation but in the absence of crossover and mutation.

Table 1.- Performance evaluation of evolutionary search algorithms

Evolutionary search algorithm	Performance measure
Random search	$Q_1 = -card(A)^L \ln(1-0.5)$
Partitioned search	$Q_2 = \frac{\log\left(1-0.5^{\frac{1}{L}}\right)}{\log\left(1-\frac{1}{card(A)}\right)}$
Darwinian selection pressure	$Q_3 = \frac{1}{\ln(n)}\left(\ln(N) + \ln(\ln(N))\right)$

Simulation experiments were conducted using the following programs in Python 3.4.4 language: RS and PS algorithms, running *weasel_3.py* and *weasel_locked_3.py* programs respectively [Pedersen, 2009]; DSP algorithm, modifying the code of a simple genetic algorithm, i.e. *SGA.py* [Lahoz-Beltra, 2016]. All simulation experiments were performed using the following initial conditions and parameter values: L=247 (19x13, see Figure 4), the alphabet of possible symbols A={0, 1, 2, 3, 4} with $card(A)$=4, offspring size per generation (population size) N=50 and mutation rate equal to 0.08. In the particular case of DSP algorithm, n=2 represents the number of individuals in the tournament step of the algorithm. The simulation experiment, i.e. the evolutionary synthesis of QCA circuits, ends when an optimum or target circuit is achieved or maximum CPU time is reached.

Results

The results obtained in the experiments support the hypothesis that evolutionary search methods based on a Hamming oracle could not be mimicking to Darwinian evolution. A plausible explanation is because knowledge it is not obtained from the exploration of the evolutionary or fitness surface: evolution is the result of the 'knowledge' embedded by the researcher or human expert into the fitness function, i.e. the oracle (Figure 1). This conclusion is supported by the following simulation results.

First, experiments with RS and PS have yielded to the following observations. Figure 5 shows the evolution of the circuits under the RS algorithm. A striking result is that in the evolutionary synthesis of XOR and inverted XOR gates seems to be some evolutionary convergence (Figure 5 a,b) whereas such a convergence is not observed for AND as well as 5-input majority gates (Figure 5 c,d). One possible explanation could be the number of cells in the array in state 0. However, when evolution takes place by means of the PS algorithm then the evolutionary synthesis of the four QCA circuits is successfully achieved, and four QCA circuits follow the same evolutionary pattern (Figure 6). Indeed, as predicted by the theory of probability Q_1=3.0649 10^{172}, thus under RS algorithm (Figure 5) the target circuit is never reached. Moreover, the time required far exceeds the age of the universe, i.e. 13.7 10^9 according to the NASA's WMAP project and for this reason the value of fitness is always below the maximum value (i.e. 1). In contrast, and according to Q_2=26.3386 value under PS algorithm (Figure 6) evolutionary synthesis of the four QCA circuits is successful reaching in approximately 26 generations the target circuit, and hence the maximum fitness (i.e. 1). These results are confirmed if we take as an example the XOR gate simulation experiment. Table 2 shows in RS and PS search algorithms the statistical summary for the maximum fitness per generation. In case we take as a measure of centralization the median, similarly to as was done for Q_1 and Q_2, there are significant differences between the medians of RS (Me=0.48) and PS (Me=0.79) as is shown in the box and whisker plot (Figure 7).

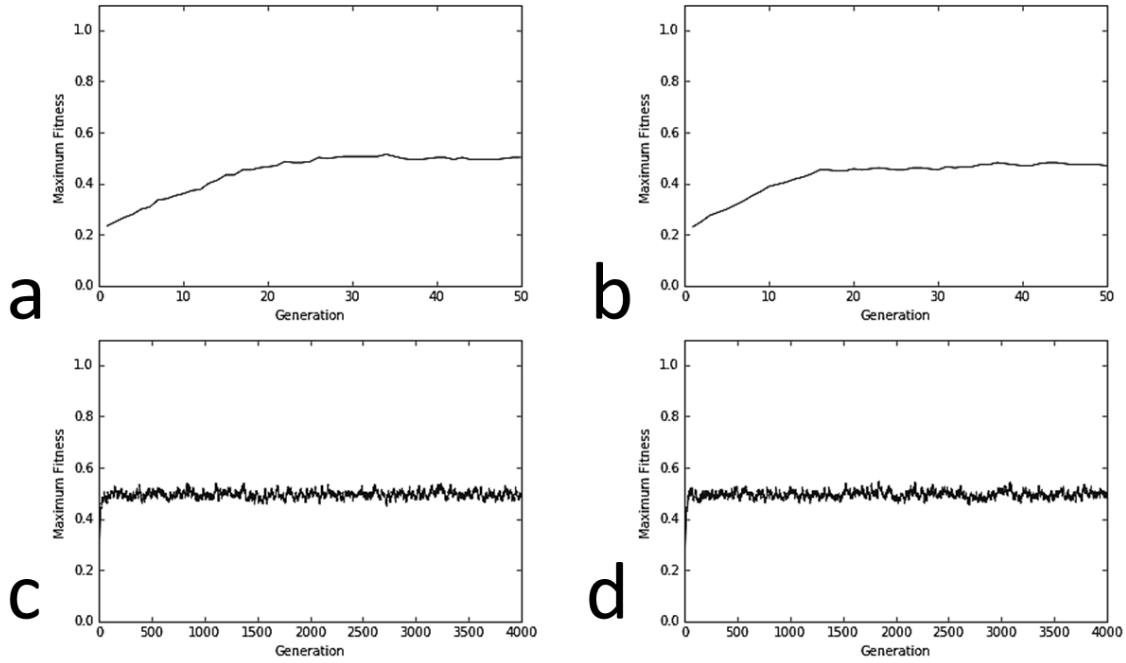


Figure 5.- Performance graph of the evolutionary synthesis of QCA circuits under RS algorithm: (a) XOR gate, (b) inverted XOR, (c) AND gate, (d) 5-input majority gate

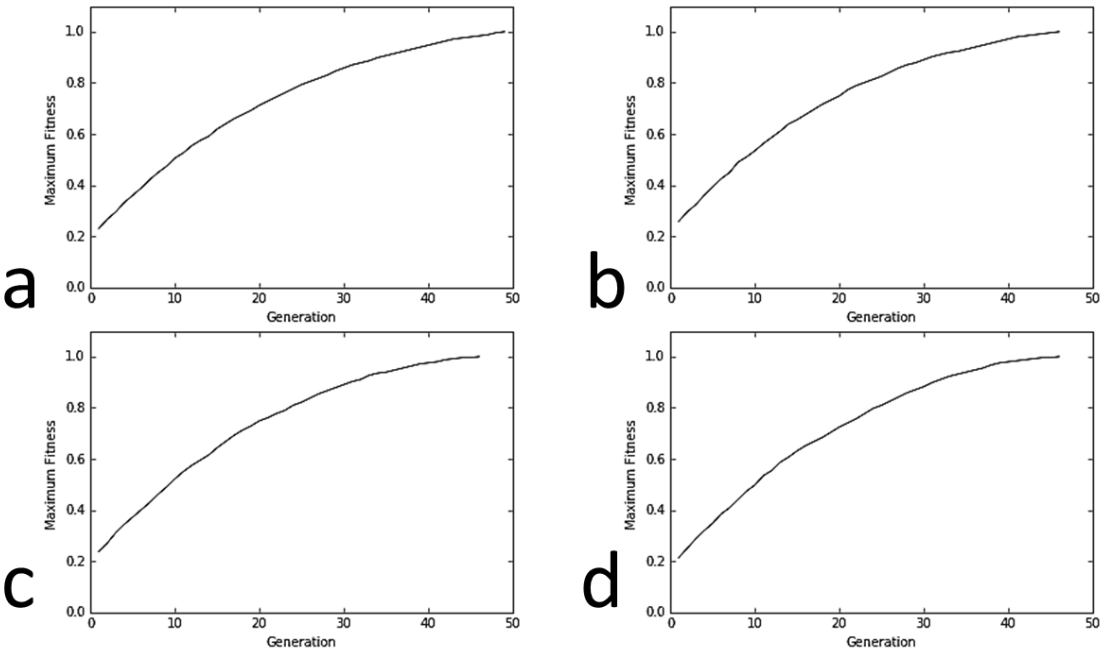


Figure 6.- Performance graph of the evolutionary synthesis of QCA circuits under PS algorithm: (a) XOR gate, (b) inverted XOR, (c) AND gate, (d) 5-input majority gate

Table 2.- Statistical summary*

	RS	**PS**
Sample size	200	98
Mean	0.493421	0.736305
Variance	0.000205907	0.0516711
Standard deviation	0.0143495	0.227313
Minimum	0.461538	0.222672
Maximum	0.534413	1.0
Rank	0.0728745	0.777328

*Note that in the case of RS, sample size is 200 because we get together in the same sample the fitness values of the last 100 iterations of two independent simulation experiments.

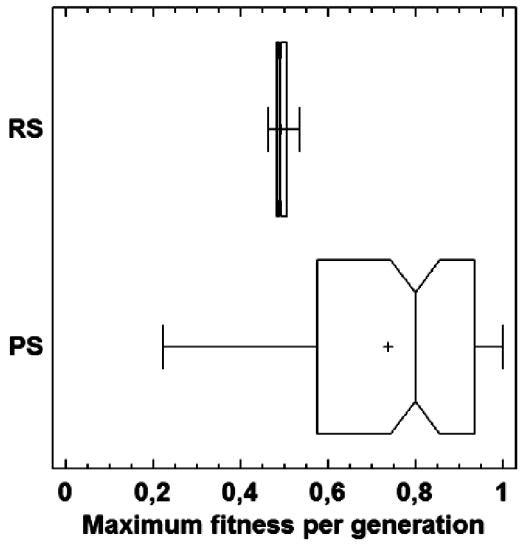


Figure 7.- Box and whisker plot of the evolutionary synthesis experiments conducted with XOR gate under RS and PS algorithms (Mann-Whitney with W = 15981.5 and *P*-value = 0.0; the notch represents a confidence interval for the median and the cross is the arithmetic mean)

Secondly, as can be seen (Figure 8) the results obtained in the study conducted with DSP algorithm, thus a genetic algorithm without crossover and mutation, show DSP search method does not find the optimum circuit. According to the theoretical model Q_3=9.1661, thus the number of generations required for the population to be composed of the best solutions is approximately equal to 9. The results show a local trapping phenomenon, without the population reaches the maximum fitness (i.e. 274). Interestingly, evolutionary convergence resembles the poor performance observed during the evolutionary synthesis of XOR and inverted XOR gates under random search (RS) algorithm (Figure 5).

Conclusion

In conclusion, evolutionary search methods based on Hamming oracle do not mimetic evolution by Darwinian natural selection because knowledge is not obtained from the exploration of the evolutionary surface: evolution is the result of the 'knowledge' embedded by the researcher or human expert into the fitness function. Since this knowledge reduces the size of the search space genetic algorithms and other related evolutionary methods based on Hamming oracle should be termed as "combinatorial search algorithms" together with techniques such as Minimax, Alpha-beta pruning, etc.

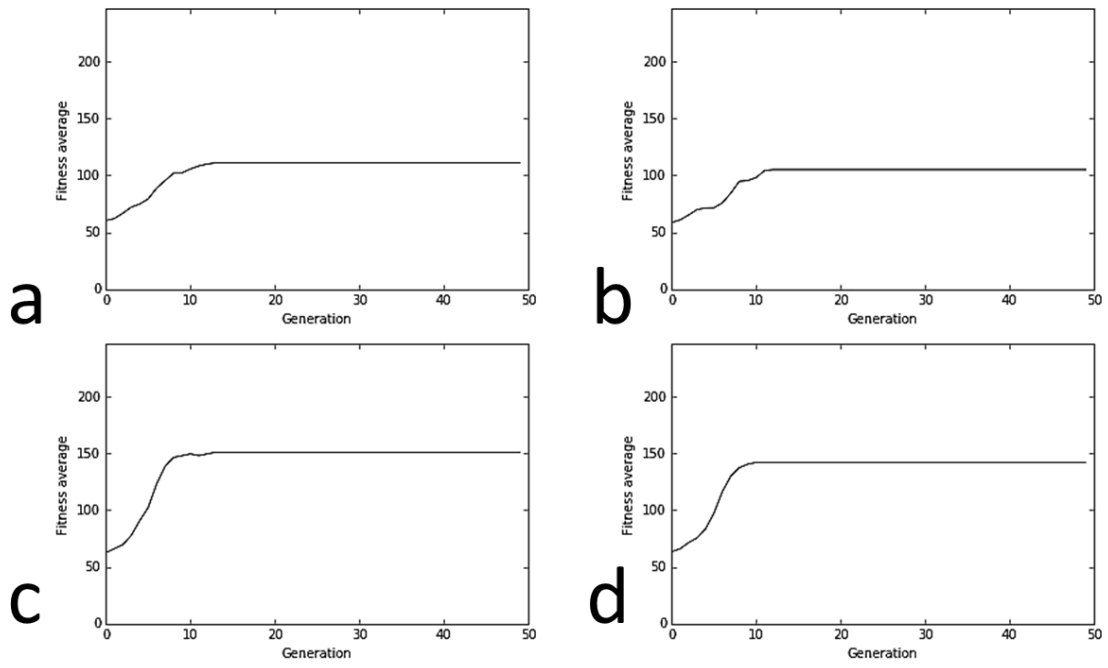


Figure 8.- Performance graph of the evolutionary synthesis of QCA circuits under DSP algorithm: (a) XOR gate, (b) inverted XOR, (c) AND gate, (d) 5-input majority gate

Bibliography

[Alajmi and Wright, 2014] A. Alajmi, J. Wright. 2014. Selecting the most efficient genetic algorithm sets in solving unconstrained building optimization problem. International Journal of Sustainable Built Environment 3: 18-26.

[Angizi et al., 2015] S. Angizi, S. Sarmadi, S. Sayedsalehi, K. Navi. 2015. Design and evaluation of new majority gate-based RAM cell in quantum-dot cellular autómata. Microelectronics Journal 46: 43-51.

[Bäck, 1996] T. Bäck. 1996. Evolutionary Algorithms in Theory and Practise. New York: Oxford University Press.

[Bonyadi et al., 2007] M.R. Bonyadi, S.M.R. Azghadi, N.M. Rad, K. Navi, E. Afjei. 2007. Logic optimization for majority gate-based nanoelectronic circuits based on genetic algorithm. In: Proceedings International Conference on Electrical Engineering, 2007. ICEE '07:1-5.

[Dawkins, 1986] R. Dawkins. 1986. The Blind Watchmaker. New York: W. W. Norton & Company.

[Dembsky et al., 2007] W.A. Dembsky, W. Ewert, R.J. Marks II. The Evolutionary Informatics Lab. http://www.evoinfo.org/index/

[Houshmand et al., 2009]. M. Houshmand, S.H. Khayat, R. Rezaei. 2009. Genetic algorithm based logic optimization for multi-output majority gate-based nano-electronic circuits. In: Proceedings of the IEEE International Conference on Intelligent Comput-ing and Intelligent Systems, 20–22 November 2009, Shanghai, China, IEEE: 584–588.

[Houshmand et al., 2011] M. Houshmand, R. Rezaee Saleh, M. Houshmand. 2011. Logic minimization of QCA circuits using genetic algorithms. In: Soft Computing in Industrial Applications, AISC 96 (Eds. A. Gaspar-Cunha et al.) Springer-Verlag, Berlin, Heidelberg: 393–403.

[Kamrani et al., 2012] M. Kamrani, H. Khademolhosseini, A. Roohi, P. Aloustanimirmahalleh. 2012. A novel genetic algorithm based method for efficient QCA circuit design. In: Advances in Computer Science, Eng. & Appl., AISC 166 (Eds. D.C. Wyld et al.) Springer-Verlag, Berlin, Heidelberg: 433-442.

[Lahoz-Beltra, 2016] R. Lahoz-Beltra. 2016. Simple genetic algorithm (SGA). figshare. dx.doi.org/10.6084/ m9.figshare.3397714.v2

[Pedersen, 2009] A.G. Pedersen. 2009. Weasel programs in Python. http://www.cbs.dtu.dk/courses/27615.mol/weasel.php

[Vilela Neto et al., 2007] O.P. Vilela Neto, M.A.C. Pacheco, C.R. Hall Barbosa. 2007. Neural network simulation and evolutionary synthesis of QCA circuits. IEEE Transactions on Computers 56: 191-201.

[Walus et al, 2004] K. Walus, T.J. Dysart, G.A. Jullien, R.A. Budiman. 2004. QCADesigner: a rapid design and Simulation tool for quantum-dot cellular autómata. IEEE Transactions on Nanotechnology 3: 26-31.

[Walus Group, 2009] QCADesigner. Walus Group at the University of British Columbia. http://www.mina.ubc.ca/qcadesigner.

Authors' Information

Ramses Salas Machado *– Research Scholar at Carlos III University of Madrid and member of the Grupo de Computación Natural, Departamento de Inteligencia Artificial, Facultad de Informática, Universidad Politécnica de Madrid ; e-mail: ramsesjsm@gmail.com*

Major Fields of Scientific Research: Natural computing

Juan B. Castellanos Peñuela *– Departamento de Inteligencia Artificial, Facultad de Informática, Universidad Politécnica de Madrid, Spain; e-mail: jcastellanos@fi.upm.es*

Major Fields of Scientific Research: Natural computing, membrane computing, molecular computing and artificial neural networks.

Rafael Lahoz–Beltra *– Department of Applied Mathematics, Faculty of Biological Sciences, Complutense University of Madrid, 28040 Madrid, Spain; e-mail: lahozraf@ucm.es*

Major Fields of Scientific Research: Evolutionary computation, bioinspired algorithms

SEARCH, PROCESSING, AND APPLICATION OF LOGICAL REGULARITIES OF CLASSES

Yury Zhuravlev, Vladimir Ryazanov

***Abstract**: A model of the type of estimates calculation, based on the systems of logical regularities of classes (LRC), to solve supervised classification problems is considered. The basic definitions are given. Two approaches for processing sets of LRC (based on the construction of the shortest logical class descriptions and clustering sets of LRC) are described. Different ways to use LRC are considered: based on LRC sets classification, construction of various logical descriptions of classes, the calculation of informative features, logical correlations of features, minimization of the feature space, assessment of outliers.*

***Keywords**: classification, pattern recognition, the calculation of estimates, logical regularity of class.*

***ACM Classification Keywords**: I.2.4 ARTIFICIAL INTELLIGENCE Knowledge Representation Formalisms and Methods – Predicate logic, I.5.1 PATTERN RECOGNITION Models – Deterministic, H.2.8 Database Applications, Data mining.*

1. Introduction

The standard task of supervised classification by precedents was considered with n features, l disjoint classes $K_1, K_2, ..., K_l$ and m reference objects $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m\}$ (training sample) [Zhuravlev, 1978]. The notation $\tilde{K}_i = X \cap K_i, i = 1, 2, ..., l,$ and assumption $\tilde{K}_i \neq \varnothing$, $i = 1,2,...,l$ were used. Arbitrary object $\mathbf{x} \in \bigcup_{i=1}^{l} K_i$ is identified by its description in the form of feature vector $\mathbf{x} = (x_1, x_2, ..., x_n)$. For simplicity, we assume that $x_i \in R$ (binary-valued and k - valued features are considered as a special case of real-valued). When the training sample analysis, we will often (especially without specifying) write simply K_i implying that we consider always in training $\tilde{K}_i$.

2. Logical Regularities of Classes and Basic Definitions.

Consider the following set of elementary predicates that depend parametrically on unknown $\Omega_1, \Omega_2 \subseteq \{1, 2, ..., n\}, \mathbf{c}^1, \mathbf{c}^2 \in R^n$ [Zhuravlev et al, 2006; Ryazanov, 2007]. We will use the notation $(x \le a) = \begin{cases} 1, & x \le a, \\ 0, & \text{otherwise.} \end{cases}$

Definition 1. Predicate

$$P^{\Omega_1, \mathbf{c}^1, \Omega_2, \mathbf{c}^2}(\mathbf{x}) = \&_{j \in \Omega_1} (c_j^1 \le x_j) \&_{j \in \Omega_2} (x_j \le c_j^2) \quad (1)$$

is called the logical regularity of class (LRC) K_λ, $\lambda = 1,2,...,l$, if

1. $\exists \mathbf{x}_t \in \tilde{K}_\lambda : P^{\Omega_1, \mathbf{c}^1, \Omega_2, \mathbf{c}^2}(\mathbf{x}_t) = 1,$
2. $\forall \mathbf{x}_t \notin \tilde{K}_\lambda : P^{\Omega_1, \mathbf{c}^1, \Omega_2, \mathbf{c}^2}(\mathbf{x}_t) = 0,$
3. $P^{\Omega_1, \mathbf{c}^1, \Omega_2, \mathbf{c}^2}(\mathbf{x}) = \underset{\{P^{\Omega_1^*, \mathbf{c}^{*1}, \Omega_2^*, \mathbf{c}^{*2}}(\mathbf{x})\}}{extr} \Phi(P^{\Omega_1^*, \mathbf{c}^{*1}, \Omega_2^*, \mathbf{c}^{*2}}(\mathbf{x}))$, where Φ - predicate quality criterion.

The predicate (1), satisfying only the first two constraints, is called admissible predicate of this class.

The predicate (1), satisfying only the first and third restrictions, is called partial logical regularity of class K_λ.

The set $N(P^{\Omega_1, \mathbf{c}^1, \Omega_2, \mathbf{c}^2}) = \{\mathbf{x} \in R^n : c_j^1 \le x_j, j \in \Omega_1; x_j \le c_j^2, j \in \Omega_2\}$ will be called the interval of predicate (analogue to intervals of elementary conjunctions in the algebra of logic).

Example of interval of LRC with $\mathbf{x}_t \in \tilde{K}_\lambda$ is presented in Figure 1. Here black marks marked objects satisfying this LRC.

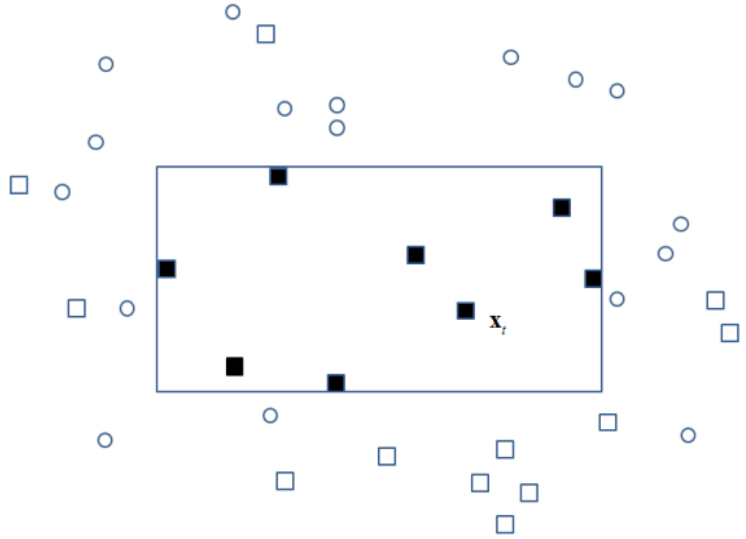


Figure 1. Example of LRC with $\mathbf{x}_t \in \tilde{K}_\lambda$

Two predicates $P^{\Omega_1,\mathbf{c}^1,\Omega_2,\mathbf{c}^2}(\mathbf{x})$, $P^{\Omega_3,\mathbf{c}^3,\Omega_4,\mathbf{c}^4}(\mathbf{x})$ are called equivalent if $P^{\Omega_1,\mathbf{c}^1,\Omega_2,\mathbf{c}^2}(\mathbf{x}_t) = P^{\Omega_3,\mathbf{c}^3,\Omega_4,\mathbf{c}^4}(\mathbf{x}_t), t = 1,2,...,m$.

Two intervals $N(P^{\Omega_1,\mathbf{c}^1,\Omega_2,\mathbf{c}^2})$, $N(P^{\Omega_3,\mathbf{c}^3,\Omega_4,\mathbf{c}^4})$ are called equivalent if $N(P^{\Omega_1,\mathbf{c}^1,\Omega_2,\mathbf{c}^2}) \cap X = N(P^{\Omega_3,\mathbf{c}^3,\Omega_4,\mathbf{c}^4}) \cap X$.

Definition 2. The following criterion

$$F(P^{\Omega_1,\mathbf{c}^1,\Omega_2,\mathbf{c}^2}(\mathbf{x})) = \left|\{\mathbf{x}_i : \mathbf{x}_i \in \tilde{K}_\lambda, P^{\Omega_1,\mathbf{c}^1,\Omega_2,\mathbf{c}^2}(\mathbf{x}_i) = 1\}\right|$$

will be called as the standard quality criterion of the predicate of class K_λ.

Definition 3. A logical regularity of class (LRC) $P^{\Omega_1,\mathbf{c}^1,\Omega_2,\mathbf{c}^2}(\mathbf{x})$ is called minimal if there is no such equivalent LRC $P^{\Omega_3,\mathbf{c}^3,\Omega_4,\mathbf{c}^4}(\mathbf{x})$ that $N(P^{\Omega_3,\mathbf{c}^3,\Omega_4,\mathbf{c}^4}) \subset N(P^{\Omega_1,\mathbf{c}^1,\Omega_2,\mathbf{c}^2})$.

3. Finding of Logical Regularities with Standard Quality Criteria.

Consider the general problem of finding LRC with standard quality criteria. The problem will be solved under the restriction $\mathbf{x}_t \in \tilde{K}_\lambda$ where this object will be called the support object. First of all, we note the following. If $P^{\Omega_1,\mathbf{c}^1,\Omega_2,\mathbf{c}^2}(\mathbf{x})$ is LRC than $P^{\mathbf{c}^1,\mathbf{c}^2}(\mathbf{x})$ is also LRC. For simplicity, we omit the indices $\Omega_1 = \Omega_2 = \{1,2,\ldots,n\}$. There is enough to supplement the parameters $\mathbf{c}^1, \mathbf{c}^2$ on features, not included in Ω_1, Ω_2, by always running conditions $\min\limits_{\mathbf{x}_i \in K_\lambda} x_{ij} = c_j^1 \le x_j, j \notin \Omega_1$ or $x_j \le c_j^2 = \max\limits_{\mathbf{x}_i \in K_\lambda} x_{ij}, j \notin \Omega_2$. LRC $P^{\Omega_1,\mathbf{c}^1,\Omega_2,\mathbf{c}^2}(\mathbf{x})$ and $P^{\mathbf{c}^1,\mathbf{c}^2}(\mathbf{x})$ are equivalent. We shall seek "minimum" LRC (stretched on some subsets of K_λ).

Let $D_i^1 = \{d_{i1}^1, d_{i2}^1, \ldots, d_{iu_i}^1\}$ and $D_i^2 = \{d_{i1}^2, d_{i2}^2, \ldots, d_{iv_i}^2\}$ are all monotonically decreasing / increasing the choices of values of left and right borders of predicates (1) for the training set (possible values of features of objects of $\tilde{K}_\lambda$), respectively.

We assume further for simplicity that the first order of the training sample objects (and only they) belong to the class in question $\tilde{K}_\lambda = \{\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_h\}$.

