“Dunarea de Jos” University of Galati-Romania

Faculty of Electrical & Electronics Engineering

Dep. of Electronics and Telecommunications

Assoc. Prof. Rustem Popa, PhD

http://www.etc.ugal.ro/rpopa/index.htm

Contents

- ☞ About PhD thesis
- ☞ Teaching experience
- ☞ Research interests
- ☞ About Genetic Algorithms
- ☞ Some results

About PhD thesis (1999)

The development of new algorithms for testing and automatic reconfiguration of digital systems

I have studied some evolutionary algorithms for synthesis, testing and reconfiguration of programmable digital structures in case of single fault, for the purpose of self repair and adaptation to the changes of the environment. I have developed standard genetic algorithms and algorithms obtained by hybridizing genetic algorithms with other search methods, such as search by induction or simulated annealing.

Teaching experience

Undergraduate:

- Digital Electronics - since 1992
- Medical Electronics - since 1994
- Evolutionary Systems - (2000 – 2003)

Graduate:

- Audio, Speech and Image Processing - (2006 – 2009)
- Signal Processing by Soft Computing - (2006 – 2009)
- Audio and Speech Processing - since 2009
- Evolutionary Techniques in Signal Processing - since 2009

Research interests

- Evolutionary Computation
- Evolvable Hardware (EHW)
- Digital Electronics, including FPGA technology
- Soft Computing
- Digital Signal Processing
- Medical Electronics
- Artificial Consciousness