We construct the numerical matrix $\mathbf{M} = \begin{pmatrix} \mathbf{B}_1^1\mathbf{B}_1^2\mathbf{B}_2^1\mathbf{B}_2^2\ldots\mathbf{B}_n^1\mathbf{B}_n^2 \\ \mathbf{C}_1^1\mathbf{C}_1^2\mathbf{C}_2^1\mathbf{C}_2^2\ldots\mathbf{C}_n^1\mathbf{C}_n^2 \end{pmatrix}_{m\times N}$,

$N = \sum\limits_{i=1}^{n}(u_i + v_i)$,

$\mathbf{B}_i^1 = (b_{ij}^{1q})_{h\times u_i}$, $q = 1,2,\ldots,h,\ i = 1,2,\ldots,n$, $j = 1,2,\ldots,u_i$,

$\mathbf{B}_i^2 = (b_{ij}^{2q})_{h\times v_i}$, $q = 1,2,\ldots,h,\ i = 1,2,\ldots,n$, $j = 1,2,\ldots,v_i$,

$\mathbf{C}_i^1 = (c_{ij}^{1q})_{(m-h)\times u_i}$, $q = 1,2,\ldots,m-h,\ i = 1,2,\ldots,n$, $j = 1,2,\ldots,u_i$,

$\mathbf{C}_i^2 = (c_{ij}^{2q})_{(m-h)\times v_i}$, $q = 1,2,\ldots,m-h,\ i = 1,2,\ldots,n$, $j = 1,2,\ldots,v_i$, где

$$b_{ij}^{1q} = \begin{cases} 1, & x_{qi} \geq d_{ij}^{1}, \\ 0, & \text{otherwise}, \end{cases} \qquad b_{ij}^{2q} = \begin{cases} 1, & x_{qi} \leq d_{ij}^{2}, \\ 0, & \text{otherwise}, \end{cases}$$

$$c_{ij}^{1q} = \begin{cases} 1, & x_{(h+q)i} < d_{ij}^{1}, \\ 0, & \text{otherwise}, \end{cases} \qquad c_{ij}^{2q} = \begin{cases} 1, & x_{(h+q)i} > d_{ij}^{2}, \\ 0, & \text{otherwise}. \end{cases}$$

Consider a set of vectors $\{< x_{ij}^{1}, x_{ij}^{2} >\}$, $< x_{ij}^{1}, x_{ij}^{2} > =$ $< x_{11}^{1}, x_{12}^{1}, ..., x_{1u_1}^{1}, x_{11}^{2}, x_{12}^{2}, ..., x_{1v_1}^{2}, x_{21}^{1}, x_{22}^{1}, ..., x_{2u_2}^{1}, x_{21}^{2}, x_{22}^{2}, ..., x_{2v_2}^{2}, ..., x_{n1}^{1}, x_{n2}^{1}, ..., x_{nu_n}^{1}, x_{n1}^{2}, x_{n2}^{2}, ..., x_{nv_n}^{2} >$ with restrictions

$$x_{ij}^{1}, x_{ij}^{2} \in \{0,1\}, \sum_{j=1}^{u_i} x_{ij}^{1} = 1, \sum_{j=1}^{v_i} x_{ij}^{2} = 1, \quad i=1,2,...,n. \tag{2}$$

We associate the choice of units in $< x_{ij}^{1}, x_{ij}^{2} >$ to the parameter values $c_i^1 \in D_i^1, c_i^2 \in D_i^2, i = 1,2,...,n$. The set of all predicates of possible boundaries D_i^1, D_i^2 is in a one-to-one correspondence with the set of binary vectors of data, so we will use also the notation $F(< x_{ij}^{1}, x_{ij}^{2} >)$ as an entry for the standard optimality criterion. We form the following systems of inequalities and equalities.

$$f_q^c(< x_{ij}^{1}, x_{ij}^{2} >) = \sum_{i=1}^{n} (\sum_{j=1}^{u_i} c_{ij}^{1q} x_{ij}^{1} + \sum_{j=1}^{v_i} c_{ij}^{2q} x_{ij}^{2}) \geq 1, q = 1,2,...,m-h \ , \tag{3}$$

$$f_q^b(< x_{ij}^{1}, x_{ij}^{2} >) = \sum_{i=1}^{n} (\sum_{j=1}^{u_i} (b_{ij}^{1q} - 1) x_{ij}^{1} + \sum_{j=1}^{v_i} (b_{ij}^{2q} - 1) x_{ij}^{2}) = 0, q = 1,2,...,h. \tag{4}$$

LRC search problem can be formulated as the following discrete optimization problem:

Task Z:

$F(< x_{ij}^1, x_{ij}^2 >)$ =< number of executed equations in (4)> → max,

with restrictions (2-3).

We associate to the task **Z** similar problem **ZC** with respect to real variables.

Task ZC:

< number of executed equations in (4)> → max,

with restrictions (3), (5-6)

$$x_{ij}^1 \geq 0\,,\ i = 1,2,...,n\,,\ j = 1,2,...,u_i\,, \tag{5}$$

$$x_{ij}^2 \geq 0\,,\ i = 1,2,...,n\,,\ j = 1,2,...,v_i\,. \tag{6}$$

Let $Q = \{q : \sum_{i=1}^{n}(\sum_{j=1}^{u_i}(b_{ij}^{1q} - 1)x_{ij}^{\circ 1} + \sum_{j=1}^{v_i}(b_{ij}^{2q} - 1)x_{ij}^{\circ 2}) = 0, q = 1,2,...,h\}$

where $< x_{ij}^{\circ 1}, x_{ij}^{\circ 2} >$ is some solution of the problem **ZC.**

Let of feature number $i = 1,2,...,n$ is fixed and $p = \min\{j : b_{ij}^{1q} = 1, \forall q \in Q\}$,

$r = \min\{j : b_{ij}^{2q} = 1, \forall q \in Q\}$. Let $x_{ip}^{*1} = 1, x_{ij}^{*1} = 0, j \neq p,$

$x_{ir}^{*2} = 1, x_{ij}^{*2} = 0, j \neq r.$ Vector $< x_{ij}^{*1}, x_{ij}^{*2} >$ is defined after performing similar operations for $i = 1,2,...,n$.

Theorem. Vector $< x_{ij}^{*1}, x_{ij}^{*2} >$ is the solution of the problem **Z.**

This theorem provides a basis for creating an algorithm of search of LRC (1) with standard quality criteria:

1. Calculation of the support object.

2. Calculation of sets D_i^1, D_i^2.
3. Problem **ZC** solving and finding solutions {Q, $< x_{ij}^{*1}, x_{ij}^{*2} >$}, {Q} of the problem **Z**.
4. Search of all minimal LRC is carried out by repetition these calculations for all objects of class, taken as support objects.

Currently combinatorial (accurate), relaxation (approximate), and genetic algorithms established for finding LRC of given training table [Kovshov et al., 2008].

4. Processing Sets of LRC

4.1. Construction of LRC of minimum complexity

Let found $P^{\mathbf{c}^1,\mathbf{c}^2}(\mathbf{x})$. Let us consider the logical regularities search equivalent to the $P^{\mathbf{c}^1,\mathbf{c}^2}(\mathbf{x})$, but with minimal complexity. For example, the task is to find equivalent for $P^{\mathbf{c}^1,\mathbf{c}^2}(\mathbf{x})$ the LRC $P^{\Omega_1,\mathbf{c}^1,\Omega_2,\mathbf{c}^2}(\mathbf{x}) = \underset{j\in\Omega_1}{\&}(c_j^1 \le x_j) \underset{j\in\Omega_2}{\&}(x_j \le c_j^2)$, for which $|\Omega_1| + |\Omega_2|$ is minimal.

Consider the next problem of integer linear programming.

$$\sum_{j=1}^{n}(y_j^1 + y_j^2) \to \min,$$

$$\sum_{j=1}^{n}(1-(c_j^1 \le x_{ij}))y_j^1 + \sum_{j=1}^{n}(1-(x_{ij} \le c_j^2))y_j^2 \ge 1, \forall \mathbf{x}_i \notin \tilde{K}_\lambda, \qquad (7)$$

$$y_j^1, y_j^2 \in \{0,1\}.$$

The set of all unit components of the solution $(y_1^1, y_2^1, ..., y_n^1, y_1^2, y_2^2, ..., y_n^2)$ uniquely determines the corresponding subsets of features Ω_1, Ω_2.

4.2. Construction of logical description of the classes of minimum complexity

Let for the class K_λ the set $P_t(\mathbf{x}), t \in T$ was calculated.

Definition 4. The logical description of class K_λ be called logical sum $D_\lambda(\mathbf{x}) = \underset{t\in T}{\vee} P_t(\mathbf{x})$.

Definition 5. The shortest logical description of class K_λ be called logical sum $D_\lambda^s(\mathbf{x}) = \bigvee_{t \in T' \subseteq T} P_t(\mathbf{x})$, where $|T'| \to \min$ and function $D_\lambda^s(\mathbf{x})$ is equal to $D_\lambda(\mathbf{x})$ with given training sample.

Next, we consider that $P_t(\mathbf{x}), t \in T$ is the set of all minimal logical regularities, corresponding LRC found.

Definition 6. The logical sum $D_\lambda^m(\mathbf{x}) = \bigvee_{t \in T^* \subseteq T} P_t(\mathbf{x})$ is called the minimum logical description of the class, in which $\sum_{t \in T^* \subseteq T} (|\Omega_{t1}| + |\Omega_{t2}|) \to \min_{T^*}$, as a function $D_\lambda^m(\mathbf{x})$ the same as $D_\lambda(\mathbf{x})$ in the training sample.

Logic (shortest, minimal) class descriptions are analogous representations of partial Boolean functions in the form of reduced disjunctive normal form (shortest, minimal), and geometric images of logical regularities of classes are analogous to the maximum intervals.

The task of searching the shortest logical descriptions formulated as a problem for covering:

$$\sum_{t \in T} y_t \to \cdot \min, \tag{8}$$

$$\sum_{t \in T} P_t(\mathbf{x}_i) y_t \geq 1, \cdots \forall \mathbf{x}_i \in K_\lambda, \cdot y_t \in \{0,1\}. \tag{9}$$

The task of searching minimal of logic descriptions formulated as a problem for covering with the other functional and a set T:

$$\sum_{t \in T} (|\Omega_{t1}| + |\Omega_{t2}|) y_t \to \cdot \min, \tag{10}$$

with constraints (9).

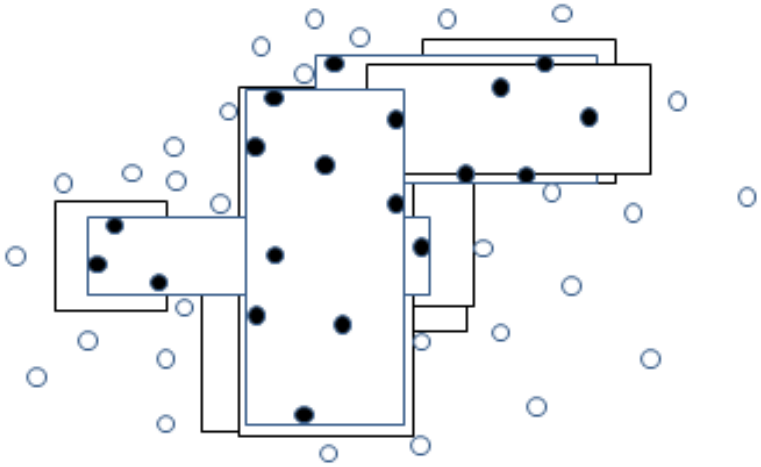


Figure 2. The points of a class are covered with a variety of LRC

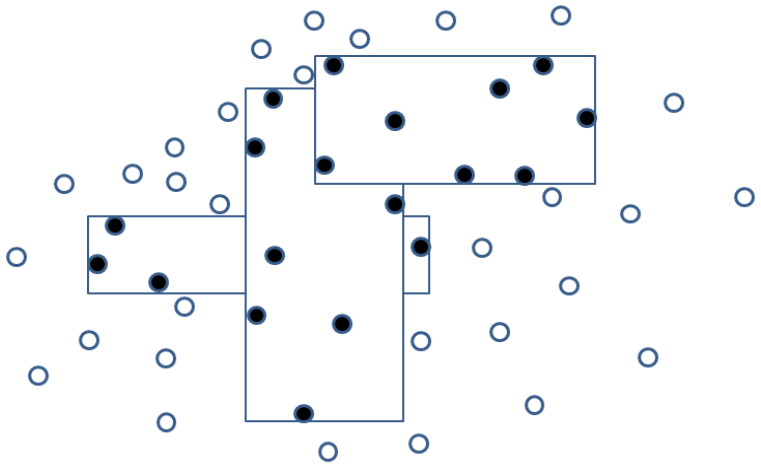


Figure 3. Shortest covering class corresponding to Figure 2

Note that the source can contain a plurality of equal or similar elements, "degenerate" solutions corresponding to the local maximum with small absolute values, the power sets of data can be quite large. At the same time, shortest and minimal logical descriptions are not redundant subsets expressing both the basic properties of data sets and the properties of the classes themselves.

Definition 7. Logical complexity (compactness) of classes are called values:

1. $\psi_1(K_j)$=< *the number of conjunctions in* $D_j^s(\mathbf{x})$>, $j = 1,2,...,l$;
2. $\psi_2(K_j)$=< *the number of variables in* $D_j^m(\mathbf{x})$>, $j = 1,2,...,l$.

The magnitude of $\Phi(X)=\sum_{i=1}^{l}\psi(K_i)$ is called the logical task complexity, if ψ is a criterion of logical complexity of class.

4.3. Processing LRC sets using a cluster analysis

The overall idea presented in [Gupal et al., 2015] and consists of the following. When processing the sets of vectors we can solve the problem of clustering in the 2, 3, ... clusters. For each cluster is calculated its "standard" (for example, the sample mean vector). The resulting system of "standards" is taken as a result of the processing of the initial set of precedent vectors. In this case, the clustering objects are functions LRC. We obtain the new predicates as a result of the clustering of the set LRC and calculations for each cluster, which in general are "partial" LRC. These predicates are evaluated. Thus, we can calculate and estimate a given number of "sufficiently" different partial logical regularities using the initial set of LRC. Figures 4 and 5 are examples. On Figure 4 points of a class (the "black circles" class) covered by system of LRC. Figure 5 shows the same example, but shows only two intervals. Intervals are substantially different and cover mostly large number of points of this class.

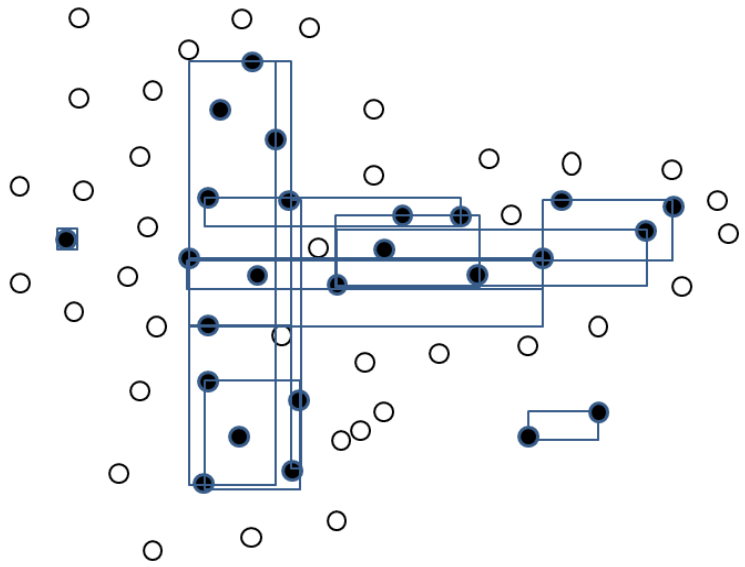


Figure 4. Class points are covered with a large number of intervals

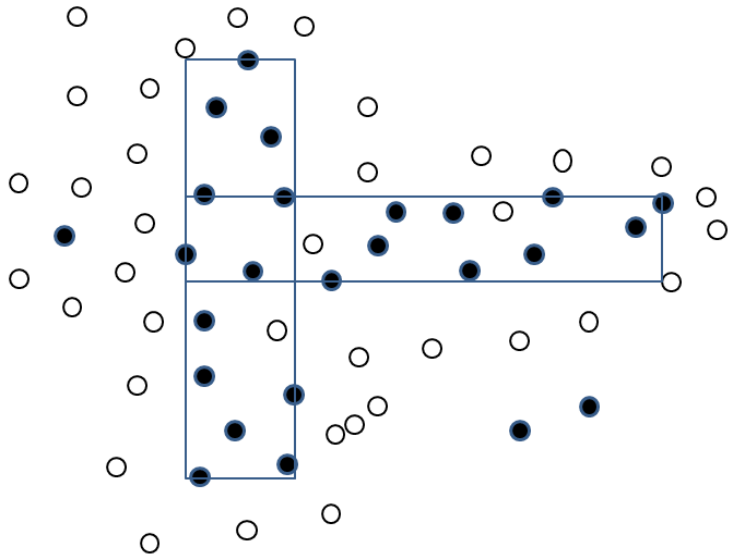


Figure 5. "Substantial" number of class points are covered by two intervals containing perhaps a small number of elements of another class

To implement this idea, we need to create a method of clustering a set of functions (predicates) and to calculate the "standard" for a variety of functions forming the cluster of functions. The set of predicates of each class should be weighted.

We put in a one-to-one correspondence of each $P_i^{\mathbf{c}^1,\mathbf{c}^2}(\mathbf{x})$ from $\mathbf{P}_t$ the binary vector $\mathbf{z}_i$ as follows: $P_i^{\mathbf{c}^1,\mathbf{c}^2}(\mathbf{x}) \Leftrightarrow \mathbf{z}_i = (z_{i1}, z_{i2}, ..., z_{ih}), z_{ij} \in \{0,1\}, j = 1,2,...,h$. Here $h = |K_t|$, vector $\mathbf{z}_i$ marks the original objects of study in the class K_t in which the predicate $P_i^{\mathbf{c}^1,\mathbf{c}^2}(\mathbf{x})$ is equal to one. The weight $y(\mathbf{z}_i)$ of each vector $\mathbf{z}_i$ (and of the corresponding LRC $P_i^{\mathbf{c}^1,\mathbf{c}^2}(\mathbf{x})$) is equal to the share of objects of class K_t, for which this LRC is equal to 1.

So, the initial problem is reduced to the clustering of the set of binary vectors $\mathbf{z}_i = (z_{i1}, z_{i2}, ..., z_{ih})$ with known weights $y(\mathbf{z}_i)$ and calculating the standard of each cluster. As a basic method of clustering we will take a method based on the minimization of variance [Duda et al, 2000].

Let the number of clusters l is fixed. We formulate clustering on l clusters by minimizing the variance criterion as follows:

$$J(\mathrm{K}) = \sum_{i=1}^{l} \sum_{\mathbf{z}_t \in K_i} y_t \left\| \mathbf{z}_t - \mathbf{m}_i \right\|^2 \to \min_{\mathrm{K}} , \tag{11}$$

where $y_t = y(\mathbf{z}_t)$, $\mathrm{K} = \bigcup_{i=1}^{l} K_i, K_i \cap K_j = \varnothing, i \neq j, i, j = 1, 2, ..., l$, $\mathbf{m}_i = \frac{\sum_{\mathbf{z}_t \in K_i} y_t \mathbf{z}_t}{\sum_{\mathbf{z}_t \in K_i} y_t}$.

It can be shown that partition $\mathrm{K} = \{K_1, K_2, ..., K_l\}$ is a local optimal one if (12) is true for all pairs of clusters and for any $\hat{\mathbf{z}}$ of K_i

$$\frac{\sum_{K_i} y\hat{y}}{(\sum_{K_i} y - \hat{y})} \left\| \hat{\mathbf{z}} - \mathbf{m}_i \right\|^2 - \frac{\sum_{K_j} y\hat{y}}{(\sum_{K_j} y + \hat{y})} \left\| \hat{\mathbf{z}} - \mathbf{m}_j \right\|^2 \leq 0 \tag{12}$$

For simplicity of notation, $\sum y$ is used herein instead of $\sum_{\mathbf{z}_t \in K_i} y_t$, $\hat{y}$ - "weight" of vector $\hat{\mathbf{z}}$.

As a result, vectors $\mathbf{m}_i = (m_{i1}, m_{i2}, ...m_{ih})$ are calculated for each cluster. $m_{ij} \in \{0, \alpha_{i1}, \alpha_{i2}, ..., \alpha_{iu}\}, 0 \leq \alpha_{i\sigma} < \alpha_{i,\sigma+1} \leq 1$ are true by construction and the values $\alpha_{i\sigma}$ are calculated from the resulting clustering.

It offers two approaches to processing sets of LRC.

1. We solve the problem of clustering of LRC and vectors $\mathbf{m}_i, i = 1, 2, ..., l$ are calculated.

The vectors $\mathbf{m}_i^*, i = 1, 2, ..., l$ are accepted as a result of processing, which enter a set of the initial $\mathbf{P}_t$ and are closest to the respective $\mathbf{m}_i, i = 1, 2, ..., l$.

2. The standard of each cluster is a Boolean vector $\mathbf{b}_i = (b_{i1}, b_{i2}, ..., b_{ih}), b_{ij} \in \{0, 1\}$, that is a result of sampling the vector $\mathbf{m}_i, i = 1, 2, ..., l$, where $b_{ij} = \begin{cases} 1, & m_{ij} \geq \theta_i, \\ 0, & \text{otherwise} \end{cases}$ Here θ_i is selected from a finite set $D_i = \{0, \alpha_{i1}, \alpha_{i2}, ..., \alpha_{iu}\}$. Vector $\mathbf{b}_i$ corresponds for each choice of θ_i, choice of the optimal vector values is carried out by solving the problem of one-dimensional optimization $\psi(P(\theta_i)) \to \max_{\theta_i \in D_i}$.

There are various quality criteria ψ for partial logical regularities which corresponds to a choice of some θ_i. An example of a criterion $\psi(P(\mathbf{x}))$ may be, for example, the criterion $\psi(P(\mathbf{x})) = \sqrt{k_i} - \sqrt{n_i}$

[William et al., 1999], where k_i is a number of training objects of class K_i, on which the predicate $P(\mathbf{x})$ of this class (corresponding to the chosen θ_i) is performed. n_i is a number of training objects of other classes, where the predicate $P(\mathbf{x})$ is not satisfied. Practical illustration of the method on the data [Mangasarian et al, 1990] is given in [Gupal et al., 2015].

4.4. Informative features, logical correlations and minimization of feature space

Standard statement of recognition problem suggests that the initial information about the classes (training information) is given by sample of vectors of feature descriptions representing all classes. In many cases, the system of features is formed "spontaneously". It includes all parameters influencing the classification (at least hypothetically) and which can be calculated or measured. Regardless of the number of available features, initial system of features is usually the excessive. It may have the features that not affect on the classification. In some practical recognition problems, the calculation of the cost of the features can be significant and compete with the cost of losses for recognition. Solving of training problems with fewer features can also be more precise and the resulting solutions more sustainable. Thus, the solution of problems of feature space minimization is important in many ways.

Let us consider the problem of minimizing the of feature space in the following statement. Let there be a pattern recognition algorithms, the original feature space R^N of feature values $x_1, x_2, ..., x_N$ and quality criterion $f(A)$ of the algorithm *A*. Required to find a subspace of features $R^n, n \le N$ with features $x_{i_1}, x_{i_2}, ..., x_{i_n}$ with minimal n $(n \le N)$, for which $f(A) \ge f_0$, where f_0 is a some minimum acceptable accuracy of the recognition algorithm *A*, built according to the training data for the subspace [Vetrov et al, 2001].

Due to its combinatorial nature, methods of enumeration a large number of different feature subspaces are practically unrealizable, so sequential selection procedures of the features systems as subsystems of k from the k-1 feature are commonly used.

The problem of minimizing of feature space was considered for recognition models based on the voting on systems of the logical regularities.

Let P be a set of all minimal LRC of minimum complexity that was found from the training data, $N = |\mathrm{P}|$.

Definition 8. The value $wei(i) = N(i)/N$ is called the measure of informativity of feature $№i$ if $N(i)$ is the number of elements of the P containing the feature $№i$.

Let $N(i,j)$ is the number of simultaneous occurrences of features into one set of LRC. $Lcorr(i,j) = 1 - \dfrac{N(i,j)}{\min(N(i), N(j))}$ is called the logical correlation of features $№i$ and $№j$. We believe $Lcorr(i,j) \equiv 0$ if ***min***($N(i)$, $N(j)$)=*0,* because of a characteristics (i or/and j) "does not depend on" (this case arises, for example, if $x_i \equiv const$).

Consider the problem of finding clusters of features that have close correlation properties.

As a clustering algorithm for a given semimetric $r(i,j)$ (was used logical correlation) and a fixed number of classes has been used clustering procedure "hierarchical grouping" in which the distance between the clusters determined by the function

$$r(K_p, K_q) = \max_{i \in K_p,\, j \in K_q}(r(i,j)).$$

After finding the n clusters, the condensed subsystem of features includes the most informative initial features (no more than one from each cluster). As $r(K_p, K_q)$ the function $1 - Lcorr(i,j)$ was used.

Figure 6 shows the variations of recognition accuracy of recognition models at two approaches to minimize the feature space on the example of the state of the ionosphere recognition problem [Sigillito et al., 1989]. Here, the black line represents the consistent screenings of less informative features, the gray line corresponds to minimizing the feature space according to the proposed algorithm in this paper. It is seen that the gray line is usually lower than the black.

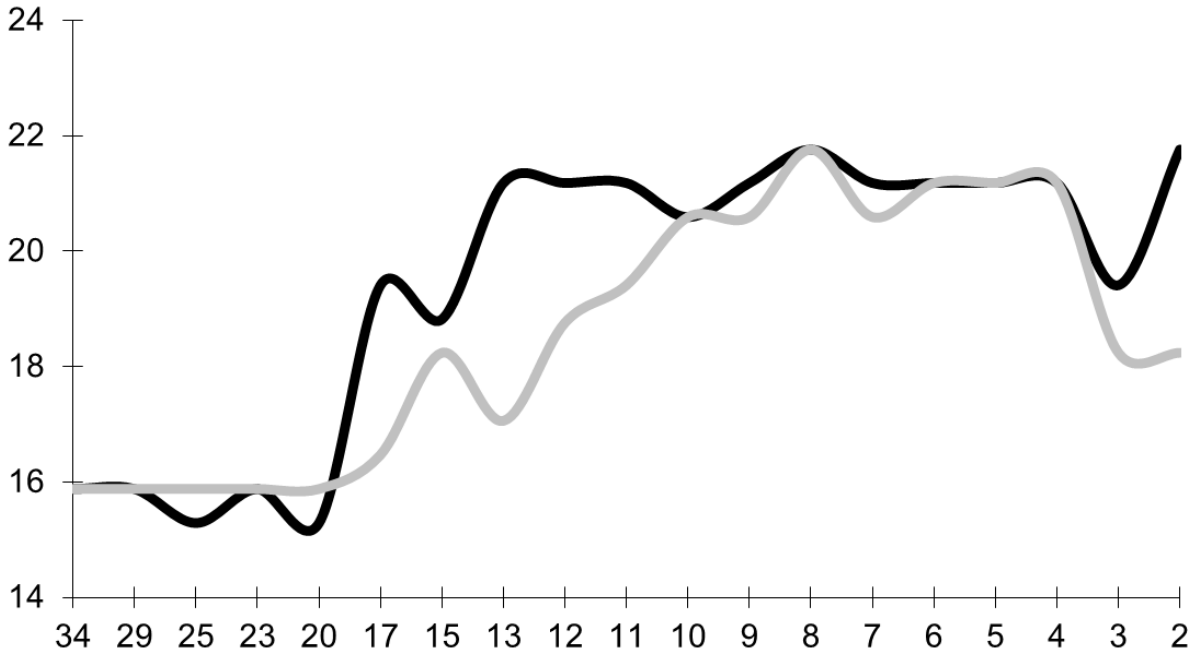


Figure 6. The dependence of the error rate of the number of features

4.5. The supervised classification procedures based on LRC

Calculation of estimates of the objects as a result of a simple voting on the found sets of LRC can lead to unsatisfactory results in recognizing of new objects [Lviv et al, 2015]. A similar effect can often be resolved by introducing the negatives LRC (which is equivalent to the use of " anti proximity") and the approximation of LRC by sigmoid functions.

Later, the case of two classes will be considered for simplicity. Suppose that the logical regularities $P_i^1(\mathbf{x}), i=1,2,...,m(1)$ are found for the first class and $P_i^2(\mathbf{x}), i=1,2,...,m(2)$ are found for the second class. Then estimation for the first class will be calculated by the formula $\Gamma_1(\mathbf{x}) = \sum_{i=1,2,..,m(1)} \alpha_i^1 P_i^1(\mathbf{x}) + \alpha_0^1 \overline{\bigvee_{i=1,2,..,m(2)} P_i^2(\mathbf{x})}$, and estimation for the second class will be calculated by the formula $\Gamma_2(\mathbf{x}) = \sum_{i=1,2,..,m(2)} \alpha_i^2 P_i^2(\mathbf{x}) + \alpha_0^2 \overline{\bigvee_{i=1,2,..,m(1)} P_i^1(\mathbf{x})}$. We will use a simple decision rule. Then the object classification will be on a sign of the following function:

$$f(\mathbf{x}) = \sum_{i=1,2,..,m(1)} \alpha_i^1 P_i^1(\mathbf{x}) + \alpha_0^1 \overline{\bigvee_{i=1,2,..,m(2)} P_i^2(\mathbf{x})} - \sum_{i=1,2,..,m(2)} \alpha_i^2 P_i^2(\mathbf{x}) - \alpha_0^2 \overline{\bigvee_{i=1,2,..,m(1)} P_i^1(\mathbf{x})}, \quad (13)$$

Here $P_i^1(\mathbf{x}), i = 1,2,\ldots,m(1)$ and $P_i^2(\mathbf{x}), i = 1,2,\ldots,m(2)$ are the logical regularities LRC of the first and second classes respectively, α_0^1, α_0^2, α_i^1, α_i^2 are the weighting coefficients. Object $\mathbf{x}$ belongs to the first class if $f(\mathbf{x}) > 0$ and belongs to the second class if $f(\mathbf{x}) < 0$. When $f(\mathbf{x}) = 0$ occurs a failure on the recognition or random classification.

Construction of the function $f(\mathbf{x})$ can be regarded as successive solution of two tasks:

1. Calculation of LRC $P_i^1(\mathbf{x}), i = 1,2,\ldots,m(1)$ and $P_i^2(\mathbf{x}), i = 1,2,\ldots,m(2)$, and the transition to the new $m(1)+m(2)+2$ - dimensional feature space of their values and the corresponding disjunction negation (with the sign "+" for the first class and "-" for the second).

2. Search of weighting coefficients by calculating a new feature space and separating hyperplane using the linear methods, for example, "Support Vector Machines", "Linear machine" or "Fisher's linear discriminant." We note that in the new feature space objects of the first class of training sample will correspond to the vectors of the form $\mathbf{y} = (\sigma_1, \sigma_2, \ldots, \sigma_{m(1)}, 1, 0, 0, \ldots, 0), \sigma_t \geq 0, \sum_{t=1}^{m(1)} \sigma_t > 0$. Objects of the second class are $\mathbf{z} = (0, 0, \ldots, 0, -\theta_1, -\theta_2, \ldots, -\theta_{m(2)}, -1), \theta_t \geq 0, \sum_{t=1}^{m(2)} \theta_t > 0$. So, classes are linearly separable in a given space. Should be noted that in the new feature space is possible to use other models.

4.6. Evaluation of outliers based LRC

Suppose that the shortest logical description $D_\lambda^s(\mathbf{x}) = \bigvee_{t \in T' \subseteq T} P_t(\mathbf{x})$ was calculated $D_\lambda^s(\mathbf{x}) = \bigvee_{t \in T' \subseteq T} P_t(\mathbf{x})$ for given training sample. Let $f(\mathbf{x}_i) = \sum_{t \in T' \subseteq T} y_t P_t(\mathbf{x}_i)$, where y_t is the weight of the corresponding logical regularity (eg, the number of objects that satisfy the given LRC). The values $f(\mathbf{x}_i)$ are ordered by an increase and normalize (for example, so that $\sum_{\mathbf{x}_i \in K_j} f(\mathbf{x}_i) = 1$). Minimum values of $f(\mathbf{x}_i)$ will be in accordance with the most unusual objects.

5. Conclusion

In the beginning we only have the training set. As a result of discrete analysis, we find sets of logical regularities (LRC) for each class. In fact, we find the conjunctions of intervals of attributes changes that characterize any class and does not hold for other classes of training objects. Found LRC are of independent interest for the practical user. What is the class? Previously, every class we were identifying with a set of its representatives. Now we can say that each class is characterized by a variety of some LRC (a variety of knowledge). It should be noted that we do not use any metric properties of objects. Features may be ordinal. Knowledge of informative features is not required. On the contrary, they can be estimated using the found sets of LRC. The article contains numerous possible applications of found sets for pattern recognition tasks. Further research in this area require the presence of cases of missing data, the linear relationships between variables, construction of optimal recognition procedures based on the found sets of LRC.

6. Acknowledgements

This work was supported by the Program of the Presidium of RAS №15 «Information, control and intelligent technologies and systems", Program №2 of Mathematical Sciences Department of RAS, RFBR № 14-01-00824_a, 15-01-05776_a, 15-51-05059 Arm_a.

Bibliography

[Zhuravlev, 1978] Yu.I. Zhuravlev. On the algebraic approach to solving the problems of recognition and classification. Problems of Cybernetics. M .: Nauka, 1978. Issure.33. pp. 5-68.

[Zhuravlev et al, 2006] Yu. I. Zhuravlev, V. V. Ryazanov, and O. V. Sen'ko, Recognition. Mathematical Methods. Program System. Practical Applications. Izd.vo "Fazis", Moscow, 2006,178 pp.

[Ryazanov, 2007] V.V.Ryazanov. Logical regularities in pattern recognition (parametric approach). Journal of Computational Mathematics and Mathematical Physics, T.47, №10, 2007, pp.1793-1808.

[Kovshov et al., 2008] N.V.Kovshov, V.L.Moiseev, V.V.Ryazanov. Algorithms for finding logical regularities in pattern recognition. Journal of Computational Mathematics and Mathematical Physics, T.48, 2008, N 2, pp. 329-344.

[Gupal et al., 2015] Anatoliy Gupal, Maxim Novikov, Vladimir Ryazanov. Processing sets of classes' logical regularities. International Journal "Information Theories and Applications", Vol. 22, Number 1, 2015, pp. 39-49.

[Duda et al, 2000] Duda, R. O., Hart, P. E., and Stork, D. G. : Pattern Classification. John Wiley and Sons. 2nd edition (2000).

[William et al., 1999] William W. Cohen and Yoram Singer Simple, Fast, and Effective Rule Learner, AAAI/IAAI 1999: 335-342.

[Mangasarian et al, 1990] Mangasarian O. L., Wolberg W.H.: "Cancer diagnosis via linear programming", SIAM News, Volume 23, Number 5, September 1990, pp 1 - 18.

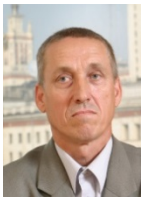
[Vetrov et al, 2001] D.P. Vetrov, V.V. Ryazanov. The minimization of feature space in pattern recognition. Reports of the 10 th All-Russian Conference "Mathematical Methods of Pattern Recognition (MMRO-10)", Moscow, 2001, 22-24.

[Sigillito et al., 1989] Sigillito, V. G., Wing, S. P., Hutton, L. V., \& Baker, K. B. (1989). Classification of radar returns from the ionosphere using neural networks. Johns Hopkins APL Technical Digest, 10, 262-266

Authors' Information

Yury Zhuravlev *– Deputy Director of Dorodnicyn Computing Centre, Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, Russia, 119991 Moscow, Vavilov's street, 40; e-mail: zhur@ccas.ru*

Major Fields of Scientific Research: Discrete mathematics, Algebra, Pattern recognition, Data mining, Artificial Intelligence



Vladimir Ryazanov *– Heard of Department; Dorodnicyn Computing Centre, Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, Russia, 119991 Moscow, Vavilov's street, 40; e-mail: rvv@ccas.ru, rvvccas@mail.ru*

Major Fields of Scientific Research: Pattern recognition, Data mining, Artificial Intelligence

TOWARDS A DAWKINS' GENETIC ALGORITHM: TRANSFORMING AN INTERACTIVE EVOLUTIONARY ALGORITHM INTO A GENETIC ALGORITHM

S. Guil López, P. Cuesta Alvaro, S. Cano Alsua, R. Salas Machado,

J. Castellanos, R. Lahoz-Beltra

Abstract*: This paper addresses the transformation of an interactive evolutionary algorithm - the biomorphs model introduced by Dawkins in his book "The Blind Watchmaker" - in a genetic algorithm. One of the critical steps was the substitution of the evaluation made by a human by a fitness function. In addition we studied two experimental situations: (i) biomorphs populations with only mutation and (ii) populations including a crossover operator. In both cases, significant evolutionary differences are observed, classifying individuals in different classes according to their genotypic frequencies. Finally, in order to assess whether there is an influence of computer "hardware" in the simulated evolution, experiments were performed on a conventional computer and using a supercomputer. Surprisingly the results obtained allow us to conclude that the type of computer used has no significant effect on evolutionary computation experiments, as long as evolution is the result of Darwinian natural selection.*

Keywords*: interactive evolutionary algorithm, Dawkins' biomorph program, artificial life, evolutionary computing on supercomputers*

ACM Classification Keywords*: I.6 Simulation and Modeling*

Introduction

One of the most popular evolutionary models in Artificial Life was introduced in 1986 by Richard Dawkins [Dawkins, 1986] in the book entitled "The Blind Watchmaker". The model introduced by Dawkins mimics the evolution of individuals termed as *biomorphs* and is inspired by the principle of continuity of 'germ plasm' introduced in 1893 by Weismann. The aim of the model was the simulation of the Darwin's principle of natural selection [Lahoz-Beltra, 2008] in individuals with asexual reproduction [Lahoz-Beltra, 2004]. Therefore, the mutation is the only source of variability and the mechanism for evolution is the familiar principle of cumulative selection. In the Dawkins' original model is the researcher - playing the role of a "blind watchmaker" - who selects in accordance with their own criteria (e.g. the largest, smaller, beautiful etc.) the optimum biomorph in every generation. One of the goals of the present paper was to overcome these limitations, transforming the algorithm introduced by

Dawkins, thus an interactive evolutionary algorithm, in a genetic algorithm (GA) [Guil Lopez, 2000]. Interactive evolutionary algorithms use human evaluation as a fitness function [Milani, 2004], e.g. aesthetic selection based on biomorph attractiveness, being Darwinian selection the result of human preferences. In this paper we substitute the evaluation made by a human by a fitness function. Also we studied the role of genetic crossover in evolution, building an evolutionary algorithm (EA without crossover) and a genetic algorithm (GA including crossover). Therefore, from a biological point of view, the genetic algorithm simulates a population of biomorphs with sexual reproduction whereas the evolutionary algorithm emulates a population of biomorphs with asexual reproduction. Simulation experiments with EAs and GAs were performed on a personal computer and a supercomputer SGI-Cray ORIGIN 2000 in order to figure out whether the evolution of a population of biomorphs depends on the type of hardware, i.e. the 'test tube', where the evolution experiments were conducted. The experiments described in this paper are part of the work done during several years of work [Guil Lopez, 2000] about *bio-inspired evolutionary algorithms* [Lahoz-Beltra, 2008; Perales-Gravan et al., 2013; Thai Dam and Lahoz-Beltra, 2014]. At present we continue working on this long term research project.

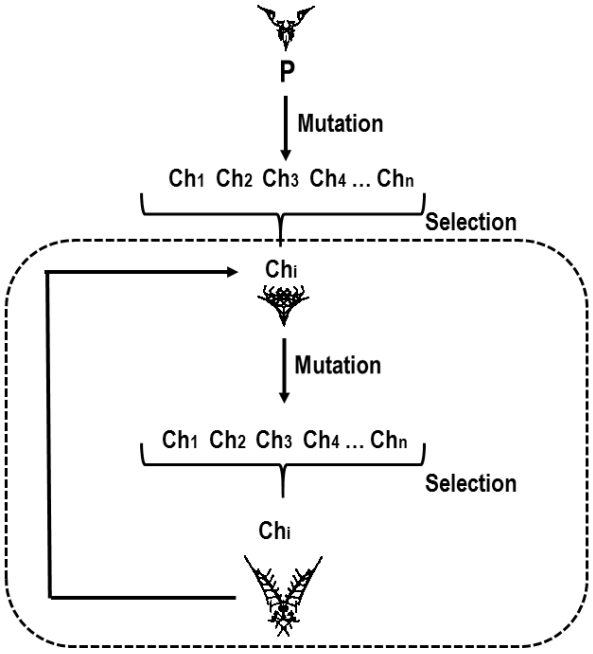


Figure 1. Dawkins' original algorithm [Dawkins, 1986]. A parental individual (P) will produce an offspring of n 'children' (Ch). The children are similar to P, except for the value of a gene which has changed by mutation. A human selects according to their judgment the 'best biomorph' (Chi) which will produce a new offspring of n children. This loop is repeated again and again until we decide to conclude the evolutionary cycle.

From the Dawkins' algorithm to a Dawkins genetic algorithm

In the original model introduced by Dawkins (Figure 1) an individual or biomorph is represented by a one-dimensional array or vector which simulates a chromosome. A position on chromosome simulates a gene, being chromosomes constituted by eight genes {$gene_1$, $gene_2$,…, $gene_8$} whose values are positive integers, i.e. $gene_i \in \mathbb{Z}^+$. The genes values codify for the biomorph shape or phenotype as defined in Table 1. From a genetic point of view the phenotypic expression of a character can be the result of the sum of several genes, e.g. branching length ***L*** (gene 3 + gene 4), denominating these genes as polymers factors or cumulative factors. In other cases some genes multiply their effects as it occurs again with the branching length ***L*** (gene 3 * gene 4). The model also includes epistasis, thus the interaction of genes affecting the same character, modifying as a consequence the Mendelian segregation (Mendel's genetic laws). For instance, in the present model the genes encoding branches angles, i.e. genes 6 and 7, have an effect on the character "branching length" that is encoded by genes 3, 4 and 5. Figure 1 shows the recursive algorithm introduced by Dawkins. In this paper we studied populations without crossover or asexual reproduction (Figure 2) and populations with crossover or sexual reproduction (Figure 3), but substituting in both cases the human selection by a selection operator. The aim of the *selection operator* is the Darwinian selection of biomorphs from which the next generation will be obtained. The fitness of each individual was calculated as follows.

Table 1. Biomorph's chromosome or genotype ($gene_1$, $gene_2$,…, $gene_8$)

Gene 1: Number of offspring individuals
Gene 2: Number iterations to draw a biomorph
Genes 3 and 4: Numbers encoding for the length of the branches
Gene 5: Length of a branch ***L*** is defined according to the following algorithm

$$L = \begin{cases} \text{Gene 3 + Gene 4} & \text{, if Gene 5 is an even number} \\ \text{Gene 3 * Gene 4} & \text{, if Gene 5 is an odd number} \end{cases}$$

Gene 6: Angle of the first branching
Gene 7: Angle between the second and first branching
Gene 8: Length of the body

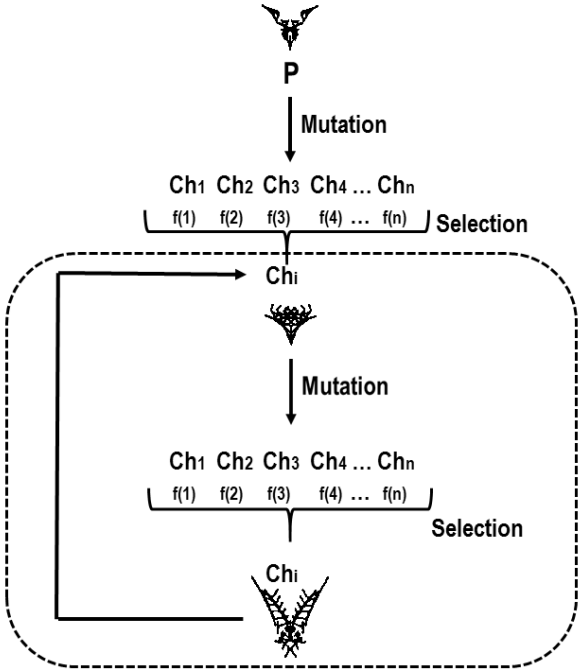


Figure 2. Dawkins' evolutionary algorithm. 'Human evaluation' is substituted by a fitness function. Note that no recombination or crossover mechanism is present. Therefore from a biological point of view we would be simulating a population with asexual reproduction.

First, we define the best individual, thus the optimal genotype or chromosome, referred as 'target biomorph'. The choice of a target individual was performed by applying one or other of the following criteria. A first protocol is the *Criterion of Equitable Distribution*. According to this criterion the Euclidean distance between the target individual and the parental individual is similar for the two parental individuals, being in the simulation experiments equal to 9.85 units. In addition, the selected genotype is obtained as follows: 50% is obtained by mutation and the remaining 50% results from the recombination or crossover of parental individuals. In the experiments conducted with crossover or sexual reproduction (Figure 3), in the initial generation (t = 0) the parental genotypes ($gene_1$, $gene_2$,…, $gene_8$) were (11, 0, 1, 2, 4, 0, 92, 10) and (16, 0, 3, 7, 6, 0, 95, 13). Likewise, the genotype of the target biomorph was (16, 5, 6, 2, 1, 2, 95, 10). The second protocol was termed *Restrictive Criterion*. In addition, fulfilled the above assumption with Euclidean distance (the distance between the target individual and the two parental individuals is equal to 9.85), in the conducted simulation experiments the maximum values of the first four genes in the genotype should be equal to a set of values: 16, 10, 8 and 8. The other values of genes are chosen within the interval [a+10, b-10], where a and b values are the higher and lower of each gene value in the parental biomorphs. According to this criterion we selected the following target biomorph (16, 5, 6, 2, 1, 2, 95, 10).

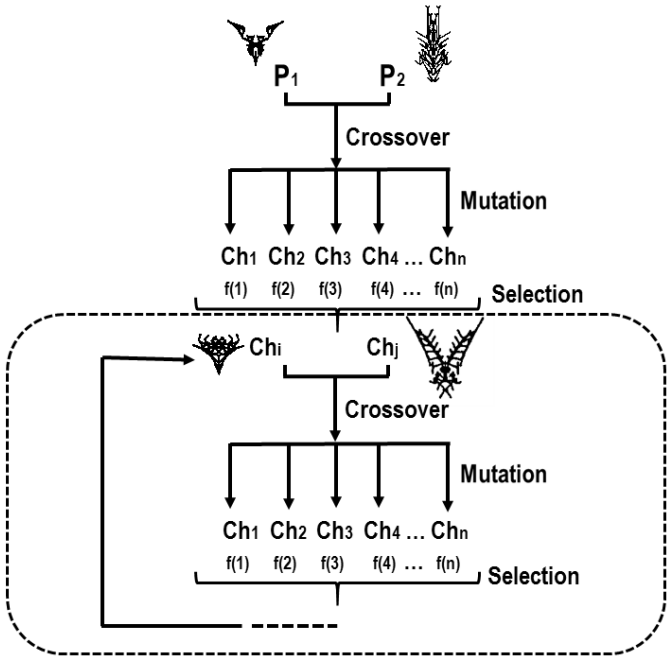


Figure 3. Dawkins' genetic algorithm. 'Human evaluation' is substituted by a fitness function. Since there is crossover, we would be simulating a population with sexual reproduction. Accordingly, Darwinian selection will select a pair of parental individuals: Chi and Chj.

Since in the experiments without crossover or asexual reproduction (Figure 2) there is only one parental individual at t = 0, all simulation experiments were performed with the initial genotype (16, 0, 3, 7, 6, 0, 95, 13).

Secondly, we calculate a 'genetic distance' (1) , i.e. the Euclidean distance d_i between the genotypes of biomorph i ($gene_i$) and target biomorph ($gene_i^{target}$):

$$d_i = \sqrt{\sum_i \left(gene_i - gene_i^{target} \right)^2} \qquad (1)$$

After calculating distance (1) the obtained value d_i will be the first argument of the fitness function. In this paper **F1-F4** (Table 2) are the proposed evaluation or fitness functions that we will use to transform Dawkins' algorithm into a true genetic algorithm. Given a fitness function and a set of distances d_i from individuals of the same generation, we obtain the minimum distance d_{min}, thus the second argument of the fitness function.

Simulation experiments were conducted based on the following fitness function parameters: $\alpha = 0.1$, $\beta = 0.9$ and $\gamma = 0.2$, where the *k* value was equal to zero (*k* = 0) for populations with a maximum of 500 individuals. Otherwise, *k* = 300 when the population size exceeded 500 individuals. Once we obtained the fitness value of each individual, the selection of parental individuals was conducted by a Bernoulli roulette [Lahoz-Beltra, 2004]. According to this procedure, the selection probability of an individual (2) represented by its chromosome *i* is calculated as the ratio between the individual fitness and total fitness:

$$p(i) = \frac{f(x_i)}{\sum_i f(x_i)} \tag{2}$$

Table 2.- Objective or fitness functions

F1	$f(\mathrm{d}_{\min}, \mathrm{d}_i) = \begin{cases} \dfrac{\alpha + d_{\min}}{\gamma - \beta(\alpha + d_{\min})} - \alpha(\alpha + d_{\min}) + k & , \quad d_i = 0 \\ \dfrac{\alpha + d_{\min}}{d_i - \beta(\alpha + d_{\min})} - \alpha(\alpha + d_{\min}) + k & , \quad d_i \neq 0 \end{cases}$
F2	$f(\mathrm{d}_{\min}, \mathrm{d}_i) = \begin{cases} \dfrac{d_{\min}}{\gamma - \beta d_{\min}} - \alpha d_{\min} & , \quad d_i = 0 \\ \dfrac{d_{\min}}{d_i - \beta d_{\min}} - \alpha d_{\min} & , \quad d_i \neq 0 \end{cases}$
F3	$f(\mathrm{d}_{\min}, \mathrm{d}_i) = \left\| \left(\dfrac{d_{\min}}{d_i - \beta d_{\min}} - \alpha d_{\min} \right) \right\|$
F4	$f(\mathrm{d}_{\min}, \mathrm{d}_i) = \begin{cases} \dfrac{d_{\min}}{\gamma - \beta d_{\min}} - \alpha d_{\min} + k & , \quad d_i = 0 \\ \dfrac{d_{\min}}{d_i - \beta d_{\min}} - \alpha d_{\min} + k & , \quad d_i \neq 0 \end{cases}$

Finally, given a random number U_i between 0 and 1, the individual or chromosome i is selected (R=1) for the next generation if it holds that:

$$\begin{cases} p(i) < U_i \,, R = 0 \\ p(i) \geq U_i \,, R = 1 \end{cases} \tag{3}$$

In the simulation experiments with populations displaying crossover, i.e. sexual reproduction (Figure 3), the *recombination operator* was simulated with one and two cutting points, as is usual in genetic algorithms. Regarding *mutation operator*, and since the chromosome is a vector of positive integers, it was not possible to use the flip-bit method [Lahoz-Beltra, 2004]. Mutation was simulated as follows. First, a gene is chosen randomly, increasing or decreasing its value by one. Obviously the sense of mutation, increase or decrease, is also conducted at random. Secondly, mutation is applied taking into account the allowable range of values of the genes. For example, in the simulation experiments genes 1, 2, 3 and 4 have maximum values equal to 16, 10, 8 and 8 respectively.

Simulation experiments

This study has been carried out for several years studying under different types of computer hardware (Tables 3-4) the evolution of biomorph populations. Populations were simulated based on two different versions of the Dawkins' algorithm (Figures 2 and 3). The first simulation experiments were conducted in the late 90s using an IBM PC-compatible computer with Pentium III processor 450 MHz and 218 MB of RAM. The algorithms depicted in Figures 2 and 3 were written at that time in TurboPascal 7.0. Preliminary experiments were performed with the program BIOMURFFS: a program showing the Dawkins' original algorithm (Figure 1) written in QBASIC for recreational purposes [Prata, 1993]. The simulation experiments (Table 3) were performed testing the fitness function **F1** (Table 2) with populations of 16 biomorphs, simulating the evolution during 500 generations. In each generation the offspring came from a single individual or a couple of individuals depending on whether the algorithm simulates asexual (Figure 2) or sexual (Figure 3) reproduction. In the latter case, and therefore when biomorphs display crossover, recombination was simulated with one and two cutting points, choosing different recombination probability values (0.25, 0.50 and 0.75).

Table 3.- Experimental conditions in the simulations conducted with IBM PC-compatible computer

	Standard population	
	Asexual reproduction	Sexual reproduction
Population size	16	16
Number parental individuals	1 individual	1 couple
Number of generations	500	500

Some years later, a second batch of simulation experiments was performed on a SGI-Cray ORIGIN 2000 supercomputer with 40 MIPS R10000 250 MHz microprocessors and 16 MIPS R12000 400 MHz microprocessors, with 12 GB of RAM and 190 GB hard drive. Of course, today many of these features have already been overcome but the results are still valid. Once again the two different versions of the Dawkins' algorithm (Figures 2 and 3) were implemented but this time in C language, and compiled with the MIPSproC compiler under IRIS 6.5 OS. The compilation was made with the option "-Ofast" in order to minimize runtime, using a C mathematical library implemented for the IRIS system. After several preliminary experiments we decided to evaluate the biomorphs fitness using **F4** function. Experiments with populations displaying sexual reproduction were performed with a two-point crossover operator and probability equal to 0.5. Simulation experiments were carried out increasing the population size, the number of parental biomorphs per generation and the number of simulated generations (Table 4).

Table 4.- Experimental conditions in the simulations conducted with SGI-Cray ORIGIN 2000 supercomputer

	Standard population				Increased population			
	Asexual reproduction		Sexual reproduction		Asexual reproduction		Sexual reproduction	
Population size	16		16		160		160	
Number parental individuals	1 individual		1 couple		10 individuals		10 couples	
Number of generations	1000	1500	1000	1500	1000	1500	1000	1500

Results

Despite the time that has elapsed since we conducted the experiments, in our opinion the results are still perfectly valid. In fact currently we are developing a quantum genetic algorithm version of the Dawkins' original algorithm. The results obtained under the original Dawkins' standard population conditions, i.e. a population size equal to 16 (Table 3 and 4), were similar in the two kinds of computers, i.e. in the experiments conducted with IBM PC-compatible computer and SGI-Cray ORIGIN 2000 computer. Therefore, we can conclude that for these experimental conditions has no influence the type of computer used to run the simulations and the number of studied generations (500, 1000 or 1500). However, the evolutionary convergence differs depending on the type of reproduction. Evolutionary convergence was analyzed for each biomorph plotting the genetic distances: on the y-axis from the parental biomorph and the x-axis from the target biomorph. In the experiments with asexual reproduction (Figure 4 a, c) populations exhibit higher evolutionary fluctuations, requiring a greater number of generations to achieve the target genotype. That is, sexually reproducing populations (Figure 4 b, d) reach the target genotype in a smaller number of generations. If we compare the evolutionary convergence of both types of reproduction with a card player, e.g. in the game of blackjack, the biomorphs with asexual reproduction would be "conservative players" avoiding a maximum adaption to their environment. Of course, in this metaphor the environment will be the casino where life is brought into play. By contrast, the biomorphs with sexual reproduction would be "risky players" showing a maximum adaptation to their environment. In both cases and for small values of genetic distances (near to the optimal genotype or target values), we observed a cluster of genetic distances which we have termed “arrowhead” (Figure 4 a-d).

However, when experiments were conducted under large population conditions (Table 4) some remarkable differences were observed. One of the most striking differences is the absence of an "arrowhead" cluster (Figure 4 e, f). Likely this result could be explained considering that in those populations with asexual reproduction the offspring is obtained by ‘bipartition’ of 10 parental individuals. Similarly, the explanation could be the same in those populations with sexual reproduction where the offspring is obtained by ‘sexual intercourse’ in 10 couples. Moreover, although we have increased the number of generations, populations do not reach the target genotype. Consequently, in large populations a lower evolutionary convergence might be due to a greater variability of individuals.

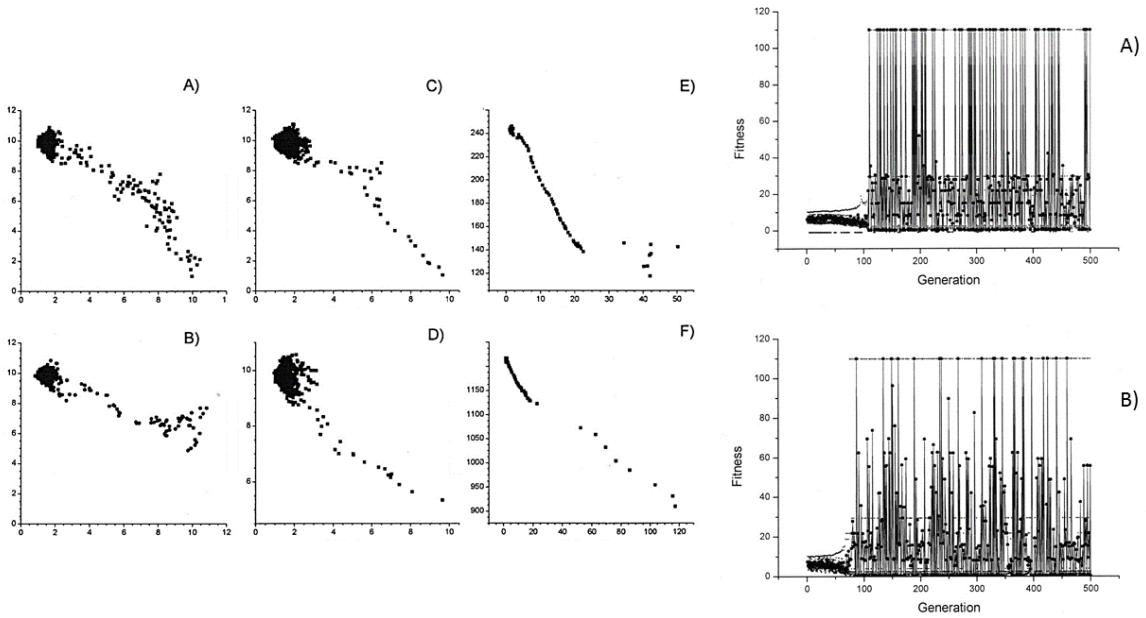


Figure 4.- Simulation experiments.

(Left) Evolutionary convergence plot (genetic distance from the parental biomorph=y-axis, genetic distance from the target biomorph=x-axis). For explanation see text.

(Right) Performance graph obtained in the simulation experiments (A) Dawkins' evolutionary algorithm (B) Dawkins' genetic algorithm

The performance graph [Lahoz-Beltra and Perales-Gravan, 2010] was similar in the two batches of experiments, the experiments conducted with IBM PC-compatible and SGI-Cray ORIGIN 2000 computers. However, the graphical representation of the average fitness and individual fitness per generation (Figure 4) depends on whether the reproduction is asexual (Figure 2) or sexual (Figure 3). For both kinds of reproduction we observed a first stage with low fitness value: from generation 1 to 108 in the populations with asexual reproduction and from 1 to 83 in populations with sexual reproduction. From 109 and 84 generations, depending on whether the reproduction is asexual or sexual, begins a second stage. In this stage individuals reach the target genotype (genetic distance near o equal to 0) or individuals are very close to the target genotype (the average fitness value is 109.89 units or equivalently a genetic distance equal to 1 unit). Nevertheless, variability is greater in the bimorphs with sexual reproduction compared to those with asexual reproduction. Moreover, if we consider the individual fitness value in both types of reproduction, we conclude the formation of genotypic classes or

clusters of individuals throughout generations. An analysis of the genotypic frequency of individuals per class, allows us to conclude that the observed number of clusters is greater in populations with sexual reproduction than asexual reproduction. In fact the number of categories or clusters increases with the variability and the number of simulated generations: 12, 23, 197, 223, 1796 and 2407 clusters for the case in which we studied asexual reproduction – 500 generations – 1 parental individual (Figure 5a), sexual reproduction – 500 generations – 1 couple parental individuals (Figure 5b), asexual reproduction – 1000 generations – 1 parental individual (Figure 5c), sexual reproduction – 1000 generations – 1 couple parental individuals (Figure 5d), asexual reproduction – 1000 generations – 10 parental individuals (Figure 5e) and sexual reproduction – 1000 generations – 10 couples parental individuals (Figure 5f), respectively. Assuming that the distribution of genotypic frequencies is multinomial, and confining the study to those experiments with 500 generations, we have e.g. that in the case of asexual reproduction with a single parent, the following frequency distribution:

$$f(x_1,x_2,x_3,x_4,x_5,x_6)=\frac{n}{x_1!\ x_2!\,x_3!\,x_4!\,x_5!\,x_6!}\left(\frac{1496}{5465}\right)^{x_1}\left(\frac{1389}{5465}\right)^{x_2}\left(\frac{327}{5465}\right)^{x_3}\left(\frac{0}{5465}\right)^{x_4}\left(\frac{919}{5465}\right)^{x_5}\left(\frac{1334}{5465}\right)^{x_6} \quad (4)$$

whereas for experiments with sexual reproduction and one couple of parents, we obtained the following frequency distribution:

$$f(x_1,x_2,x_3,x_4,x_5,x_6)=\frac{n}{x_1!\ x_2!\,x_3!\,x_4!\,x_5!\,x_6!}\left(\frac{1453}{5465}\right)^{x_1}\left(\frac{1043}{5465}\right)^{x_2}\left(\frac{189}{5465}\right)^{x_3}\left(\frac{720}{5465}\right)^{x_4}\left(\frac{440}{5465}\right)^{x_5}\left(\frac{1334}{5465}\right)^{x_6} \quad (5)$$

The study of above frequency distributions (4, 5) allows us to obtain the following conclusions. Firstly, it is noteworthy that there are certain classes that appear with high frequency regardless of type of reproduction. Secondly, when simulations are conducted with asexual reproduction there are more classes with zero frequency compared with those experiments with sexual reproduction. Finally, and for both types of reproduction, the classes with higher frequencies are those that contain a high number of individuals with high fitness values.

One of the achievements of this work was to introduce a fitness function (Figure 6) in the original Dawkins' evolutionary algorithm [Guil Lopez et al., 2000]. The aim was to replace the evaluation performed by a human (aesthetic selection) by a selection operator, transforming the Dawkins

interactive evolutionary algorithm into a true genetic algorithm. **F1** function was the first fitness function proposed in this study, evaluating biomorphs in the simulation experiments conducted with IBM PC-compatible. This function was a mathematical expression that was obtained from the original expression shown in Figure 6. In **F1** function and the remaining fitness functions we considered the case where the genetic Euclidean distance (1) is equal to or different from zero, avoiding the emergence of a 'dominant chromosome' with a disproportionately high fitness value. Parameters α and β were defined in the function with the following purpose. For large values of the genetic distance the α parameter included into the negative term of the fitness function decreases the fitness value. However the parameter plays the opposite role in the denominator of the fitness function. Another effect to correct occurs when small value of the Euclidean distance causes a large increase in the fitness value. Parameter β was introduced in order to moderate the increase in the fitness value. Finally, *k* term was introduced in order to ensure that the function is always positive, because when the population size increases also increases the minimum distance (d_{min}). **F2**, **F3** and **F4** fitness functions are very similar to **F1** function and were used to evaluate the biomorphs in the simulation experiments conducted with SGI-Cray ORIGIN 2000 supercomputer. **F2** displays a trouble taking negative values for large Euclidean distances. For this reason **F3** was introduced, returning the absolute value of **F2**. Finally, all simulation experiments with the supercomputer were performed with **F4**.

Conclusion

We show an example of how to transform an interactive evolutionary algorithm, i.e. the biomorphs model introduced by Dawkins, in a genetic algorithm. One of the critical steps was the search for a fitness function that measures the 'physical quality' of individuals. In addition we studied two experimental situations - biomorphs populations with only mutation (asexual reproduction) and populations including crossover (sexual reproduction). In both cases, significant evolutionary differences were observed, classifying individuals in different classes or clusters according to their genotypic frequencies. Finally, we found that the type of computer hardware has no significant effect on evolutionary computation experiments, as long as evolution is explained by the Darwinian principle of natural selection.

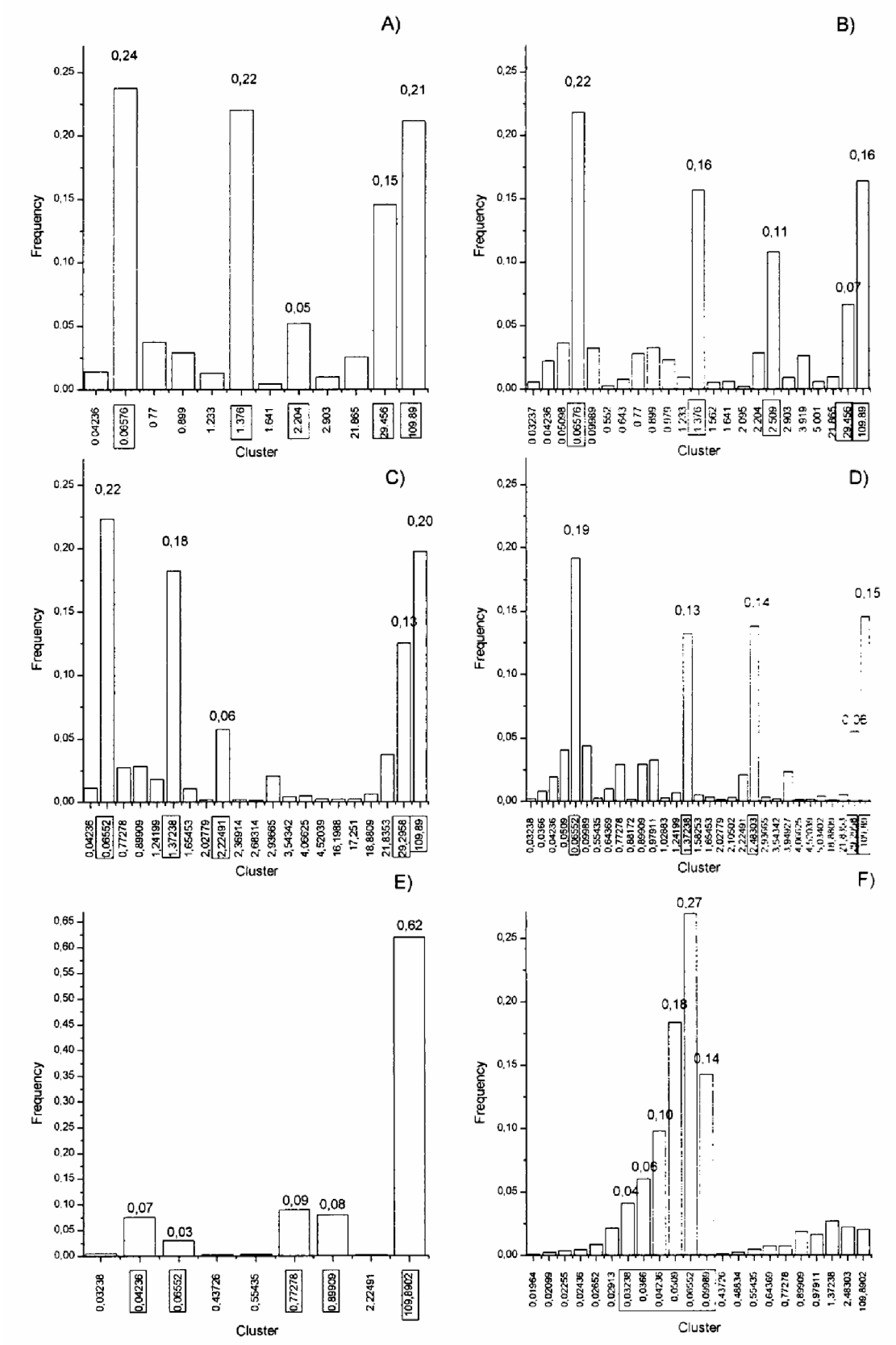
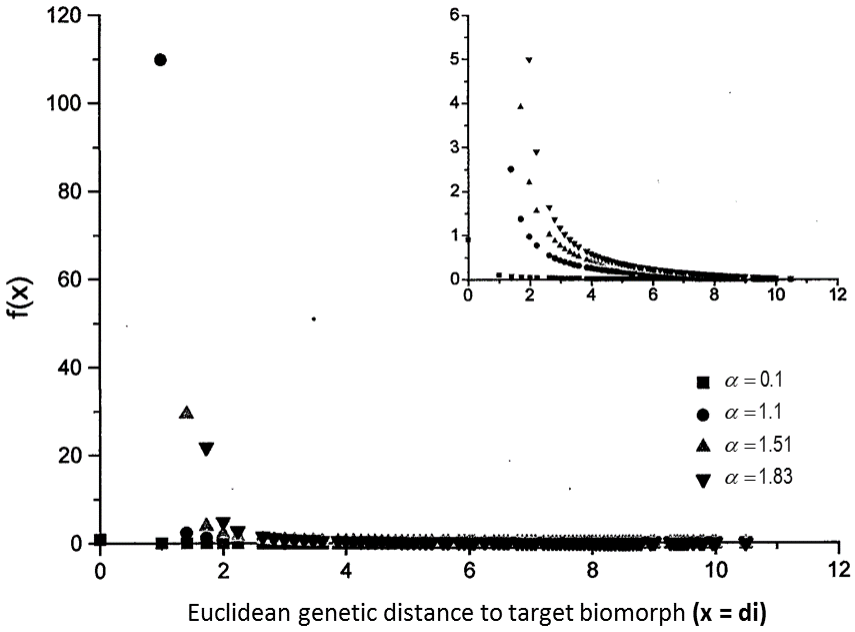


Figure 5.- Genotypic frequency of individuals per class or cluster (for explanation see text)



$$f(x)=\begin{cases}\dfrac{\alpha}{x-0.9\alpha}-0.1\alpha & , \quad x>0\\ \dfrac{\alpha}{0.2-0.9\alpha}-0.1\alpha & , \quad x=0\end{cases}$$

Figure 6.- 'Seminal fitness function' *f(x)* from which the functions **F1**-**F4** of the Table 2 were obtained

Bibliography

[Dawkins, 1986] R. Dawkins. 1986. The Blind Watchmaker. New York: W. W. Norton & Company.

[Guil Lopez, 2000] S. Guil Lopez. 2000. El juego de las siete y media y el problema de la convergencia adaptativa en poblaciones de biomorfos simuladas con ordenador Tesina, Facultad de Biología, Universidad Complutense de Madrid. (Transl.: Spanish).

[Guil Lopez et al., 2000] S. Guil Lopez, R. Lahoz-Beltra, A. Perez de Vargas Luque. 2000. El juego de las siete y media y el problema de la convergencia adaptativa en poblaciones de biomorfos simuladas con ordenador. XXV Congreso Nacional de Estadística e Investigación Operativa, Spain, Vigo, 4-7 de abril de 2000: 221-222. (Transl.: Spanish).

[Lahoz-Beltra, 2004] R. Lahoz-Beltra. 2004. Bioinformática: Simulación, Vida Artificial e Inteligencia Artificial. Madrid: Ediciones Diaz de Santos. (Transl.: Spanish).

[Lahoz-Beltra, 2008] R. Lahoz-Beltra. 2008. ¿Juega Darwin a los Dados? Madrid: Editorial NIVOLA. (Transl.: Spanish).

[Lahoz-Beltra and Perales-Gravan, 2010] R. Lahoz-Beltra, C. Perales-Gravan. 2010. A survey of nonparametric tests for the statistical analisis of evolutionary computational experiments. International Journal Information Theories and Applications, 17: 49-61.

[Milani, A] A. Milani. 2004. Online Genetic Algorithms. International Journal of Information Theories and Applications 11: 20–28.

[Perales-Gravan et al., 2013] C. Perales-Gravan, J de Vicente Buendia, J. Castellanos, R. Lahoz-Beltra. 2013. Modeling, simulation and application of bacterial transduction in genetic algorithms. International Journal of Information Technologies & Knowledge 7: 11-22.

[Prata, 1993]. S. Prata. 1993. Artificial Life Playhouse. Corte Madera, CA: The Waite Group.

[Thai Dam and Lahoz-Beltra, 2014] D. Thai Dam, R. Lahoz-Beltra. 2014. MICRORAM: A simulation model of a colony of bacteria evolving inside an artificial world. International Journal of Information Theories and Applications 21: 328-338.

Authors' Information

Sara Guil Lopez *– Unidad de Inmunología Viral, Centro Nacional de Microbiología, Instituto de Salud Carlos III, Ctra. Pozuelo km 2, E-28220 Majadahonda, Madrid, Spain.*

Major Fields of Scientific Research: Molecular biology, cell biology, clinical immunology

Pedro Cuesta Alvaro *– Servicios Informáticos, Complutense University of Madrid, Madrid 28040, Spain; e-mail: pcuesta@ucm.es*

Major Fields of Scientific Research: Teaching and research support

Santiago Cano Alsua *– Servicios Informáticos, Complutense University of Madrid, Madrid 28040, Spain; e-mail: scano@ucm.es*

Major Fields of Scientific Research: Teaching and research support

Ramses Salas Machado *– Research Scholar at Carlos III University of Madrid and member of the Grupo de Computación Natural, Departamento de Inteligencia Artificial, Facultad de Informática, Universidad Politécnica de Madrid ; e-mail: ramsesjsm@gmail.com*

Major Fields of Scientific Research: Natural computing

Juan B. Castellanos Peñuela *– Departamento de Inteligencia Artificial, Facultad de Informática, Universidad Politécnica de Madrid, Spain; e-mail: jcastellanos@fi.upm.es*

Major Fields of Scientific Research: Natural computing, membrane computing, molecular computing and artificial neural networks.

Rafael Lahoz–Beltra *– Department of Applied Mathematics, Faculty of Biological Sciences, Complutense University of Madrid, 28040 Madrid, Spain; e-mail: lahozraf@ ucm.es*

Major Fields of Scientific Research: Evolutionary computation, bioinspired algorithms

ON THE BEHAVIOR OF A CLASS OF INFINITE STOCHASTIC AUTOMATON IN A RANDOM ENVIRONMENT

Tariel Khvedelidze

Abstract. *It is proposed the behavior algorithm of a wide class of infinite stochastic automaton in a stationary random environment that reacts to the behavior of the automaton with three possible reactions (win, loss, indifference). Explicit analytical formulas and offered numerical algorithm for computing the probability characteristics of the behavior of this automaton. In terms of these characteristics is given complete classification of the possible behavior of infinite stochastic automaton in this environment.*

Keywords: *infinite automaton, random environment, reaction of the environment, the behavior of the* automaton, change of actions, win of automaton.

Introduction

The problem of the behavior of finite automata in a random environment was first formulated Tsetlin [1], in which, as in the works of other authors, the study behavior of the asymptotic sequences of finite automata was based on a study of the final probability (at time $t \to \infty$) Markov chains describing the behavior of finite automata in random environments. However, the disadvantage of this method is what the behavior of finite automata in random environments have been studied insufficiently full , in particular, there was no complete classification of possible asymptotic behavior of finite automata in stationary random environments. Such an analysis proved to possible thanks to the study of the behavior of infinite (with countably many states) stochastic automata, the definition of convergence (in a reasonable sense) sequences of finite automata to their respective infinite automata. With this approach the asymptotic behavior of finite automaton is classified in accordance with behavior of the limit automaton [2].

Studies related to the study of the behavior of automata in stationary random environments have shown that the construction of the automaton, is the best for of some signs in any environment, is unrealistic. Therefore, need to build structure and development analytical and numerical methods for finding the statistical characteristics of the behavior of broad classes of Automata that can be used for solutions a variety of practical problems.

In this paper propose an algorithm of behavior of a wide class of infinite stochastic automaton operating in a stationary random environment in the assumption that all possible reactions of the environment perceived by automaton, as belonging to one of three classes: class favorable reactions (win), class of adverse reactions (loss) and the class of neutral reactions (indifferent). Explicit analytical formulas and offered numerical algorithm for computing the probability characteristics of the behavior of this automaton. In terms of these characteristics is given complete classification of the possible behavior of infinite stochastic automaton in this environment.

Analysis of the behavior of infinite stochastic automaton $T_2^{(x)}(l, 1; \varepsilon, \eta)$ in a ternary stationary random environment $C(a_1, r_1; a_2, r_2)$

Consider the scheme of behavior of automata in a random environment under the assumption that every possible reactions $S\epsilon\{s_1, s_2, \dots, s_g\}$ environment perceived automaton, as opposed to [1.2], as referring to one of three classes: class of favorable reactions (win, $s = +1$), class of adverse reactions (loss, $s = -1$) and the class of neutral reactions (indifference, s = 0).

Let automaton A_k functioning in ternary stationary random environment $C(a_1, r_1; a_2, r_2; \dots; a_k, r_k)$ and if the automaton produces action $f_\alpha(\alpha = \overline{1; k})$, then the environment C generates value of the signal on the input of automaton $s = +1$ with probability $q_\alpha = \frac{1-r_\alpha+a_\alpha}{2}$, value of the signal $s = -1$ with probability $p_\alpha = \frac{1-r_\alpha-a_\alpha}{2}$ and value of the signal $s = 0$ with probability $r_\alpha = 1 - q_\alpha - p_\alpha$ $(\alpha = \overline{1; k})$.

Here value $a_\alpha = q_\alpha - p_\alpha$ ($|a_\alpha|$ < 1-r_α) it makes sense to mathematical expectation of win for action f_α in environment $C(a_1, r_1; a_2, r_2; \dots; a_k, r_k)$. For definiteness we assume that $a_1 > a_2 \geq \cdots \geq a_k$, so that the action f_1 automaton A_k with middle win a_1 in environment $C(a_1, r_1; a_2, r_2; \dots; a_k, r_k)$ it is optimal.

In studying the of the possible behavior of the infinite automaton in a stationary random environment $(a_1, r_1; a_2, r_2; \dots; a_k, r_k)$, essential is the calculation such of the statistical characteristics of the behavior of the automaton, how probability σ_α change (when - ever) action f_α and mathematical expectations of random τ_α time before change f_α action at start of the automaton $x \in L_\alpha$ $(\alpha = \overline{1; k})$ [2].

In terms of this set of characteristics behavior of the infinite automaton A_k in a random environment is classified as follows.

Definition. Following [2], we say that the infinite automaton A_k , functioning in the ternary stationary random environment $C(a_1, r_1; a_2, r_2; \dots; a_k, r_k)$, is:

optimal, when $\sigma_{x,1} < 1,\ \sigma_{x,\alpha} = 1,\ \alpha = \overline{2;k},\ \forall x;$

strictly optimal, when $\sigma_{x,1} < 1,\ \sigma_{x,\alpha} = 1,\ \tau_{x,\alpha} < \infty,\ \alpha = \overline{2;k},\ \forall x;$

quasi optimal, when $\sigma_{x,\alpha} = 1,\ \alpha = \overline{1;k},\ \tau_{x,1} = \infty,\ \tau_{x,\alpha} < \infty,\ \alpha = \overline{2;k},\ \forall x;$

retractable, when $\sigma_{x,\alpha} < 1\ ,\ \forall x, \alpha;$

pushed out, when $\sigma_{x,\alpha} = 1,\ \ \tau_{x,\alpha} < \infty\ ,\ \forall x, \alpha;$

anti-optimal, when $\sigma_{x,k} < 1,\ \sigma_{x,\alpha} = 1,\ \alpha = \overline{1;k-1},\ \forall x;$

anti- quasi - optimal, when $\sigma_{x,\alpha} = 1,\ \alpha = \overline{1;k},\ \tau_{x,k} = \infty,\ \forall x.$

Let infinite (with countably many states) stochastic automaton $T_2^{(x)}(l, 1; \varepsilon, \eta)$ $(0 \le \varepsilon, \eta \le 1,\ 0 \le \varepsilon + \eta \le 1,\ l = 1,2, \dots)$ with $L = L_1 \cup L_2 = \{0, \pm 1, \pm 2, \dots, \pm n, \dots\}$ internal states and two $F_2 = \{f_1, f_2\}$ actions, functioning in the ternary stationary random environment $C(a_1, r_1; a_2, r_2)$. The automaton in the states of the area L_1 with numbers $x = \{\dots, -n, \dots, -1, 0'\}$ performs action f_1, in the states of the area L_2 with numbers $x = \{0,1,2, \dots, n, \dots\}$ - action f_2. For symmetry automaton the area L_1 and L_2 we have implemented with the number 0. If this will cause misunderstanding, then the state 0 the area L_1 will be called $0'$.

We define algorithm behavior of the infinite stochastic automaton $T_2^{(x)}(l, 1; \varepsilon, \eta)$ in the ternary stationary random environment $C(a_1, r_1; a_2, r_2)$ as follows: when the signal $s = +1$ (win) states with the numbers $x = i$ and $x = -i$ $(i = 0,1,2, \dots)$ respectively, goes into a states with the numbers $x = i + 1$ and $x = -(i + 1)$ $(i = 1,2, \dots)$; when the signal $s = -1$ (loss) state of with the numbers $x = i$ and $x = -i$ $(i = 1,2, \dots)$ are moving to states with the numbers $x = i - 1$ and $x = -(i - 1)$ respectively; the state with number $x = 0$ goes over at any one state with number $x = -i$, $i = 0', 1,2, \dots$, and the state with number $x = 0'$ at any one state with number $x = i$, $i = 0,1,2, \dots$; when the signal $s = 0$ all states $x = i$ $(i = 0, \pm 1, \pm 2, \dots)$ with probability $1 - \varepsilon - \eta$ are mapped into themselves, and with probability ε (with probability η) transitions between states are determined also as a signal when $s = -1$ $(s = +1)$.

Thus, the automaton $T_2^{(x)}(l, 1; \varepsilon, \eta)$ can make jumps on one state of in the direction of the area L_α with a probability $P_\alpha = p_\alpha + \varepsilon r_\alpha$; on l of state $(l = 1,2, \dots)$ deep into the area L_α with probability $Q_\alpha = q_\alpha + \eta r_\alpha$ or stay at the same of states with probability $R_\alpha = (1 - \varepsilon - \eta) r_\alpha,\ \alpha = 1,2$. Easy

to see that a change the actions of in one clock cycle of functioning of only possible from one state $x = 0\ (0')$.

In the future we will mainly consider the behavior of the automaton in the area, marked by some action before replace it and the index , for brevity ignore.

Let $u_{x,d}$ the probability that infinite stochastic automaton $T_2^{(x)}(l, 1; \varepsilon, \eta)$ at time d for the first time replace the action of f, lifting off from any state with number x area L.

Taking into account the tactics of behavior of the automaton $T_2^{(x)}(l, 1; \varepsilon, \eta)$ in a stationary random environment $C(a_1, r_1; a_2, r_2)$, relative to probabilities of $u_{x,d}$ will have the difference equation

$$u_{x,d+1} = Pu_{x-1,d} + Qu_{x+l,d} + Ru_{x,d}, \quad x = 1,2, \ldots \quad d = 0,1,2, \ldots \ , \tag{1}$$

where

$$P = p + \varepsilon r, \ \ Q = q + \eta r, \quad R = (1 - \varepsilon - \eta)r, \ P + Q + R = 1$$

and arising from the of the probabilistic of meaning $u_{x,d}$ boundary conditions

$$u_{-1,0} = 1, \ \ u_{x,0} = 0 \ \ \forall x \geq 0. \tag{2}$$

Then the respect to the derivative function probabilities change action

$$U_x(z) \ = \sum_{d=0}^{\infty} u_{x,d}\, z^d,$$

from (1), (2) obtain the boundary value problem

$$U_x(z) = \frac{Pz}{1-Rz} U_{x-1}(z) + \frac{Qz}{1-Rz} U_{x+l}(z), \quad x = 0,1,2, \tag{3}$$

$$U_{-1}(z) = 1.$$

A solution of equation (3) is

$$U_x(z) = \lambda^{x+1}(z), \tag{4}$$

where $\lambda(z)$ the roots of the characteristic equation

$$Qz\lambda^{l+1}(z) - (1 - Rz)\lambda(z) + Pz = 0. \tag{5}$$

On roots of the equation (5) we have the following lemma, evidence which is based on the Rouche theorem and take place in the same way as in [3].

Lemma. 1. For $|z| < 1$ $(z \neq 0)$all the roots of equation (5) is a simple; one root $\lambda_1(z)$ is situated in unit circle k_1 complex λ plane, others roots l $\lambda_j(z)$, $j = \overline{2, l+1}$ - outside.

2. For $P > lQ$: $\lambda_1(1) = 1$, $|\lambda_{j+1}(1)| > 1$, $j = \overline{1, l}$;

For $P = lQ$: $\lambda_1(1) = \lambda_2(1) = 1$, $|\lambda_{j+1}(1)| > 1$, $j = \overline{2, l}$;

For $P < lQ$: $\lambda_1(1) < 1$, $\lambda_2(1) = 1$, $|\lambda_{j+1}(1)| > 1$, $j = \overline{2, l}$.

From the limited function $|U_x(z)| \leq 1$, based on the lemma and the boundary condition (3), have

$$U_x(z) = \lambda_1^{x+1}(z). \tag{6}$$

Since when $P \geq lQ$ $\lambda_1(1) = 1$, while $P < lQ$ $|\lambda_1(1)| < 1$, then

$$\sigma_x = U_x(1) = \begin{cases} 1, & at \ P \geq lQ \\ \lambda_1^{x+1}(1) < 1, & at \ P < lQ \end{cases}.$$

To calculate the of the mathematical expectation of time τ_x , which automaton spends in the area L before change the action f, multiply the (1) on $d+1$ and sum over all $d = 0,1,2,\ldots$ As a result for of define the τ_x obtain the equation

$$\tau_x = \frac{P}{Q+P}\tau_{x-1} + \frac{Q}{Q+P}\tau_{x+l} + \frac{1}{Q+P}, \ \tau_{-1} = 0, \ x = \{0,1,2,\ldots\}. \tag{7}$$

When $P > lQ$ equation (7) has a solution

$$\tau_x = \frac{x+1}{P-lQ}. \tag{8}$$

Be noted that when $P = lQ$ $\tau_x = \infty$. In the case of $P < lQ$ $\sigma_x < 1$ and $\tau_x = \infty$ [4].

Thus, the following theorem holds.

Theorem1. Statistical characteristics for infinite stochastic automaton $T_2^{(x)}(l, 1; \varepsilon, \eta)$ in the ternary stationary random environment $C(a_1, r_1; a_2, r_2)$ defined by the formulas:

at $P_\alpha > lQ_\alpha$: $\sigma_{x,\alpha} = 1$, $\tau_{x,\alpha} = \frac{|x|+1}{P_\alpha - lQ_\alpha}$;

at $P_\alpha = lQ_\alpha$: $\sigma_{x,\alpha} = 1$, $\tau_{x,\alpha} = \infty$;

at $P_\alpha < lQ_\alpha$: $\sigma_{x,\alpha} = \lambda_1^{|x|+1}(1)$, $\tau_{x,\alpha} = \infty$ $(\alpha = 1,2)$.

Note that the condition $P_\alpha < lQ_\alpha$ equivalent to the condition

$$a_\alpha > \frac{1 - l + [(1-2\eta)l - (1-2\varepsilon)]r_\alpha}{l+1}, \qquad \alpha = 1,2$$

and the results obtained makes it possible to make the following statement for classification depending on the values of the quantity $[(1-2\eta)l-(1-2\varepsilon)](r_1-r_2)$.

Theorem. The behavior of infinite stochastic automaton $T_2^{(x)}(l,1;\varepsilon,\eta)$ in the ternary stationary random environment $C(a_1,r_1;a_2,r_2)$ is:

1) if $[(1-2\eta)l-(1-2\varepsilon)](r_1-r_2)<0$, then

at $a_1 \le \frac{1-l+[(1-2\eta)l-(1-2\varepsilon)]r_1}{l+1}$, $a_2 > \frac{1-l+[(1-2\eta)l-(1-2\varepsilon)]r_2}{l+1}$ – optimal:

$$\sigma_{x,1}=1,\ \sigma_{x,2}<1,\quad \tau_{x,1}=\frac{|x|+1}{P_1-lQ_1},\ \tau_{x,2}=\infty;$$

at $a_1 < \frac{1-l+[(1-2\eta)l-(1-2\varepsilon)]r_1}{l+1}$, $a_2 > \frac{1-l+[(1-2\eta)l-(1-2\varepsilon)]r_2}{l+1}$ – strictly optimal:

$$\sigma_{x,1}=1,\quad \sigma_{x,2}<1,\quad \tau_{x,1}=\frac{|x|+1}{P_1-lQ_1}<\infty,\quad \tau_{x,2}=\infty;$$

at $a_1 > \frac{1-l+[(1-2\eta)l-(1-2\varepsilon)]r_1}{l+1}$, $a_2 \le \frac{1-l+[(1-2\eta)l-(1-2\varepsilon)]r_2}{l+1}$ – optimal or anti optimal:

$$\sigma_{x,1}=\lambda_1^{|x|+1}(1)<1,\ \sigma_{x,2}=1,\ \tau_{x,1}=\infty,\quad \tau_{x,2}=\frac{x+1}{P_2-lQ_2};$$

at $a_1 = \frac{1-l+[(1-2\eta)l-(1-2\varepsilon)]r_1}{l+1}$, $a_2 < \frac{1-l+[(1-2\eta)l-(1-2\varepsilon)]r_2}{l+1}$ – quasi optimal or anti quasi-optimal: $\sigma_{x,\alpha}=1,\ \alpha=1,2,\ \tau_{x,1}=\infty,\quad \tau_{x,2}=\frac{x+1}{P_2-lQ_2}<\infty$;

at $a_\alpha > \frac{1-l+[(1-2\eta)l-(1-2\varepsilon)]r_\alpha}{l+1}$ – retractable:

$$\sigma_{x,\alpha}=\left(\frac{P_\alpha}{Q_\alpha}\right)^{|x|+1}<1,\ \tau_{x,\alpha}=\infty,\alpha=1,2;$$

at $a_\alpha < \frac{1-l+[(1-2\eta)l-(1-2\varepsilon)]r_\alpha}{l+1}$ – pushed out: $\sigma_{x,\alpha}=1,\ \tau_{x,\alpha}=\frac{|x|+1}{P_\alpha-lQ_\alpha}<\infty,\ \alpha=1,2;$

2) if $[(1-2\eta)l-(1-2\varepsilon)](r_1-r_2)\geq 0$, then

at $a_1 > \frac{1-l+[(1-2\eta)l-(1-2\varepsilon)]r_1}{l+1}$, $a_2 \leq \frac{1-l+[(1-2\eta)l-(1-2\varepsilon)]r_2}{l+1}$ – optimal:

$$\sigma_{x,1}=\lambda_1^{|x|+1}(1)<1,\ \ \sigma_{x,2}=1, \tau_{x,1}=\infty,\ \tau_{x,2}=\frac{x+1}{P_2-lQ_2};$$

at $a_1 > \frac{1-l+[(1-2\eta)l-(1-2\varepsilon)]r_1}{l+1}$, $a_2 < \frac{1-l+[(1-2\eta)l-(1-2\varepsilon)]r_2}{l+1}$ – strictly optimal:

$$\sigma_{x,1}=\lambda_1^{|x|+1}(1)<1,\quad \sigma_{x,2}=1,\ \tau_{x,1}=\infty,\ \ \tau_{x,2}=\frac{x+1}{P_2-lQ_2}<\infty;$$

at $a_1 = \frac{1-l+[(1-2\eta)l-(1-2\varepsilon)]r_1}{l+1}$, $a_2 < \frac{1-l+[(1-2\eta)l-(1-2\varepsilon)]r_2}{l+1}$ – quasi optimal:

$$\sigma_{x,\alpha}=1,\qquad \alpha=1,2,\ \ \tau_{x,1}=\infty,\ \ \tau_{x,2}=\frac{x+1}{P_2-lQ_2}<\infty;$$

at $a_\alpha > \frac{1-l+[(1-2\eta)l-(1-2\varepsilon)]r_\alpha}{l+1}$ – retractable:

$$\sigma_{x,\alpha}=\lambda_1^{|x|+1}(1)<1,\ \tau_{x,\alpha}=\infty,\ \alpha=1,2;$$

at $a_\alpha < \frac{1-l+[(1-2\eta)l-(1-2\varepsilon)]r_\alpha}{l+1}$ – pushed out:

$$\sigma_{x,\alpha}=1,\ \ \tau_{x,\alpha}=\frac{|x|+1}{P_\alpha-lQ_\alpha}<\infty,\ \ \alpha=1,2.$$

From these formulas see that in the case of $P_\alpha < lQ_\alpha$ probability $\sigma_{x,\alpha}$ action change f_α is a function of the smallest root of the characteristic equation (5) for $z=1$.

The decomposition of the generating function obtained by the method of the Newton-Puiseux diagrams (at starting of the automaton $x=0$ and at $P_\alpha < lQ_\alpha$) allows approximately calculate the probability of $\sigma_{0,\alpha}$ action change [5].

This expansion in this case is as follows:

$$U_0(z)=\frac{P_\alpha z}{1-R_\alpha z}+\frac{Q_\alpha z}{1-R_\alpha z}\left(\frac{P_\alpha z}{1-R_\alpha z}\right)^{l+1}+(l+1)\left(\frac{Q_\alpha z}{1-R_\alpha z}\right)^2\left(\frac{P_\alpha z}{1-R_\alpha z}\right)^{2l+1}+ \\ +\frac{(l+1)(3l+2)}{2}\left(\frac{Q_\alpha z}{1-R_\alpha z}\right)^3\left(\frac{P_\alpha z}{1-R_\alpha z}\right)^{3l+1}+\cdots \tag{9}$$

and $\sigma_{0,\alpha}=U_0(1)$.

With the help of this expansion can be carried out approximate computations of the $\sigma_{0,\alpha}$ at $l \geq 3$.

In the particular case, at $l = 1$: $\sigma_{0,\alpha} = \frac{P_\alpha}{Q_\alpha}$, and when $l = 2$: $\sigma_{0,\alpha} = \frac{\sqrt{1+4\frac{P_\alpha}{Q_\alpha}}-1}{2}$.

Observe, that the expansion (9) allows obtain an approximate value of the probability $T_2^{(x)}(l, 1; \varepsilon, \eta)$ action changing of the automaton in case of when the initial state automata is $x = 0$.

In the general case, for an approximate calculation $\sigma_{x,\alpha}$ and $\tau_{x,\alpha}$, for any starting state of $x \geq 0$ of the automaton, can be used a numerical algorithm, built on the basis of the expansion of the function generating $U_0(z)$ [5].

On the basis of (9) we introduce the following parameters

$$\xi = \sqrt[l+1]{QP^l}, \quad \theta = \sqrt[l+1]{\frac{Q}{P}}$$

and consider the function

$$W_x(z) = \theta^{x+1} U_x(z).$$

Multiplying (3) by θ^{x+1} with respect to $W_x(z)$ we obtain the following boundary value problem:

$$W_x(z) = \frac{\xi z}{1 - Rz} W_{x-1}(z) + \frac{\xi z}{1 - Rz} W_{x+l}(z), \qquad (10)$$
$$W_{-1}(z) = 1.$$

From (10) it follows that

$$W_x(z) = \sum_{d=0}^{\infty} \sum_{j=1}^{d} \xi^j \, R^{d-j} \, A_x^{(d-j)}(d) \; z^d,$$

where the magnitudes $A_x^{(d-j)}(d)$ require the determination. Then

$$U_x(z) = \theta^{-(x+1)} \sum_{d=0}^{\infty} \sum_{j=1}^{d} \xi^j \, R^{d-j} \, A_x^{(d-j)}(d) \; z^d,$$

$$u_{x,d} = \theta^{-(x+1)} \sum_{j=1}^{d} \xi^j \, R^{d-j} \, A_x^{(d-j)}(d) \, z^d. \qquad (11)$$

From (1), taking into account (11), with respect to the quantities $A_x^{(d-j)}(d)$ obtain the equations:

$$A_x^{(d-j)}(d) = A_{x-1}^{(d-j)}(d-1) + A_{x+l}^{(d-j)}(d-1) + A_x^{(d-j-1)}(d-1),$$

$$x \geq 0, \qquad j = 0,1,2,..d,$$

$$A_x^{(-1)}(d-1) = A_{x-1}^{(d)}(d) = A_{x+l}^{(d)}(d) = 0,$$

$$A_{-1}^{(0)}(0) = 1, \ A_{-1}^{(j)}(0) = 0, \ j = 1,..,d, \ A_x^{(j)}(0) = 0 \ \ \forall x \geq 0, \qquad j = 0,1,..,d.$$

To find quantities $A_x^{(d-j)}(d)$ is necessary to determine the number of paths when the automaton with a starting state $x \geq 0$ change action (Fig. 1).

If for some d $(d = 1,2, ...)$ $x - d \geq 0$ condition is not satisfied, then this means that the automaton can at the d-th action change operation. Since automaton $T_2^{(x)}(l, 1; \varepsilon, \eta)$ makes one state transition in the direction of the area, then such a state is only one state $x = -1$.

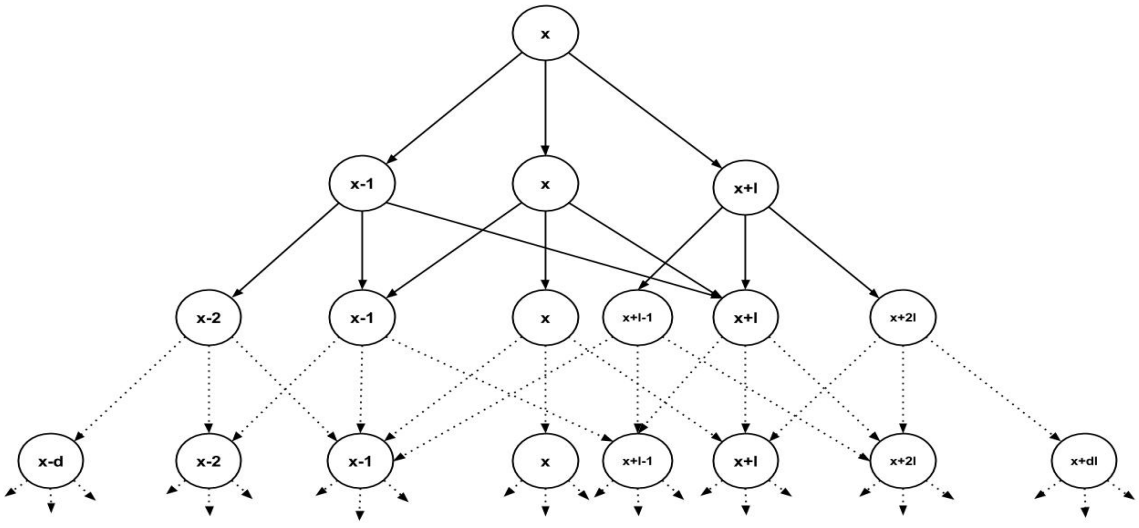


Fig.1. Graf of transition of the automaton $T_2^{(x)}(l, 1; \varepsilon, \eta)$ in the area of the states provided $x - d \geq 0$.

Let $N(d)$ the maximum number of possible states of d-tier, and by $H^{(d-j)}(x,i,d)$- the number of paths that lead from state automaton with the number x in the i- th state of the d- tier at $d-j$ stopovers.

Easy notice that the $N(d)=x+ld$ and for determining $H^{(d-j)}(x,i,d)$ have the following recursive relation:

$$H^{(d-j)}(x,N(d)-i,d)=$$

$$=H^{(d-j)}(x,N(d-1)+l+1-i,d-1)\ \mu(N(d-1)+l+1-i)+$$

$$+H^{(d-j)}(x,N(d-1)-i,d-1)\ \mu(N(d-1)-i)+$$

$$+H^{(d-j)}(x,N(d-1)+l-i,d-1)\ \ \mu(N(d-1)+l-i),$$

$$H^{(d-j)}(x,N(d),d)=\begin{cases}1, & j=d\\ 0, & j\neq d\end{cases},\qquad H^{(d-j)}(x,x-1,1)=\begin{cases}1, & j=d\\ 0, & j\neq d\end{cases},$$

$$H^{(d-j)}(x,x,1)=\begin{cases}1, & j=d-1\\ 0, & j\neq d-1\end{cases},\ H^{(d-j)}(x,i,1)=0\ \ ,\quad \forall\ i\neq x,\ x-1,x+l,$$

$$H^{(-1)}(x,i,d)=0\ ,\ \forall\ i,\ d,\ N(d)=x+ld.$$

$$j=1,2,\dots,d,\ \ i=1,2,\dots,x+dl,\ \ \mu(x)=\begin{cases}1, & x\geq 0\\ 0, & x<0\end{cases};$$

Then $A_x^{(d-j)}(d)=H^{(d-j)}(x,-1,d)$ and finally will have:

$$\sigma_x=\theta^{-(x+1)}\sum_{d=0}^{\infty}\sum_{j=1}^{d}\xi^j\ R^{d-j}\ \ H^{(d-j)}(x,-1,d),$$

$$\tau_x=\theta^{-(x+1)}\sum_{d=0}^{\infty}\sum_{j=1}^{d}d\ \ \xi^j\ R^{d-j}\ H^{(d-j)}(x,-1,d).$$

It should be noted that on the fruitfulness of the of the method of diagrams Newton-Puisaye, both for the constructing numerical algorithms, and for the direct calculation of the statistical characteristics of the behavior of infinite automata, was first pointed out in [3].

Conclusion

Thus, the possible behavior of infinite stochastic automaton $T_2^{(x)}(l, 1; \varepsilon, \eta)$ in the ternary stationary random environment $C(a_1, r_1; a_2, r_2)$, is completely determined by the statistical characteristics of behavior $\sigma_{x,\alpha}$ of the automaton and $\tau_{x,\alpha}$, $\alpha = 1,2$.

The decomposition of the generating function, obtained by the method of the Newton - Puiseux diagrams (when the starting state of the automaton $x = 0$) allows calculate the probability of approximately by $\sigma_{0,\alpha}$ change action. The number of terms in the expansion (9) depends not only on the environmental parameters, but also a on a parameter l, and of the degree of accuracy. In the general case, numerical algorithm allows with sufficient accuracy to calculate $\sigma_{x,\alpha}$ and $\tau_{x,\alpha}$ for any starting state the automaton $x \in \{0,1,2, \dots\}$.

Bibliography

[1] Michael Tsetlin, Research on the theory of automata and modeling of biological systems. M, Science, 1969.

[2] Vladimir Koroljuk, Alexander Pletnev, Samuel Adelman, Automats. Wanderings. Games. Successes of Mathematical Sciences. No. 1 (259) 1988.

[3] Alexander Lobanov, Leonid Pletnev, Samuel Adelman, Analytical and numerical methods for calculating the probability characteristics of behavior in random media a lot of input stochastic automata. Cybernetics, No 6, 1984.

[4] William Fellerr, An Introduction to Probability Theory and Its Applications. Volume 1, 1967.

[5] Tariel Khvedelidze, Numerical methods for finding the probability characteristics of behavior of automata with three types of reactions in a stationary random environment. Proceedings of the TSU, No .300, 1990.

Authors' Information

Tariel Khvedelidze *- Iv.Javakhishvili Tbilisi State University, Faculty of Exact and Natural Sciences, professor of Computer Sciences*

e-mail: tariel.khvedelidze@tsu.ge

Scientific Research*: General theoretical information research, information systems and computer Sciences*

TEST GENERATION FOR ESTIMATING POWER CONSUMPTION OF SEQUANTIAL CIRCUITS

Liudmila Cheremisinova

Abstract*: The reliability of electronic circuits is closely related to the power dissipated by them. Tools for evaluating the power consumption of sequential circuits is becoming a primal concern for designers of low-power circuits. The problem of estimation of the projected power, consumed by the CMOS sequential circuits, by means of its simulation is discussed. The task of forming test sequences of input actions to estimate the average circuit consumption is considered for the case when for the circuit automaton description in the form of Finite State Machine is available. The proposed method of test generation is based on special graph models of sequential circuits that allow formalizing the process of generating test sequences.*

Keywords*: power consumption, low-power design, test generation.*

ACM Classification Keywords*: B.6.1 Logic design: Design Style – Sequential circuits; B.7.3 Integrated Circuits: Reliability and Testing – Test Generation*

Introduction

Power consumption has become the major issue in electronic research since about 1990, it is being given increased weightage in comparison to area and speed. The minimization of power dissipation has become a task of critical concern with the advent of high density integrated circuits and portable micro-electronic devices. In the first case the heat generated by integrated circuits begins to exceed the ability of packaging to dissipate it, and in the second case new portable applications (such as notebook computers, cellular phones) require high speed, yet low power consumption. Low power ASIC design results in improved reliability and increased battery life. The Semiconductor Industry Association technology roadmap [SIA, 2014] has identified low power design techniques as a critical technological need in semiconductor industry today.

The development of methods and software tools that can help designers to optimize digital circuits for power consumption has received increasing attention. Accurate and efficient power estimation during design phase is required. The appropriate tools must have efficient means to estimate the power consumed by a circuit on different design phases. At present an increasing attention is focused not only

on transistor-level design but on higher levels of abstraction because early power estimation is important in VLSI circuits, because it has a significant impact on the reliability of the circuits under design. In the process of optimizing circuits for low power a designer is interested in knowing the effects of specific design techniques on the power consumption of the projected circuit. With the relevant information about power characteristics designer can redesign or correct a circuit in early design stages if it is stated that it can consume more power than expected.

Currently, the simplest and most direct power estimation can be done by circuit simulation when the monitoring of the power supply current is done [Benini, 2002]. Power consumption values are determined which depend on the given test sequence. So, using simulators, power is measured for a specific set of input patterns (often chosen randomly) referred to as test sequence. There are circuit-level power estimators available as commercial tools. For example, the most known SPICE [Nagel, 1973]. But the simulation results are highly related to the input patterns given to the circuit [Kang, 1986]. Simulation methods suffer from two major drawbacks. First, they are very time consuming, especially for large circuits (because to produce a meaningful power estimate the required number of simulated vectors is usually high). Second, it needs to know the set of input patterns when the power for a designed circuit embedded in a large system is to be calculated. Thus, the calculated power may be erroneous because some of input patterns used for estimation may never occur during normal circuit operation.

Using simulators, power is measured for a specific set of input vectors, and can be referred to average power consumption. Many investigations were focusing on the average power estimation [Arasu, 2013; Chou, 1996; Ghosh, 1992; Najm, 1994; Wang1, 1996]. The proposed methods are not only simulation-based but probabilistic methods are very popular too [Chou, 1996; Ghosh, 1992; Najm, 1994].

Power and switching activity estimation for sequential circuits is significantly more complex task than that for combinational circuits because power value depends not only on input test patterns but on the state the circuit is in. Although the problem of estimation of power in VLSI circuits is essential for determining the appropriate packaging and cooling techniques, optimizing the power and ground routing networks, there are a limited number of papers devoted to the problem of average and maximum power estimation of sequential circuits.

In the paper the task of average switching activity estimation for CMOS synchronous sequential circuit is considered when its automaton description in the form of Finite State Machine (FSM) is available. The methods of generating test sequences are based on finding out directed paths of special type for proposed graph models generated by FSM state transition graph (STG). The paths are closely related with input patterns for simulating the sequential circuit for power estimation.

In the proposed approach, we construct test sequences that, as we expect, allow obtaining the estimate of average power consumption and:

1) we are not interested in detail of the target circuit structure and use only its global structure – STG of FSM;

2) we expect the good correlation between switching frequency used for test sequence estimate and the actual switching frequency during normal mode of circuit operation;

3) the process technology is not taken into account.

Once the test sequences have been determined circuit-level simulation should be performed to determine accurately the associated values of average power consumption.

Basic definitions

Let any input pattern of the test sequence (looked for) be a Boolean vector from the Boolean space of dimension *n*, where *n* is the number of the circuit inputs.

The difference between combinational and sequential circuits is the memory elements issue. The states of sequential circuits cannot be assigned to arbitrary values but only to reachable ones. If the initial state is initialized to any arbitrary value during the power estimation, then the power value will be wrong since unreachable states are not allowed. The simulation-based power calculation procedure is comprised of three phases: generation (may be randomly) of a sequence of input patterns to be tested; simulation of the tested circuit on the sequence of input patterns estimating power dissipation on each clock cycle and then calculation of the average value of power dissipation. In the case of sequential circuit, the initial sequence of input patterns should start from some reachable state (it may be reset state). So, there are the following difficulties of usage of simulation-based method for the power estimation:

1) the need to generate such a sequence of input patterns that ensures energy normal mode of circuit operation (otherwise we do not get estimate of the average power consumption);

2) the simulation process is very time consuming because of the great number of simulated vectors for large circuits to produce a meaningful power estimate;

3) the necessity to initialize the tested sequential circuit, to start simulation from a reachable state;

4) baffling complexity of the task because a sequential circuit can be considered as a series of combinational circuits with different initial reachable states.

At the level of logic design a gate-level netlist is generated often from a FSM description, so a circuit operation is reflected by an appropriate FSM structure. We make an assumption that the sequential circuit automaton description in the form of FSM state transition graph (STG) is available. We seek for test sequence of input vectors that are the candidates to be tested for the average power dissipation in sequential circuit. It is proposed the test sequence is derived from augmented STG of the given FSM.

STG is a directed graph whose vertices correspond to the automaton states s_i, and its arcs – to the transitions between the states. Any arc of the graph between the states s_i and s_j is marked with an input pattern $\boldsymbol{x}_{ij}$ (a set of values of input variables $\boldsymbol{x} = (x_1, x_2, \dots, x_n)$) which causes the corresponding transition from the state s_i into s_j. It is assumed that the automaton STG is strongly connected, i.e. for any pair of states always there exists a sequence of input pattern that brings the automaton from one state to another. Figure 1 shows an example of such a STG, each arc of the graph is labeled by a pair: the arc number / the input pattern $\boldsymbol{x}_{ij}$.

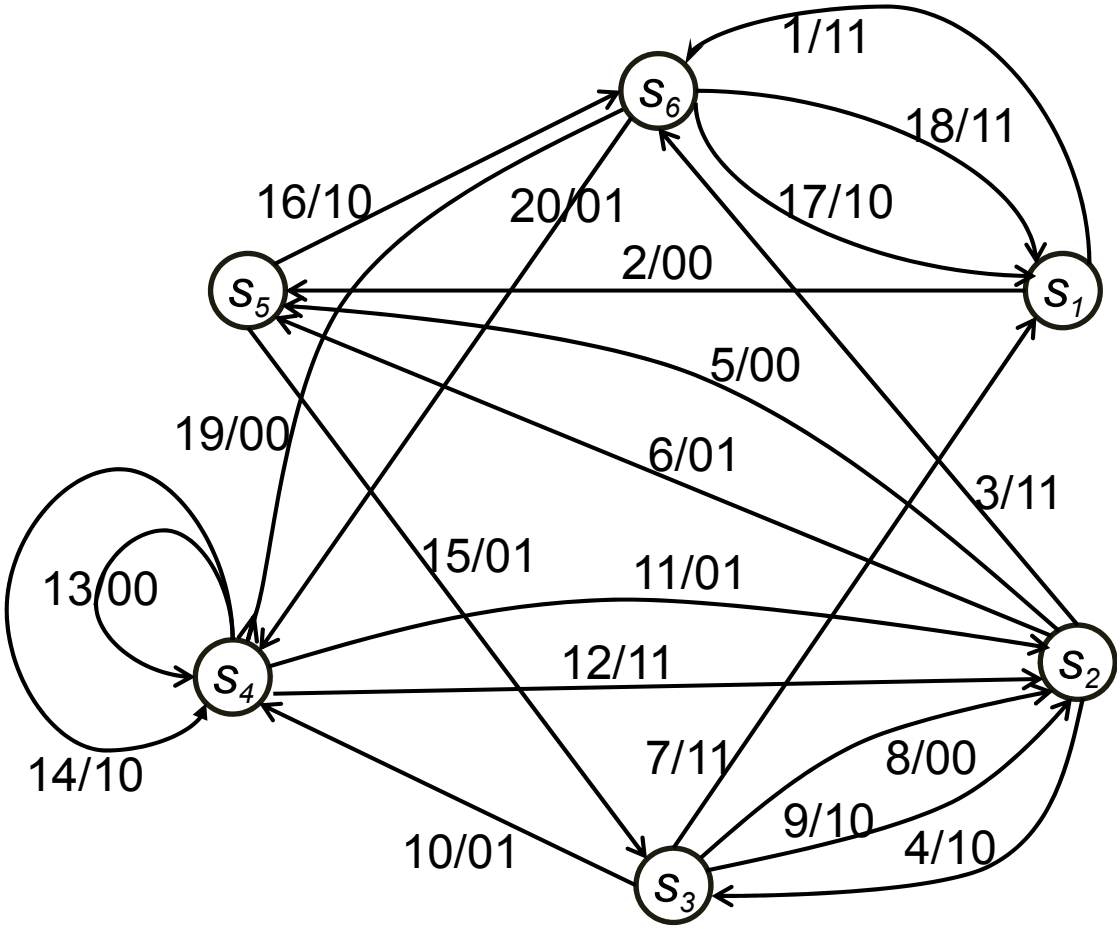


Figure 1. Automaton state transition graph

Let T_i denote test sequence in the form ($\boldsymbol{s}^i$, $\boldsymbol{x}_1^i$, $\boldsymbol{x}_2^i$,…), where $\boldsymbol{s}^i$ is a FSM internal state represented by a Boolean vector of memory element states, $\boldsymbol{x}_j^i$ is a Boolean vector of input variable values representing a FSM input pattern feeding circuit at the *j*-th clock cycle. The values of $\boldsymbol{s}^i$ and $\boldsymbol{x}_1^i$ initialize the circuit at the first clock cycle, before the process of estimating the series of switching's in the circuit. During the simulation of the circuit under the test T_i , the sequence ($\boldsymbol{s}^i$, $\boldsymbol{x}_1^i$, $\boldsymbol{s}_1^i$, $\boldsymbol{x}_2^i$, $\boldsymbol{s}_2^i$, …) of automaton states changes is generated.

Problem statement

For adequate estimation of power consumption a large number of input patterns should be considered to allow making statistically significant conclusions with this test sequence. And the test sequence should correspond to a normal mode of test circuit operation to provide calculating average power consumption. If the conditions of the circuit use are not known, the most effective toll will be pseudo-random test sequence of the exhaustive search, which must satisfy to following demands:

1) it includes every possible ordered pair of input patterns;

2) adjacent elements in the test sequence are presented exactly once.

The minimum size of such a test sequence is $2^n(2^n - 1) + 1$ (that follows from the algorithm of enumeration of oriented pairs in finite sets [Zakrevskij, 2010]). This estimate, however, can be achieved only for the case of combinational circuits. For the circuits with memory, this problem is much more complicated, since it is necessary to take into account changes in the states of the memory elements and their attainability during sequential circuit operation.

So, when simulating the circuit to estimate its average energy consumption it is desirable to analyze its responses to every possible change of its inputs. It is advisable to consider all possible ordered pairs of input combinations that are valid at the normal mode of the circuit functioning.

When testing sequential circuit, input patterns following one after another are separated by different states of memory elements. In other words, it is necessary to test not only pairs but various three element fragments ($\boldsymbol{s}^i$, $\boldsymbol{x}_1^i$, $\boldsymbol{x}_2^i$) which are allowable by the given automaton STG. For example, the arc 7 (Figure 1) corresponding to the automaton transition under input condition $x_1 x_2$ (corresponding to Boolean vector 1 1) is immediately preceded with the arcs 4 or 15 (corresponding to automaton transitions that move up the automaton into the state s_3), i.e. the test sequence should include three element fragments (s_2, x_1 $\bar{x}_2$, s_3, $x_1 x_2$) and (s_5, $\bar{x}_1 x_2$, s_3, $x_1 x_2$) corresponding to pairs of transitions 4, 7 and 15, 7.

The test sequence ($\boldsymbol{s}^i$, $\boldsymbol{x}_1^i$, $\boldsymbol{s}_1^i$, $\boldsymbol{x}_2^i$, $\boldsymbol{s}_2^i$, …) for calculating the energy consumption of a circuit implementing an automaton description should satisfy the following conditions:

1) tinternal state $\boldsymbol{s}^i$ should be reset state as any test scenario can run from the test circuit reset state;

2) test sequence should consist of alternating input patterns and automaton internal states, which are provided by a traversal of all the arcs of the STG (not necessarily once), starting with a given internal state $\boldsymbol{s}^i$;

3) test sequence should include every possible triple mentioned above.

The existence of such a sequence ($\boldsymbol{s}^i$, $\boldsymbol{x}_1{}^i$, $\boldsymbol{s}_1{}^i$, $\boldsymbol{x}_2{}^i$, $\boldsymbol{s}_2{}^i$, …) is provided by the assumption that an automaton STG is strongly connected. The question is how to find it in an efficient manner.

Graph models to search test sequence

The input data for generating the test sequence is a connected directed graph $G = (V, E)$ corresponding to the initial automaton STG. It is multidigraph that may have multiple arcs and sometimes loops.

It would seem, the task of forming the test sequence could be reduced to searching the shortest directed walk (as an open finite alternating sequence of vertices and arcs) that visits each arc in the directed graph G at least once.

The task can be stated as the Chinese postman problem for the case of digraphs: given a directed graph, find the shortest closed walk that visits all the arcs at least once. Indeed, the traversal of every arc of the multidigraph is the necessary condition. But since one of the distinctive features of an automaton model is that there is typically a large number of possible “next actions” at every vertex in the graph $G = (V, E)$ and we would like to test these combinations too. The Chinese Postman (and its variations) solutions guarantee visiting every arc, but not every arc combination of the length 2. The last demand is greatly difficult to perform solving the Chinese Postman problem on the graph $G = (V, E)$.

The idea of the proposed solution of the problem consists in the use of another automaton graph model, which allows easy to deal with combinations of the length 2 of arcs of the graph $G = (V, E)$. Such a graph model proves to be the line graph $L(G)$ of the graph G. The vertexes of the line graph $L(G)$ [Harary, 1969] correspond to the arcs of $G = (V, E)$ and $L(G)$ represents the adjacencies between arcs of G. If $G = (V, E)$ is a directed graph, its line graph is directed too. The line digraph $L(G)$ has one vertex for each arc of G. Two vertices of the line digraph $L(G)$, that correspond to the directed arcs from p to q and from u to v in $G = (V, E)$, are connected by an arc from $(p\ q)$ to $(u\ v)$ in the line digraph when $q = u$. That is, each arc in the line digraph $L(G)$ represents a length-two directed path in $G = (V, E)$ or a pair of automaton transitions.

The directed graph $G = (V, E)$ under consideration has 20 arcs, therefore the line graph $L(G)$ will have 20 vertexes too. The adjacency matrix $\boldsymbol{R}$ of the line graph $L(G)$ (Figure 1) is shown in Table 1. The

columns and rows of the adjacency matrix are associated with of the line graph vertexes labeled the same manner as the corresponding arcs of the graph G. The matrix element $r_i{}^j \in \boldsymbol{R}$ is equal to 1 if there exists an arc coming from the i-th vertex of the graph $L(G)$ to the j-th one (which corresponds to the presence of an ordered pair of arcs i and j in the graph $G = (V, E)$. The last column and row of the adjacency matrix indicate the values of out-degrees d^+ and in-degrees d^- of vertexes associated with the corresponding rows and columns of the matrix.

Table 1. The adjacency matrix ***R*** of the line graph $L(G)$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	d^+
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	4
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**	0	0	0	0	2
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	4
4	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	4
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**	0	0	0	0	2
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**	0	0	0	0	2
7	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
8	0	0	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
8	0	0	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
10	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0	0	0	4
11	0	0	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
12	0	0	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
13	0	0	0	0	0	0	0	0	0	0	**1**	**1**	0	0	0	0	0	0	0	0	2
14	0	0	0	0	0	0	0	0	0	0	**1**	**1**	0	0	0	0	0	0	0	0	2
15	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	4
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	4
17	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
18	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
19	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0	0	0	4
20	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0	0	0	0	0	4
d^-	3	3	4	4	4	4	2	2	2	2	5	5	3	3	3	3	3	3	3	3	

Thus, if we find such a walk in the digraph $L(G)$ that passes through all its arcs at least once, we ensure that all three element fragments ($\boldsymbol{s}^i$, $\boldsymbol{x}_1{}^i$, $\boldsymbol{x}_2{}^i$) generated by STG of FSM are considered. So, the task of finding a test sequence to estimate the average power dissipation of the sequential circuit will be solved. The walk should begin with one of the arcs proceeding from the initial automaton state (reset

state) that is zero encoded. For our example, this arc may be the arc 1 in the graph $G = (V, E)$ and so, the start point should be vertex v_1 of the digraph $L(G)$. The desired test sequence $(s_1, \boldsymbol{x}_1^t, \boldsymbol{x}_2^t, \boldsymbol{x}_3^t, \ldots)$ will consist of input patterns assigned to passable vertexes of the line graph $L(G)$ (and corresponding to the arcs of the digraph $G = (V, E)$.

Searching for the test sequence

As stated above the problem is to find the shortest walk in digraph $L(G)$, which passes through each digraph arc at least once. And the task is to find the shortest walk among all visiting each digraph arc. It is clear that the minimum walk length is achieved when each arc is traversed exactly once. Such a decision may take place in the special case when $L(G)$ is Euler graph (rare graph type). In other cases, the desired walk will contain repeated arcs. The goal is to minimize the number of repeated passages of the arcs. The solved problem is similar to the problem of the Chinese postman and its variation for digraphs – the New York Street Sweeper Problem [Bodin, 1983]. The difference between the mentioned problems and considered in the paper is that we do not need necessary to obtain a cycle and the weights of all arcs are equal in our case.

It is well known that Chinese postman problem (and its variations) is NP-hard. So, there appears a modest chance to find out such a solution for digraphs of high dimension. As it is easy to see, each vertex of degree k in the original graph $G = (V, E)$ creates $k(k\text{-}1)/2$ arcs in the line graph $L(G)$. This means that transforming a "thick" graph into a line one results in considerable increasing its complexity.

For the tasks of the practical dimension it is advisable to use approximate methods of constructing the shortest walks. We will form the desired path in the digraph $L(G)$ starting with some initial vertex v_k (by agreement, it is the v_1) of the graph and selecting at each step one of the arcs outgoing from of v_k that is reached at the step.

Before the start of the shortest walk in digraph $L(G)$ let make a copy $\boldsymbol{C}$ of the matrix $\boldsymbol{R}$ to have the possibility to keep record of arcs that are passed already and repass them only in necessary cases. Then let introduce the sequence D of vertexes that will constitute the walk under consideration. First D has the only initial vertex v_k (by agreement initially $k = 1$) then in the process of walk construction D will appended.

Step 1. The vector $\boldsymbol{s}_k$ is set to be equal the row $\boldsymbol{c}_k \in \boldsymbol{C}$ or, if $\boldsymbol{c}_k = \boldsymbol{0}$ ($\boldsymbol{0}$ vector), to the row $\boldsymbol{r}_k \in \boldsymbol{R}$.

Step 2. From the set of vertices corresponding to the units in the vector $\boldsymbol{s}_k$ (that succeed from the vertex v_k) such a vertex v_l is chosen that corresponds to the row $\boldsymbol{c}_l \in \boldsymbol{C}$ that has the largest weight (the number of units). But here the following cases can take place: A) There exist several such rows (they have the same weight), then for each of them the disjunction of rows $\boldsymbol{c}_i$ mentioned in it is formed

and the resulting vertex v_l will be chosen with accordance to weights of these disjunctions if they differ in weights. Otherwise we could form new disjunctions or simply choose the first of the rows $\boldsymbol{c}_l$ that generates one of disjunctions with the largest weight. B) All the rows $\boldsymbol{c}_l$ generated by $\boldsymbol{s}_k$ have the weight 0, then the rows $\boldsymbol{r}_l \in \boldsymbol{R}$ are considered that correspond to the units in the vector $\boldsymbol{s}_k$ and for them Step 2 is executed.

Step 3. After selecting the vertex v_l the element $c_k^l \in \boldsymbol{C}$ is reset to zero, the index number l is appended to the sequence D of passed vertices and the index number k is set to l. If the resulting matrix $\boldsymbol{C}$ has unit elements, the process of the walk forming continues with the Step 1. Otherwise, the desired shortest walk is found.

The vertices of the digraph $L(G)$, listed in the obtained sequence D, correspond to the numbers of arcs of the graph $G = (V, E)$. The input patterns assigned to the appropriate automaton transitions give the desired test sequence.

Let us pay attention to the considered directed graph $G = (V, E)$ and its line graph $L(G)$. Peeking at the values of out-degrees d^+ and in-degrees d^- of vertexes associated with the corresponding rows and columns of the adjacency matrix $\boldsymbol{R}$ of the line graph $L(G)$ we make conclusion that the graph $L(G)$ is not Euler's one (and not semiEuler). So, the walk will contain recurring arcs. In such a case, it is necessary to minimize the number of repetitions using the proposed procedure.

In the first step of the procedure the first row of the adjacency matrix $\boldsymbol{R}$ is considered. The corresponding vertex v_1 is adjacent to the vertexes of v_{17}, v_{18}, v_{19} and v_{20} (as follows from the first row of the matrix $\boldsymbol{R}$). Two vertices v_{19} and v_{20} (or rows of $\boldsymbol{R}$) of the great weight are equal. So, the first one may be chosen.

Repeating the procedure of constructing a walk to the end, we get the walk of the length of 85, while the digraph $L(G)$ contains 64 arcs only. Therefore, the obtained walk repasses through 21 arcs. A part of the obtained walk includes the following arcs:

$$D = \{1,19,11,3,20,12,4,8,3,19,12,5,15,9,6,16,17,1,20,11,4,10,13, \ldots\}.$$

Accordingly, the initial fragment of the corresponding obtained test sequence to estimate the average power consumption of considered sequential circuit begins with the state $\boldsymbol{s}_1 = 000$ of memory elements and consists of the following input patterns:

{11,00,01,11,01,11,10,00,11,00,11,00,01,10,01,10,01,11,01,01,10,01,00,…}

Conclusion

The task of estimating average power consumption for sequential circuits is simplified when its initial automaton description is known. In this case it is shown how test sequence of input patterns that ensures estimation of energy consumption in normal mode of circuit operation can be found out.

The suggested graph models are an excellent instrument to solve the problems of generating test sequences for power consumption of sequential circuits for which there exists an initial automaton description. The process of forming such a test sequence can be viewed as traversing walks through the line digraph of the circuit automaton model.

Acknowledgement

The paper is published with partial support by the project ITHEA XXI of the ITHEA ISS (www.ithea.org) and the ADUIS (www.aduis.com.ua).

Bibliography

[Arasu, 2013] S.A.K. Arasu, F.E.Josy, N.Manibharathi, K.Rajasekaran. BPNN Based Power Estimation of Sequential Circuits. In: Intern. Journal of Advanced Research in Computer Science and Software Engineering, 2013, vol. 3, no. 11, pp. 256–260.

[Benini, 2002] L. Benini, G.De. Micheli. Logic Synthesis for Low Power. In: Logic Synthesis and Verification. Ed. S. Hassoun, T. Sasao and R.K. Brayton. Boston, Dardrecht, London: Kluwer Academic Publishers, 2002, pp. 197–223.

[Bodin, 1983] L. Bodin. A Model for Municipal Street Sweeping Operations. In: Discrete and System Models (Modules in Applied Mathematics, 1983. V. 3). 1983, pp. 76-111.

[Chou, 1996] T. Chou, K. Roy. Accurate Power Estimation of CMOS Sequential Circuits. In: IEEE Trans. VLSI Systems, vol. 4, no. 3, 1996, pp. 369–380.

[Ghosh, 1992] A. Ghosh, S. Devadas, K. Keutzer, J. White. Estimation of Average Switching Activity in Combinational and Sequential Circuits. In: 29th ACM / IEEE Design Automation Conference. Tech. Dig, June 1992, pp. 253–259.

[Harary, 1969] F. Harary. Graph theory. Addison-Wesley Publishing Company, 1969. 274 p.

[Kang, 1986] S.M. Kang. Accurate simulation of power dissipation in VLSI circuit. In: IEEE Journal of Solid-State Circuits, 1986, vol. 21, no. 5, pp. 889–891.

[Nagel, 1973] L.W. Nagel, D.O. Pederson. Spice (simulation program with integrated circuit emphasis). In: Technical Report UCB/ERL M382. EECS Department, Berkeley: University of California, April 1973.

[Najm, 1994] F.N. Najm. A survey of Power Estimation Techniques in VLSI Circuits. In: IEEE Trans. on VLSI, 1994, no. 12, pp. 446–455.

[SIA, 2014] Semiconductor Industry Association (SIA) ITRS Roadmap, available: http://www.sia-online.org/backgrounders_itrs.cfm.

[Zakrevskij, 2010] A.D. Zakrevskij. Minimization of enumeration of oriented pairs. In: Tanaevskie Chtenia: 4th Intern. Scientific Conf., Minsk, March 29 – 30, 2010, pp. 58–62.

Authors' Information

Liudmila Cheremisinova *– The United Institute of Informatics Problems of National Academy of Sciences of Belarus, principal researcher, Surganov str., 6, Minsk, 220012, Belarus; e-mail: cld@newman.bas-net.by*

Major Fields of Scientific Research: Discrete mathematics, Logic design automation

LOW-POWER SYNTHESIS OF COMBINATIONAL CMOS CIRCUITS

Dmitry Cheremisinov, Liudmila Cheremisinova

***Abstract**: An approach to logic synthesis using CMOS element library is suggested, it allows to minimize the area and the average value of power consumption of microcircuit implemented on CMOS VLSI chip. The case of synthesis of combinational CMOS networks is considered when, for the purposes of energy estimation during the synthesis process, the static method based on probabilistic properties of input signals is used. The synthesis is comprised of the technology independent phase where logic minimization and decomposition are performed on the Boolean functions equations and the technology dependent phase where mapping to a physical cell library is performed.*

***Keywords**: power consumption, low-power synthesis, CMOS circuit.*

***ACM Classification Keywords**: B.6.1 Logic design: Design Style – Sequential circuits; B.7.3 Integrated Circuits: Reliability and Testing*

Introduction

In the VLSI (Very Large Scale Integration) chip design performance, area and cost were historically the major considerations. However, in the last years power consumption has become the major issue in electronic research, it is being given increased weight age in comparison to area and speed because of three main reasons: increasing use of portable and battery operated electronic devices which have limited battery life; continuous increase in chip density resulting in VLSI circuits that contain up to hundreds of millions of transistors and topicality of high performance computing resulting in VLSI circuits that have clock frequencies in the GigaHertz range. So, the minimization of power dissipation has become a task of critical concern with the advent of high density integrated circuits and portable micro-electronic devices.

Static CMOS logic style is used for the vast majority of logic gates in digital integrated circuits because they have technological parameters and good power dissipation characteristics. Many ASIC methodologies allow only complementary CMOS circuits, custom designs use static CMOS for 95% of the logic [Zimmermann, 1997]. In CMOS based digital circuits, the most part of energy is dissipated during charging and discharging of node capacitances. To reduce the power dissipation, internal load

capacitances and switching activities of circuit gates must be lowered. Thus, at the stage of logic optimization the majority of the overall energy savings were achieved by minimizing the switching activities in the circuit. At present an increasing attention is focused not only on transistor-level design but on higher levels of abstraction because early power estimation is important in VLSI circuits, because it has a significant impact on the reliability of the circuits under design. And in the process of optimizing circuits for low power a designer is interested in knowing the effects of specific design techniques on the power consumption of the projected circuit. With the relevant information about power characteristics designer can redesign or correct a circuit in early design stages if it is found to consume more power than expected.

In the paper we consider a task of optimization of multilevel CMOS networks intended to obtain not only minimal area representation network but one optimized according to total gate switching activity, which helps in reducing the average power dissipation of the circuit. Techniques and program tools are suggested which should minimize the average power dissipation during technology-independent and technology-dependent phases of combinational logic synthesis.

Power dissipation in CMOS circuits

The power dissipation of CMOS circuits results from three parts: static, short-circuit and dynamic components. A small amount of current flow is termed the static component and is due to leakage currents (due to spurious currents in the non-conducting state) subthreshold currents. Short-circuit power happens briefly during switching and it usually accounts for 15%-20% of the overall power dissipation [Balasubramanian, 2007].

Dynamic power dissipation, also called as the switching power is related to a node capacitor which is charged and discharged. The dynamic component of power consumption normally dominates in CMOS system-on-chip and accounts for roughly 75% of the total power consumption. The dynamic power dissipation of a synchronous CMOS circuit with n gates is represented by the following approximation [Benini, 2002]:

$$P_{dyn} = \frac{1}{2} V_{dd}^2 f_{clk} \sum_{i=1}^{n} E_i C_i ,$$

where V_{dd} is the supply voltage, f_{clk} clock frequency, n the number of nodes in the circuit, C_i the node output capacitance, E_i estimates node switching activity and is called as the node transition density [Najm, 1994], that is the average number of logic transitions per a second. E_i is the transition probability at signal i, i.e., the probability that there is a $1 \rightarrow 0$ or a $0 \rightarrow 1$ transition on signal i from one clock cycle to the next.

At logic synthesis level the dynamic dissipation, that is the major source of power dissipation in static CMOS circuits, can be minimized by means of reducing switching activity (i.e. logic transitions from 0 to 1 or from 1 to 0 made by circuit nodes) in a designed logic circuit without changing its functionality. Taking out of context all the constants for used technology and capacitances that are unknown during logical synthesis we may estimate power consumption by the sum of values of switching activities of circuit nodes.

Logic synthesis transforms a circuit step by step, and each step optimizes with respect to the cost function. One transformation step (e.g., decomposition) changes only a small part of a circuit. The general power minimization strategy at logic level is to decrease $\sum_{i=1}^{n} E_i C_i$ at all stages of logical synthesis.

Switching activities will be parameters of the cost function, and after each optimization step, they must be re-estimated. Thus, an estimation of the switching activity should be accurate and fast. Accurate estimation is necessary to guide the optimization process in a proper manner. Fast estimation allows applying a large number of optimization steps and thus also contributes to the design quality.

The existing techniques for determining switching activity of nodes in a Boolean network [Najm, 1994] can be divided into two classes: statistical techniques (also called dynamic techniques) and probabilistic (or static) techniques. Statistical techniques simulate the circuit repeatedly until the power values converge to an average power, based on statistical measures. The methods are computationally intensive and are not feasible for iteratively updating switching activities as the network changes.

Probabilistic techniques propagate input statistics (probabilities) through the circuit to obtain the switching probability for each gate in the circuit. Though both the above techniques exist, the static techniques enable a quick approximate estimate of the power consumption of a digital integrated circuit at the logic level, without the need for extensive simulation.

Further in the process of logic synthesis we use the following two fast simple enough estimations of switching activity based on spatial independence assumption (circuit inputs and internal nodes are assumed to be independent). These techniques can introduce errors in the cases when input events are not independent. But empirical data shown that these errors are about hundredth parts of the calculated signal probabilities for nodes.

1) The first estimation technique is based on a zero delay model and so on assumption of signal's temporal independence assumption. For this case we are given the signal probabilities P_i (the average fraction of clock cycles in which the signal value is a logic 1) of signals on CMOS gates e_i outputs. The following equation is used to estimate the switching activity of g_i:

$$E_i = 2 P_i (1 - P_i). \quad (1)$$

If the input probabilities to a network are provided then they are propagated through to evaluate the probabilities at each node. For example, the output signal probabilities for $n(e)$-input AND and OR gates can be computed such a manner:

$$P_e^{\wedge} = \prod_{i=1}^{n(e)} P_i ; \quad P_e^{\vee} = 1 - \prod_{i=1}^{n(e)} (1 - P_i) ; \quad (2)$$

2) The second estimation technique applies a real delay model, so it computes power due to glitches too. In this case we are given transition densities A_i (the average number of signal transitions per a second) of signals. If a gate implements the function $f(x_1, x_2, \ldots x_n)$ and its inputs x_i are independent then the transition density of its output is computed as [Najm, 1994]:

$$A_f = \sum_{i=1}^{n} P(\frac{dy}{dx_i}) A_{xi} , \quad (3)$$

where the Boolean difference of the function y with respect to x_i is defined as

$$\frac{dy}{dx} = y(x = 1) \oplus y(x = 0) .$$

For a synchronous circuit with a clock period T, the relationship between transition density and switching activity is

$$A \geq \frac{E}{T} .$$

Low-power logic synthesis

In the process of logic synthesis an abstract form of desired circuit behavior (system of Boolean functions) is turned into a design implementation in terms of logic gates of CMOS cell library. The synthesis is comprised of two stages: the technology independent phase where logic minimization and decomposition is performed on the Boolean functions equations with no regard to physical properties and the technology dependent phase where mapping to a physical cell library is performed. The power dissipation of the mapped circuit is highly dependent on the structure obtained at the technology independent phase. Therefore, a power conscious design must consider power at all stages of the design process.

It is found that area minimized solutions has, most of the time, a lower power dissipation, and moreover, reducing power estimates (such as switching activity) in the synthesis process could increase the power dissipation of resulted hardware. This may be due to an increase in any of the other power components, besides switching power. Therefore, we use area criterion (for example, the number of literals) as the first one in all techniques and only then switching activity criterion.

At each synthesis phase, the problem is formulated as: given a system a set of Boolean equation (specifying two- or multi-level AND-OR circuit) and signal probabilities P_i for each input, find an implementation such that the estimates of the resulting circuit area and total gate switching activity are minimized.

Two-level minimization for low power

Two-level networks, or system of disjunctive normal forms (DNFs), are logic networks in which the primary inputs feed AND gates whose outputs are inputs to OR gates. We denote n network primary inputs as $x_1, x_2, \dots x_n$ (among which there are ones corresponding to input signals and their complements), m AND gates as $k_i = x_1 x_2 \dots x_{ni}$ and r OR gates as $f_i = k_1 \vee k_2 \vee \dots \vee k_{mi}$. If a gate implements the function $f(x_1, x_2, \dots x_n)$ and its inputs x_i are independent then the transition density of its output is computed as in the formula (2) where P_x is the equilibrium signal probability of a logic signal x defined as the average fraction of time that the signal is 1, $A(x_i)$ is the transition density of x_i that is supposed to be given before logic optimization.

For power, the switching statistics of the inputs and conjunctions must be taken into account. We will estimate the signal activity by transition density (2). In the case of the conjunction $k = x_1x_2\dots x_n$ the transition density is computed easy enough:

$$\frac{dy}{dx_i} = x_1\dots x_{i-1}\dots x_{i+1}\dots x_m ,$$

$$A_k = \sum_{i=1}^{n} P_{x1}\dots P x_{i-1} P_{xi+1}\dots P_{xn} A_{xi} .$$

Assume for the sake of simplicity that transition densities of inputs are the same as their signal probabilities:

$$A_{xi} = A_{\overline{xi}} = P_{xi} .$$

The proposed minimization techniques are extensions of known methods of Boolean function minimization by adding heuristics that turn the minimization process towards lowering the power dissipation in the sought CMOS-circuits. The results of computer experiments are given in [Cheremisinov, 2011], which allow to evaluate power driven minimization influence on power dissipation of resulting CMOS-circuits.

Power consumption of two-level network as whole is estimated by the sum of estimations for all terminals (AND gates outputs and primary inputs) they depend on transition densities (2) depending on signal probabilities. Thus, the proposed methods of minimization of system of Boolean functions try to find out only prime implicants with less switching activities to minimize the sum of estimations for all terminals of two-level network under construction.

The two-level minimization problem that is obtained by appending the don't care conditions to the function of a node is typically solved by a heuristic algorithms. The set of primes of a function can be partitioned into three classes: essential, partially redundant and totally redundant primes. The possible source of power savings during Boolean functions minimization concerns only partially redundant primes. The typical basic operations of any minimization algorithm are: cube reduction and expansion, searching for irredundant cover. So, they should be done such a way to lead potentially to power reduction. In the first, the cubes of the function being minimized are reduced as much as possible; in the second step they are expanded so that as many cubes as possible are covered by other cubes and can therefore be dropped. In the third and final step, an irredundant set of cubes that cover the function is extracted from the set of those that have survived the expansion phase. In expanding a cube, two objectives must be taken into account. The first of then is the quality of the cube taken in isolation. For power, the switching statistics of the inputs must be taken into account. In the case of power minimization, the value of an expansion depends not only on the number of cubes that should be covered, but also on the activity of the cubes that are eliminated (using (2)) because they are covered.

The maximal expansions of a single cube do not depend on the order in which the cubes of cover of network S are processed. However, the order in which the cubes are processed for expansion is important because of the effect it has on which cubes are covered and hence dropped: Expanding a cube too early may prevent another cube from covering it. In order to minimize power consumption, the switching activity of each reduced cube is computed and then they are processed according to increasing switching activities. Thus, the cubes with high switching activity are kept last, in the hope that some other cubes will expand to cover them.

Therefore, in expanding a cube, two objectives are taken into account: 1) to reduce the activity of resulting prime we try first of all to exclude the most active literals; 2) to cover the most active cubes

because they will be eliminated. To ensure the first demand we put literals in order of decreasing their switching activities to analyze for eliminating first the most active literals. To ensure the second demand we compute the switching activity of each cube to be expanded and then process them according to increasing switching activities. Thus, the cubes with high switching activity are kept last, in the hope that some of them will be covered (and hence dropped) when expanding other cubes. In reducing a cube not any but the least active literals are appended to it. When searching for irredundant cover we try to find a set of cubes with the least total cube switching activities.

The developed program set implements a set of methods that give suboptimal solutions of minimization task [Cheremisinov, 2011]. The program set allows varying:

1) the object of minimization: a completely or incompletely specified Boolean function; a system of completely or incompletely specified functions;

2) the method of minimization: interval competing method [Toropov, 2001], modified method ESPRESSO, iterative method, fast-acting one passing method [Cheremisinov, 2011];

3) individual or joint minimization of functions in a system;

4) minimization with taking into account the output polarity assignment, limiting runtime and others.

In Table 1 some experimental results showing the efficiency of the used techniques aimed at switching activity reduction. Here two versions of interval competing method of minimization was investigated; the initial method [Toropov, 2001] and its low-power modification. More details can be found in [Cheremisinov, 2011]. Here:

n, m, k are the number of inputs, outputs and conjunctions of the DNF system under processing;

kmin1, kmin2 and l1, l2 are the number of conjunctions and literals of the minimized DNF systems;

Ps1, Ps2 are switching activity estimations calculated as $\sum_{i=1}^{n} E_i C_i$ *of the minimized DNF systems;*

t1, t2 are computing times of two investigated methods.

When minimizing Boolean systems the following values of signal probabilities were used:

$p_1 = 0.10$; $p_2 = 0.13$; $p_3 = 0.16$; $p_4 = 0.19$; $p_5 = 0.22$; $p_6 = 0.25$; $p_7 = 0.28$; $p_8 = 0.31$; $p_9 = 0.34$; $p_{10} = 0.37$;

$p_{11} = 0.40$; $p_{12} = 0.43$; $p_{13} = 0.46$; $p_{14} = 0.49$; $p_{15} = 0.52$.

Multi-level minimization for low power

The paper focuses on the task of generating multi-level logical network that is best suited for the technology mapping process for CMOS gate arrays. The proposed methods [Cheremisinova, 2013] are targeted minimization of integrated microcircuit area implemented on CMOC chip and total value of gates switching activity. The methods take into account the specific features of the CMOS cell library during technology independent decomposition in terms of chosen primitive base functions (NOTs, ANDs, ORs). And in contrast to existing approaches it is proposed to decompose a Boolean network, to be mapped, into a *k*–bounded network where the number of fan–ins of each node is less than or equal to the greatest fan–in of the gates specified by the structure of the library cells. As the base functions in our case it might to choose elementary functions: NOT, *k*AND, *k*OR realized by component gates of primitive cells.

Table 1. The investigation results

DNF system	DNF system under minimization			Results of usual minimization				Results of low-power minimization			
	n	m	k	k_{min}^1	l^1	P_s^1	t^1	k_{min}^2	l^2	P_s^2	t^2
b12	15	9	431	45	286	61.8653	0.06	44	267	**61.8647**	0.06
in0	15	11	138	111	1348	**385.478**	0.01	111	1351	385.894	0.01
life	9	1	512	84	756	233.603	0.01	84	756	**230.48**	0.01
mlp4	8	8	256	160	1574	355.837	0.03	160	1577	**355.488**	0.03
root	8	5	256	57	430	**93.8343**	0.01	58	393	95.2714	0.01
tms	8	16	30	30	484	69.3707	0.00	30	484	**69.3705**	0.00
z9sym	9	1	420	90	630	**178.538**	0.01	101	707	197.323	0.04
ADDM4	9	8	512	249	2477	613.69	0.09	249	2485	**612.889**	0.09

Factorization technique is a key tool in facilitating multilevel synthesis. Finding a minimum factored expression is a cumbersome task. The proposed heuristic methods consist in the following two phases.

1) Nontrivial factorization of a system of Boolean expressions via extraction of common single- or multiple-term factors (subexpressions) for two or more Boolean expressions. Each time when factoring the system of Boolean expressions (conjunctions or disjunctions), the best factor is chosen according cost and power estimates.

The cost estimate computes the gain in "area": $T_s = c(s)(|U_s| - 1)$, where $c(s)$ is the size of the factor (number of literals) s and $|U_s|$ is the number of factored expressions. Nontrivial factorization supposes that only those factors are used for which we have $|U_s| > 1$ and $c(s)$ exceeds some predefined value.

The power gain of the factor s is proposed to estimate as:

$$P_s = \sum_{z_i \in s} E_{z_i} (|U_s| - 1),$$

where E_{z_i} is switching activity of a literal $z_i \in s$, computed according to (1) and (2).

As the result of the factorization a multilevel representation of a system of Boolean, an expression is found that has minimal number of literals but still it has some "long" conjunctions and disjunctions.

2. Factorization of each separate Boolean function independently via searching for common literals of its expression terms and then partitioning the rest "long" terms. The procedure is based on the following decomposition of a DNF D of every function: $D = k(A) + B$, where D, A, B are some DNFs and k is conjunction. The core of k is some "best" literal which enters in maximal number of conjunctions and which has the maximal switching activity. The last demand allows decreasing the load on the active node and promoting the active node to be as close as possible to the circuit output.

As the result of the factorization, a multilevel k–bounded network will be found.

Technology mapping

The resulting Boolean network consists of some non-overlapping multi-level k–bounded subnetworks with the only output that are then covered. These subnetworks consist of AND, OR, NOT gates. Each

library cell is decomposed into a superposition of the same functions. The next step is the technology mapping, it is performed by replacing subnetworks of the Boolean network with cell library instances.

The idea of the implemented method of technology mapping [Cheremisinova, 2010] is to cover the generated AND-OR network using elements of CMOS sell library. Alongside with the area criterion the energy consumption criterion is used too. The general idea of power saving is the known one – to hide high switching nodes within complex elements. So, the variants of covering are compared on the cost of cover (the ratio of the covered fragment cost according to Quine to the covering element cost) and the total switching activity of gates falling within the library element.

In Table 2 some experimental results showing the efficiency of the used synthesis techniques aimed at switching activity reduction. Here the minimization of DNF systems was fulfilled using the modified ESPRESSO method. The combinational circuits were implemented using elements of the native CMOS library. Two methods of implementation of technology independent synthesis phase have been used. The first of them is based on usual approach where a Boolean network, to be mapped, is decomposed using two input NANDs. And the second method takes into account the specific features of the used CMOS cell library constructs the circuit with factorization techniques mentioned above.

Table 2. *The experimental results of synthesis of combinational circuits*

DNF system	The of inputs,outputs conjunctions	DNF system minimization			Synthesis of multilevel circuit of CMOS library elements					
					The first method			The second method		
		p	*s*	*t*	*p*	*s*	*t*	*p*	*s*	*t*
br1	12, 8, 34	108	279	<1	141,34	668	3	54,11	386	6
br2	12, 8, 35	78,5	212	<1	115,04	526	2	48,29	348	4
in0	15, 11, 138	448,5	1118	<1	438,03	1956	2	229,2	1508	5
in2	19, 10, 137	596,5	1469	<1	451,38	2040	4	248,37	1726	5
mlp4	8, 8, 256	396	979	<1	324,68	1418	3	255,79	1406	6
root	8. 5, 256	150	393	<1	155,84	672	3	141,55	692	5
tms	8, 16, 30	108,5	471	<1	302,98	1276	3	139,1	916	5

z9sym	9, 1, 420	258,00	602	<1	160,31	678	2	136,72	702	5
GenP1	20, 4, 50	315,5	739	<1	548,69	2794	3	228,11	1374	7
GenP2	30, 10, 100	768	1733	1	1331,12	6790	3	631,87	4090	5
GenP3	30, 10, 300	2215,5	5052	14	3807,66	19502	4	1545,63	10418	11
GenP4	30, 8, 400	1798,5	4737	13	4232,16	21060	4	1583,5	9480	11
GenP5	20, 6, 400	1176,5	3029	15	1822,95	8526	4	945,59	5270	6
GenP6	20, 6, 400	1076,5	2948	15	1961,76	9016	5	945,32	5154	7
GenP7	30, 12, 50	244,5	558	<1	328,99	1632	2	252,52	1376	4
GenP9	30, 12, 400	1908,5	4667	7	3442,93	17268	5	1732,14	10366	10
GenP10	30, 12, 700	2314,0	5759	17	3238,65	15420	6	1776,18	9736	18
GenP13	30, 5, 600	2072,0	5492	19	3965,55	18948	8	1577,43	8762	44
GenP14	30, 5, 500	1710,5	4291	13	2618,67	12546	4	1343,04	7396	9
GenP15	30, 5, 400	2184,5	5213	7	3820,06	19416	4	1708,22	10628	12
GenP17	30,10,400	1610,5	4011	9	2785,49	13702	4	1530,39	8776	12
GenP22	24, 7, 790	2356	6223	19	4010,45	18762	10	1852,17	10588	18

Here, in Table 2 p and s are the switching activity and the area (estimated by the transistor number) of the circuit. t is program execution time in seconds on the computer with processor Intel i5-2400@3,1 GHz and 2,99 ГБ.

From the experimental results follows that taking into account the specific features of used CMOS cell library during technology independent decomposition allows synthesize better circuit variants. More details can be found in [Kirienko, 2015].

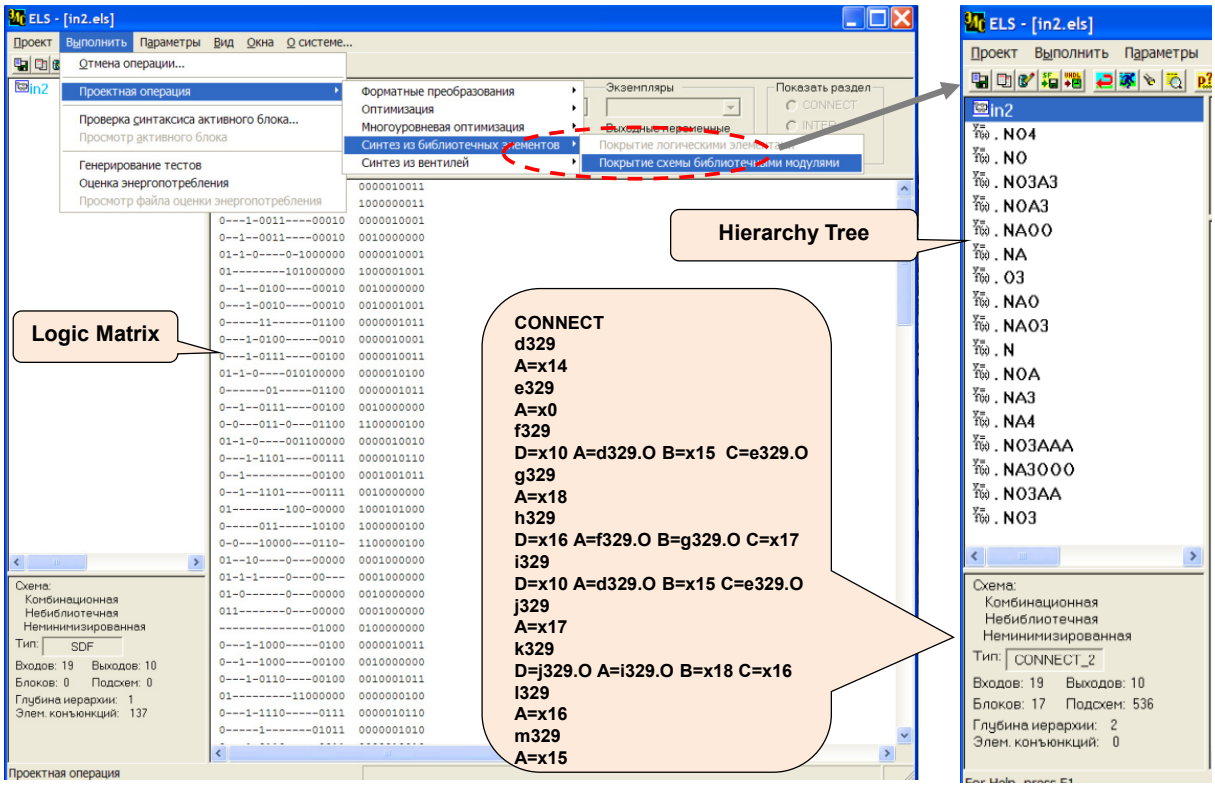


Figure 1. Technology mapping CAD system

Conclusion

The program implementations of the proposed methods are included as project operations in the software system for energy-saving logical synthesis [Bibilo, 2012] developed in the United Institute of Informatics Problems of NAS of Belarus. The system is intended for cell library design automation of custom very large-scale integration CMOS circuits. Figure 1 shows the typical window of the system in the case when multi-level *k*–bounded AND-OR network is covered with CMOS cell library instances. The estimations of complexity and power consumption are accepted as optimality criteria when designing CMOS circuits.

Acknowledgement

The paper is published with partial support by the project ITHEA XXI of the ITHEA ISS (www.ithea.org) and the ADUIS (www.aduis.com.ua).

Bibliography

[Bibilo, 2012] P.N. Bibilo, L.D. Cheremisinova, S.N. Kardash, N.A. Kirienko, V.I. Romanov, D.I. Cheremisinov. Low-Power Logical Synthesis of CMOS Circuits Automation. In: Programnaia indzeneria, 2013, no 8, pp. 35–41 (in Russian).

[Balasubramanian, 2007] P. Balasubramanian, C. H. Narayanan, K. Anantha. Low Power Design of Digital Combinatorial Circuits with Complementary CMOS Logic. In: Intern. Journal of Electronics, Circuits and Systems, 2007, vol. 1, no 1, pp. 10–18.

[Benini, 2002] L. Benini, G.De. Micheli. Logic Synthesis for Low Power. In: Logic Synthesis and Verification (Eds. S. Hassoun, T. Sasao, R.K. Brayton): Boston, Dardrecht, London: Kluwer Academic Publishers, 2002.

[Cheremisinov, 2011] D.I. Cheremisinov, L.D. Cheremisinova. Low Power Driven Minimization of Two-Level CMOS Circuits. In: Informacionnie technologii, 2011, no 5, pp. 17–23 (in Russian).

[Cheremisinova, 2010] L.D. Cheremisinova. Low-power synthesis of combinational CMOC networks. In: Informatics (UIIP of NAS of Belarus), 2010, no 4, pp. 112–122 (in Russian).

[Cheremisinova, 2013] L.D. Cheremisinova, N.A.Kirienko. Low power driven synthesis of multi-level Logical circuits. In: Informacionnie technologii, 2013, no 3, pp. 8–14 (in Russian).

[Kirienko, 2015] N.A. Kirienko, D.I. Cheremisinov, L.D. Cheremisinova. Optimization of multi-level representations of logic circuits to reduce power consumption and VLSI chip area. In: Vesti of NAS of Belarus. Ser. physics-mathematics sciences, 2015, no 2, pp. 103–111 (in Russian).

[Najm, 1994] F.N. Najm. A survey of Power Estimation Techniques in VLSI Circuits. In: IEEE Trans. on VLSI, 1994, no 12, pp. 446–455.

[Toropov, 2001] N.R. Toropov. Minimization of a system of Boolean with output polarity assignment. In: Proc. of the 4th Intern. Conf. on Computer-Aided Design of Discrete Devices, Minsk, Rep. of Belarus, Nov.10–12, 2001, vol. 2, pp. 92–104 (in Russian).

[Zimmermann, 1997] R. Zimmermann, W. Fichtner. Low-power logic styles: CMOS versus pass-transistor logic. In: IEEE Journal of Solid-State Circuits, 1997, vol. 32, no. 7, pp. 1079–1090.

Authors' Information

Dmitry Cheremisinov *– The United Institute of Informatics Problems of National Academy of Sciences of Belarus, leading researcher, Surganov str., 6, Minsk, 220012, Belarus; e-mail: cher@newman.bas-net.by*

Major Fields of Scientific Research: Logic design automation, System programing

Liudmila Cheremisinova *– The United Institute of Informatics Problems of National Academy of Sciences of Belarus, principal researcher, Surganov str., 6, Minsk, 220012, Belarus; e-mail: cld@newman.bas-net.by*

Major Fields of Scientific Research: Discrete mathematics, Logic design automation

EMPLOYMENT OF BUSINESS INTELLIGENCE METHODS FOR COMPETENCES EVALUATION IN BUSINESS GAMES

Olga Vikentyeva, Alexandr Deryabin, Nadezhda Krasilich, Lidiia Shestakova

Abstract: *This research involves justification of Business Intelligence methods usage in order to analyze the outcome of computer-based business games. In such games, it is necessary to evaluate efficiency of player's actions that is associated with the development of professional competences acquired by participants as well as the game effectiveness in regard to the qualitative development of the specified competences. Computer-based business games development and conduction is performed within the source environment of "Competence-based Business Games Studio". The challenge of player's competences evaluation and the game effectiveness is tackled by Business Intelligence methods implementation into analysis subsystem of "Competence-based Business Games Studio". The analysis of participants' actions allows getting player's behavior characterization based on the information about the results of all games that the player participated. In order to guide the course of the business game, analysis is conducted on the basis of all players' actions during all games and the reference models of the games. The paper defines data source subsystems and the main measurements; it also considers the process of data warehouse development and the requirements to the output of analysis subsystem.*

Keywords*: business intelligence methods, data warehouse, competencies, active learning methods, business-game.*

ACM Classification Keywords*: K.3 Computers and Education: K.3.2 Computer and Information Science Education – Information systems education. K.4 Computers and Society: K.4.3 Organizational Impacts – Employment. I. Computing Methodologies: I.2 Artificial Intelligence: I.2.1 Applications and Expert Systems – Games.*

Introduction

Currently, significant changes in education area are taking place. These changes are triggered by new requirements to graduates, intensification of labor force market competition, emergence of new professions, spread of information technologies and international integration in education sector. This makes traditional approaches to the educational process impossible to use.

Therefore, competence building is becoming the main purpose of education. Competency is defined as the ability to implement knowledge, skills and personal qualities to succeed in a certain professional area. For competences' formation it is necessary to use new forms of education, that improve cognitive, communicative and personal student activity. In this case, active forms of education such as training, role-playing games, case study, business games, computer simulators are gaining popularity. The thematic justification is evidenced by numerous publications of Russian and foreign authors [Biggs, 1990], [Draganidis, 2006], [Aldrich, 2009], [Girev, 2010], [Vikentyeva, 2013], [Bellotti, 2013], [Bazhenov, 2014].

The standard approach to competences' evaluation process implies testing and estimation of its results. However such approach leads to low accuracy of competences' level determination [Bellotti, 2013].

At the same time, there are alternative methods of competences' assessment. Nowadays there is a great interest in business games that are included in the subset of serious games. Business games aim to build competences' within educational process. A review of existing researches in evaluation of the business games efficiency and its' participants actions is conducted in the research of [Bellotti, 2013].

In business games it is necessary to evaluate the players' actions efficiency that is associated with the results of competences' development. Participants' actions evaluation process will allow to design individual educational trajectory for each player, as well as to provide feedback to the player. Furthermore, the game quality evaluation is required in order to conclude whether the game is suitable for the development of given set of competences.

The "serious" games adopted the following approach to competences formation assessment. The game is divided into several levels and each level corresponds to some level of competences' development. Therefore, the gameplay of such a game should comply with educational goals. Thus, the following aspects can be identified: a participant actions assessment; a participant actions characterization that includes the game (scenario, level of complexity and so on) characterization and characterization of a player (individual information); inclusion of assessment into a game [Bellotti, 2013].

Such an approach, however, is prone to the following issues:

– The difficulty of obtaining personal information, for instance, neurophysiological signals analysis required the use of complex equipment.

– The complexity of the assessment mechanism implementation into a game, i.e. the game scenario has to include training elements and the evaluation of training results.

Therefore the challenge of the research is identification of evaluation methods of competences acquired by a player during a business game conduction.

Business Intelligence Methods using in Educational Processes

Business Intelligence (BI) is a direction in data processing, that helps to make a decisions in different situations based on meaningful information for given domain. BI methods include an information provision, integration and analysis tools.

Information analysis methods encompass the following:

– OLAP (Online Analytical Processing) represents multidimensional analysis where each dimension includes data consolidation consisting of sequential generalization levels, each level corresponds to more aggregated for the relevant dimension data. OLAP provides the following features to operate with multi-dimensional data: flexible review of information, optional data slices, drill down, drill up, rotation, inter-temporal comparisons.

– Data Mining is used to highlight the significant patterns in data that is stored in data warehouses. Data Mining is based on statistical modeling, neural networks, genetic algorithms, etc. Problems solved by Data Mining Methods include: classification – objects (observations, events) assignment to one of previously known classes, regression including forecasting tasks, identification of dependency in continuous variables, clustering – objects (observations, events) grouping based on data (properties) describing the nature of these objects, association – identification of patterns between the time-related events, variance analysis – identifying the most atypical patterns.

Educational Data Mining (EDM) – DM methods implementation for analysis of data, generated by educational processes in order to solve educational challenges such as adaptation of the curriculum for a particular learner, improving the understanding of the learning process and so on.

As of today, there is a number of researches in EDM domain. However, these researches have a fairly wide range of goals, i.e. authors use similar data sets but they pursue completely different goals, for instance:

- Schedule planning and scheduling.

- Preparation of recommendations for students.

- Predicting student performance.

- Determination of undesirable behavior of students.

- Splitting students into groups according to their personal qualities.

- Educational software development.

The figure 1 represents the concept of most such systems [Hung, 2012]. This algorithm can be applied in absolutely different studies, for example, determination of factors that have influence on students' progress [Jeong, 2013], failure on exam prediction [Sahedani, 2013], study of on-line learning improvements ways [Hung, 2012], etc.

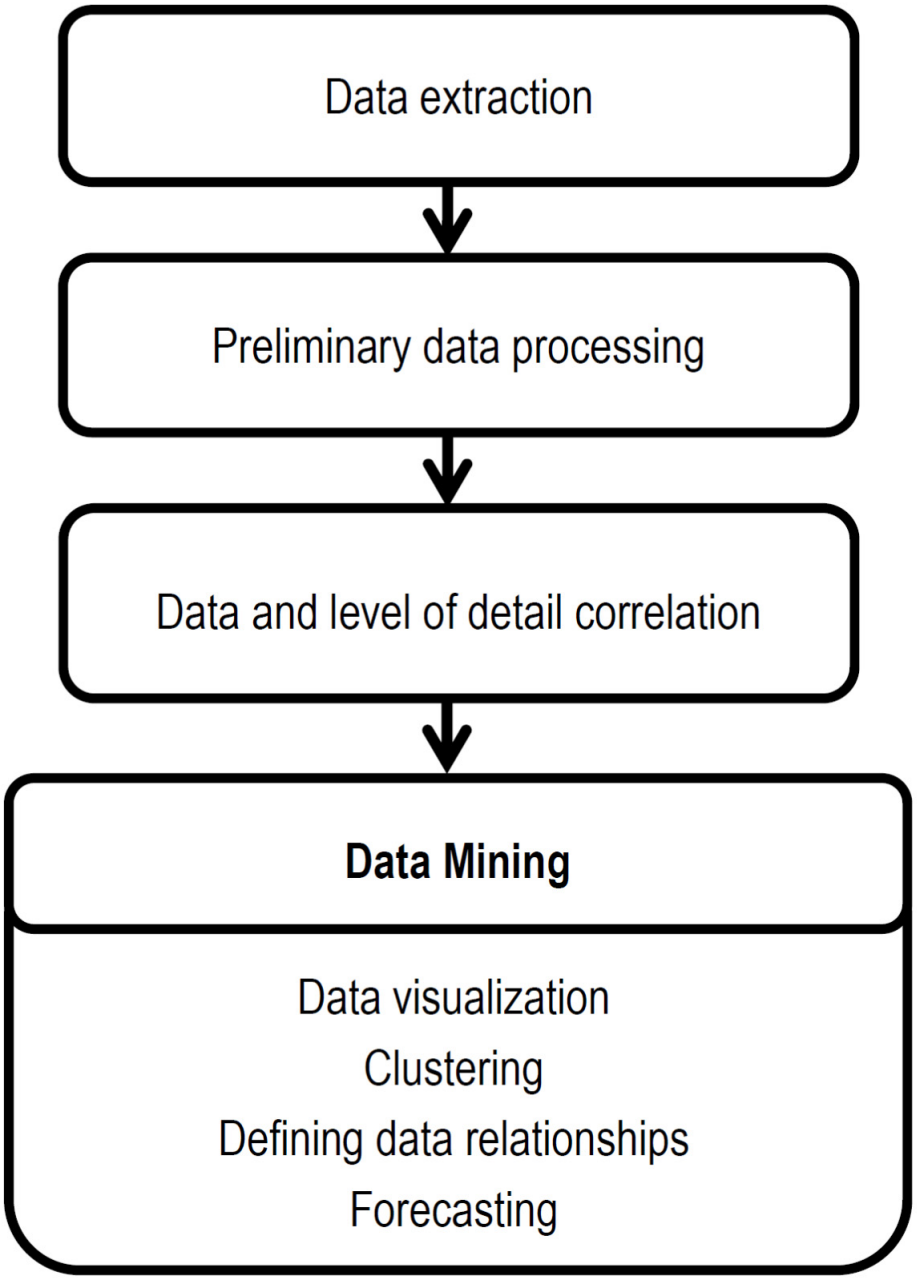


Figure 1. Generalized algorithm of EDM-systems performance

Information System of Business Games Design and Conduction

Business games development and conduction may be carried out with use of information systems. Competence-based Business Game Studio (CBGS) information system includes tools for game designing, game conduction, measurement of obtained competences, game process monitoring, player's actions analysis and business game analysis, player's behavior correction and game scenario correction [Vikenteva, 2013].

Competence is a combination of knowledge, skills, experience and personal characteristics of an employee that are necessary to carry out his professional duties successfully [Kozodaev, 2015].

CBGS analysis subsystem should provide assessment of player's competences (knowledge, skills and experiences) based on his actions within Decision Making Points during game conduction [Formalization of Domain]. Decision Making Points allow participant to choose the course of action within a business process. The design subsystem includes the reference model of game, against which actual data of player's actions may be compared. CBGS information contours associated with analysis subsystem are represented on figure 2.

Analysis subsystem should perform business game analysis based on comparison of the results of different participants' actions. Based on data obtained from analysis subsystem the correction of player's behavior may be performed. This allows developing individual educational trajectory for each player (contour A). Moreover, correction subsystem allows changing elements of business games in order to reset it for more qualitative development of defined competences (contour B).

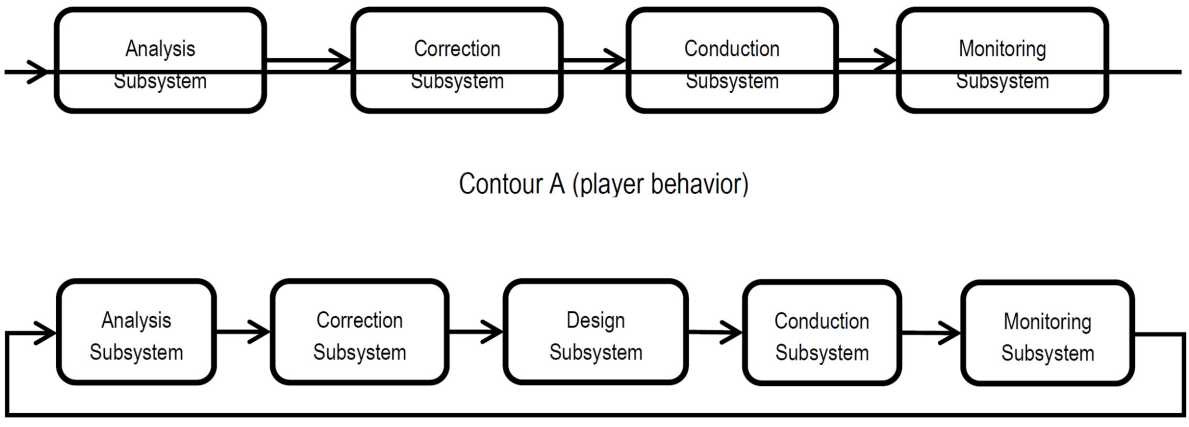


Figure 2. CBGS information contours related to analysis

Analysis Subsystem should perform two major analysis procedures:

1. Player's actions analysis that allows providing player's characterization based on all business games, which the player participated.

2. Game analysis to its correction in the case of bottlenecks identification. For such an analysis input data is information about all players' actions during all games and the reference models of all games.

In order to develop analysis subsystem it is required to design data warehouse and define dimensions for player's and game's assessments.

Competence includes two major characteristics: knowledge and skills. Experience assessment will not be performed within business game. Therefore data mart designing for player's actions evaluation will be multidimensional and will consist of as least two dimensions. There is no limit to the amount of characteristics and key figures. They must be defined on the basis of business processes' characteristics.

The results of conducted games will be divided into clusters for the game bottlenecks identification. However in order to define the reason of these bottlenecks or errors it is required to set some patterns. This may be provided by usage of Data Mining method – instructed classification.

Initially results of all conducted games will be divided into clusters but in order to identify bottlenecks average clusters' characteristics will be compared with objects from the learning sample.

Thus in the process of CBGS analysis subsystem development the following Business Intelligence methods will be implemented (table 1, table 2):

– Slice and Dice.

– Drill Down/Up.

– Pivoting.

– Nesting.

– Clustering.

– Instructed Classification.

Table 1. Formalized representation of Business Intelligence method group selection

		Requirements for the Analysis Subsystem			
		Multi-dimensional data marts appliance	Patterns determination during game conduction	A comparison of patterns with the specified templates	Search of game process deviances from reference models
Groups of Business Intelligence Methods	Complex Analysis	yes	no	no	no
	Forecasting	no	no	no	no
	Business Forecasting Model	no	no	no	no
	Data Mining	yes	yes	yes	yes

Table 2. Formalized representation of Business Intelligence method selection

				Requirements for the Analysis Subsystem			
				Multi-dimensional data marts appliance	Patterns determination during game conduction	A comparison of patterns with the specified templates	Search of game process deviances from reference models
Business Intelligence Methods	Complex Analysis	Slice and Dice		yes	no	no	no
		Drill Down/Up		yes	no	no	no
		Pivot		yes	no	no	no
		Nesting		yes	no	no	no
	Data Mining	Clustering		no	yes	no	no
		Classification	instructed	no	no	yes	yes
			uninstructed	no	no	no	no
			self-learning	no	no	no	no
		Association		no	no	no	no
		Sequencing		no	no	no	no
		Regression		no	no	no	no
		Forecasting		no	no	no	no

Source Data for Analysis Subsystem

Source data for player's actions analysis are stored in following CBGS subsystems (fig. 3):

– Design Subsystem (contains competences' matrix for each operation and the reference models of business processes)

– Monitoring Subsystem (contains the results of conducted games by a player based on data acquired from Conduction Subsystem).

Source data for game analysis are formed by the Design Subsystem, including Competence Designing Module.

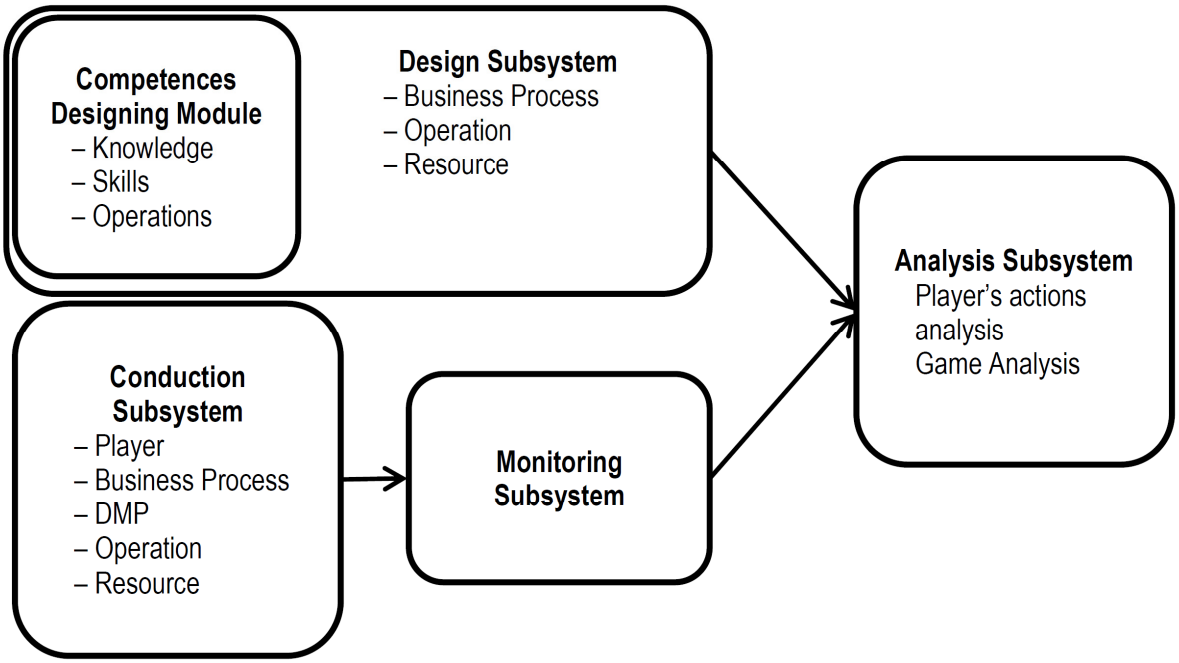


Figure 3. Source Data for Analysis Subsystem

The process of competence planning is resolved to a composition of business process operations' coverage matrix of competences. The matrix has to associate operations and competences (figure 4). However, it is important to define the extent to which the competence is obtained by a player and which knowledge and skills he has. In order to create this possibility, multi-dimensional array of competencies has to include resources, since they allow defining if a player possesses required set of knowledge and skills to perform operations within a business process (player knows which resources are needed and how to implement them).

			Competence 1				Competence 2				Competence u			
			Knowledge 1.1	Knowledge 1.r	Skill 1.1	Skill 1.l	Knowledge 2.1	Knowledge 2.q	Skill 2.1	Skill 2.w	Knowledge 1.1	Knowledge u.r	Skill u.1	Skill u.p
Business Process 1	Operation 1.1	Resource 1.1.1	1		1									
		Resource 1.1.2		1										
		Resource 1.1.x				1								
	Operation 1.2	Resource 1.2.1									1			1
		Resource 1.2.2											1	
		Resource 1.2.y					1		1			1		
	Operation 1.n	Resource 1.n.1						1						
		Resource 1.n.2								1				
		Resource 1.n.z												
Business Process t	Operation t.1	Resource 1.1.1	1		1									
		Resource 1.1.2		1										
		Resource 1.1.x				1								
	Operation t.2	Resource t.2.1					1							
		Resource t.2.2.						1	1					
		Resource t.2.f								1				
	Operation t.k	Resource t.k.1									1			
		Resource t.k.2										1	1	
		Resource t.k.g												1

Figure 4. Data Structure for Competences Planning

Figure 5 represents a simplified diagram of the Design Subsystem database that does not include secondary attributes; it means that each entity contains only those attributes that are meaningful for the Analysis Subsystem development.

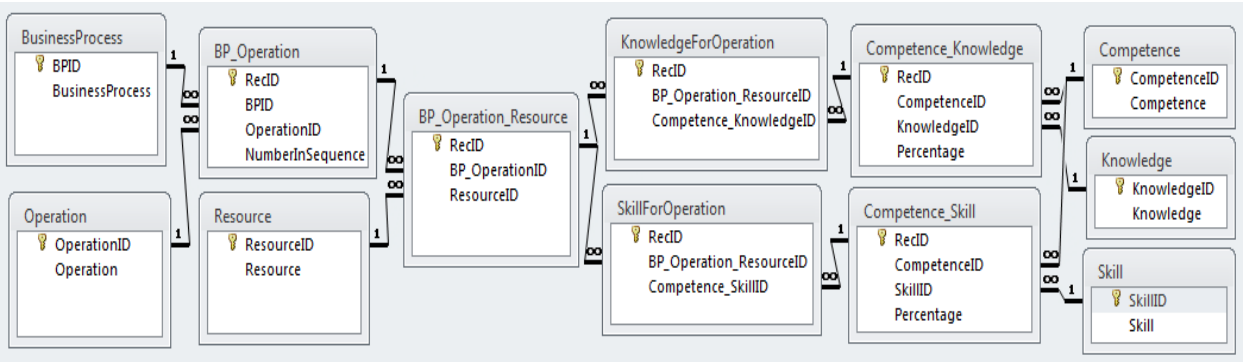


Figure 5. Database diagram for storage information about game

The simplified database diagram for storage of game results data can be seen on Figure 6. It represents log of operational data of conducted games by players.

Date and time of start and end of a game are stored in table “BP_Player”. Based on this data it can be determined whether the player has already played this game, and if he did, then how many times.

Table “BP_Player_Operation” stores actual number of operations performed by a player within a certain game. Thus, it is possible to define whether a player performs actions in the right order. Number of operations is derived from information about Decision Making Points and chosen resources.

Table “BP_Player_Operation_Resource” stores a set of resources chosen by a player in order to perform a certain operation within a business process. Data from this table together with the data from Competences’ Planning DB allow defining the extent of certain competences obtaining by a player.

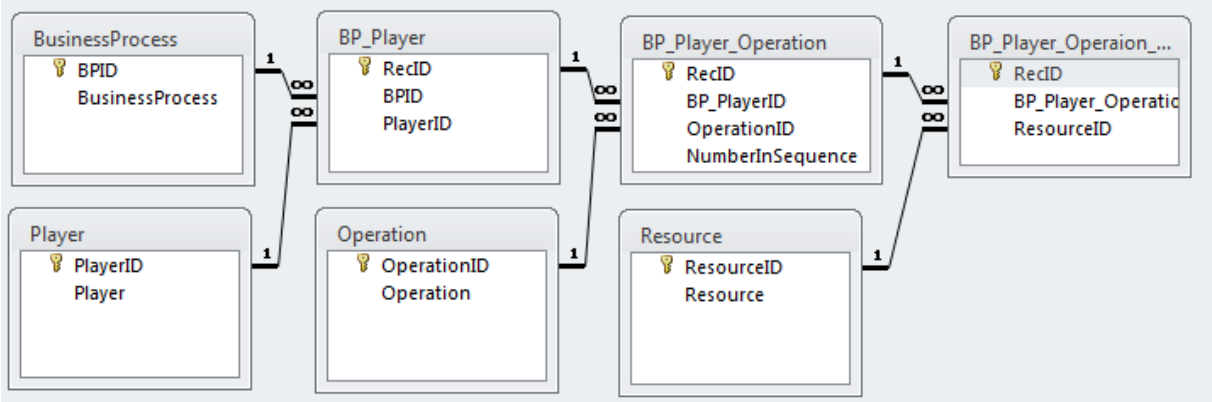


Figure 6. Results of Conducted Games Database Diagram

Databases of all subsystems are developed in DBMS MS SQL Server Management Studio.

As data of multi-dimensional cubes of data warehouse will be loaded from several subsystems (from several databases), requirements for source systems’ data models and data itself were formulated. Loaded data should require minimum change in steps of pretreatment and transformation.

Pretreatment of data implies data scrubbing from noise, getting rid of the redundancy and the selection of significant characteristics. Data transformation is required to bring the data to fit for further analysis mind.

The Analysis Subsystem Data Warehouse Designing

Based on source data that may be extracted from the Design Subsystem and the Monitoring Subsystem, it was revealed that players' actions assessment can be performed using the following criteria:

– Correspondence of operations sequence, performed by a participant during a game to the reference model.

– Player's expertise.

– Satisfactory time of game duration.

According to these criteria the data warehouse will consist of dimensions represented in table 3. Key figures (facts, measures) are shown in table 4.

Table 3. Dimensions

Name	Type	Length
A Game Duration	Time	-
Number of a Game for a Player	Integer	-
Business Process	String	255
Operation	String	255
Resource	String	255
Competency	String	255
Competency Type (Knowledge or Skill)	String	6
Knowledge/Skill Name	String	255
Operation Number of the Reference Model	Integer	-
Actual number of operation	Integer	-

Table 4. Key figures (facts, measures)

Name	Type	Unit of measure
The Deviation in Sequence	Integer	-
Operation Performance Indicator	Integer (0 or 1)	-
Formed Percentage of Knowledge/Skill	Double	Percentage

The physical model of data warehouse designed for player's actions assessment was developed within MS SQL Server Management Studio 2012. The diagram represents a database developed by the schema "Snowflake". The physical model of the cube "Assessment of Player's Actions" is presented on Figure 7.

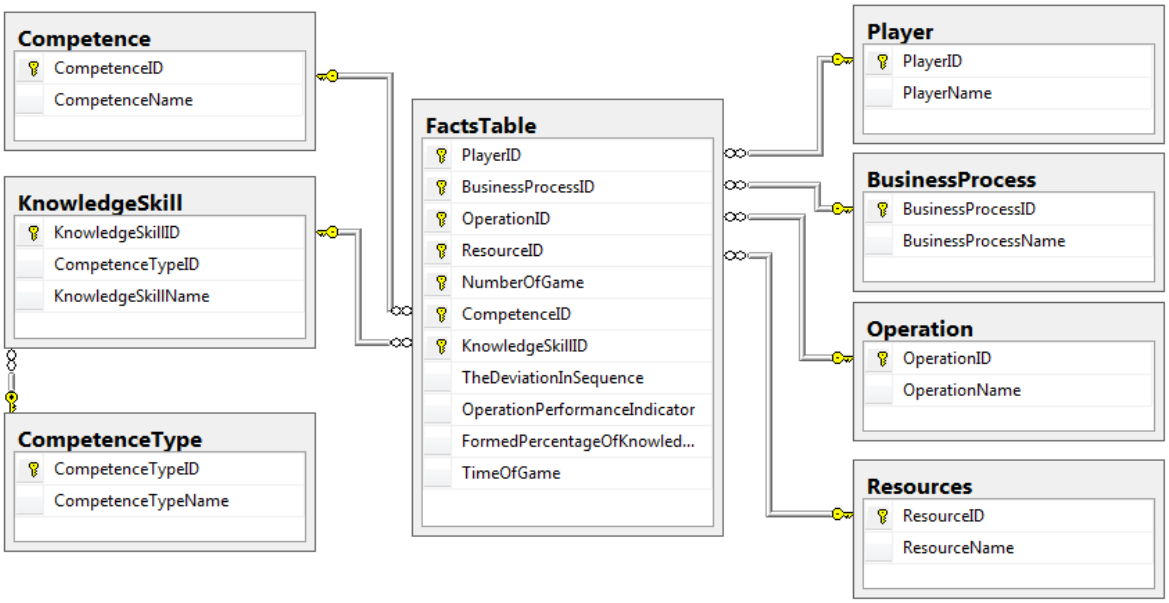


Figure 7. The Data Warehouse Diagram «Player's Actions Assessment»

The diagram contains a fact table and 7 tables of dimensions. The fact table includes a composite key containing references on all cube dimensions.

Conclusion

The research is devoted to the challenge of evaluation of competences obtained by participants during the computer-based business game playing and assessment of business game itself. Computer-based business game is designed and implemented within tool environment "Competence-based Business Game Studio" consisting of several subsystems. The paper describes the Analysis Subsystem development process. The subsystem is used to evaluate player's actions and business game quality.

During the research, data stored in source systems were considered. Based on this data major characteristics of the data warehouse were revealed. Moreover, the requirements for output data of the Analysis Subsystem were formulated. This allowed defining a set of needed key figures. The criteria for player expertise assessment were derived in the process of requirements development for competence-planning-related data storage.

During Business Intelligence methods' analysis it was revealed that in order to implement functional requirements to the Analysis Subsystem it is necessary to implement all tools of Complex Analysis as the designed data warehouse is multi-dimensional and different reports will be formed. To reveal bottlenecks in business processes and the business game instructed classification and clustering (Maximin Algorithm) will be used.

The data warehouse for player's actions assessment was designed and load, transformation flows were developed.

Further work includes OLAP-cube "Bottlenecks of the Business Game" development and clustering and classification methods implementation in order to create analytical reports within the task.

Bibliography

[Aldrich, 2009] Aldrich, C. Virtual worlds, simulations, and games for education: A unifying view. *Innovate, 5*(5). Retrieved June 2, 2009.

[Biggs, 1990] Biggs William D. Introduction to Computerized Business Management Simulations // Guide to Business Gaming and Experiential Learning / авт. книги Gentry J. W.. - London : Nichols/GP Publishsing, 1990.

[Draganidis, 2006] F. Draganidis. Chamopoulou P and Mentzas G An Ontology Based Tool for Competency Management and Learning Paths 6th International Conference on Knowledge Management I-KNOW 06, Special track on integrating Working and Learning, 6th September 2006, Graz, (2006).

[Belotti, 2013] Bellotti F. and others. Educational Data Mining: Assessment in and of Serious Games: An Overview. Advances in Human-Computer Interaction. Hindawi, 2013.

[Hung, 2012] Hung, J.L. and others. An Educational Data Mining Model for Online Teaching and Learning / J.L. Hung, K. Rice, A. Saba // Journal of Educational Technology Development and Exchange. – 2012.

[Jeong, 2013] Jeong, H. Educational Data Mining. How Students' Self-motivation and Learning Strategies Affect Actual Achievement // Department of Computer Science, Indiana University-Purdue University Fort Wayne. – 2013.

[Kozodaev, 2015] Козодаев, М.А. Оценка проектного персонала: не забыть бы, для чего это делается (часть 1) // Управление проектами и программами. – 2015.

[Sahedani, 2013] Sahedani K.S. A Review: Mining Educational Data to Forecast Failure of Engineering Students / Komal S. Sahedani, B. Supriya Reddy // International Journal of Advanced Research in Computer Science and Software Engineering. – 2013. – С. 628-635.

[Bazhenov, 2014] Баженов Р.И. Об организации деловых игр в курсе «Управление проектами информационных систем, Научный аспект, 2014. – т.1, №1, С. 101-102.

[Vikentyeva, 2013] Викентьева, О.Л. Концепция студии компетентностных деловых игр [Электронный ресур] / О.Л. Викентьева, А.И. Дерябин, Л.В. Шестакова // Современные проблемы науки и образования. – 2013. – № 2; [URL: http://www.science-education.ru/108-8746] (дата обращения: 03.04.2013).

[Girev, 2010] Гирев, П.Е. Инновационные подходы к использованию интерактивных моделей в обучении / П.Е. Гирев, О.И. Мухин, О.А. Полякова // Дистанционное и виртуальное обучение, 2010. – С.84.

[Zarukina, 2010] Зарукина, Е.В. Активные методы обучения: рекомендации по разработке и применению: учебно-методическое пособие / М.И. Магура, Н.А. Логинова, М.М. Новик. – СПб.:СПбГИЭУ, 2010. – 59 с.

Authors' Information

Olga Vikentyeva *– National Research University Higher School of Economics, City of Perm, Perm, Russia, e-mail: oleovic@rambler.ru.*

Major Fields of Scientific Research: General theoretical information research, Multi-dimensional information systems

Alexandr Deryabin *– National Research University Higher School of Economics, City of Perm, Perm, Russia, e-mail: paid2@yandex.ru.*

Major Fields of Scientific Research: General theoretical information research, Multi-dimensional information systems

Nadezhda Krasilich *– National Research University Higher School of Economics, City of Perm, Perm, Russia, e-mail: mefaze@yandex.ru.*

Major Fields of Scientific Research: General theoretical information research, Multi-dimensional information systems

Lidiia Shestakova *– National Research University Higher School of Economics, City of Perm, Perm, Russia, e-mail: L.V.Shestakova@gmail.com.*

Major Fields of Scientific Research: General theoretical information research, Multi-dimensional information systems

TABLE OF CONTENTS

Evolutionary Synthesis of QCA Circuits: A Critique of Evolutionary Search Methods Based on the Hamming Oracle

R. Salas Machado, J. Castellanos, R. Lahoz-Beltra 203

Search, Processing, and Application of Logical Regularities of Classes

Yury Zhuravlev, Vladimir Ryazanov 216

Towards a Dawkins' Genetic Algorithm: Transforming an Interactive Evolutionary Algorithm into a Genetic Algorithm

S.Guil López, P.Cuesta Alvaro, S.Cano Alsua, R.Salas Machado, J.Castellanos, R.Lahoz-Beltra . 234

On the behavior of a class of infinite stochastic automaton in a random environment

Tariel Khvedelidze 250

Test generation for Estimating Power Consumption of sequantial Circuits

Liudmila Cheremisinova 261

Low-Power synthesis of Combinational CMOS Circuits

Dmitry Cheremisinov, Liudmila Cheremisinova 272

Employment of Business Intelligence Methods for Competences Evaluation in Business Games

Olga Vikentyeva, Alexandr Deryabin, Nadezhda Krasilich, Lidiia Shestakova..... 286

Table of contents 300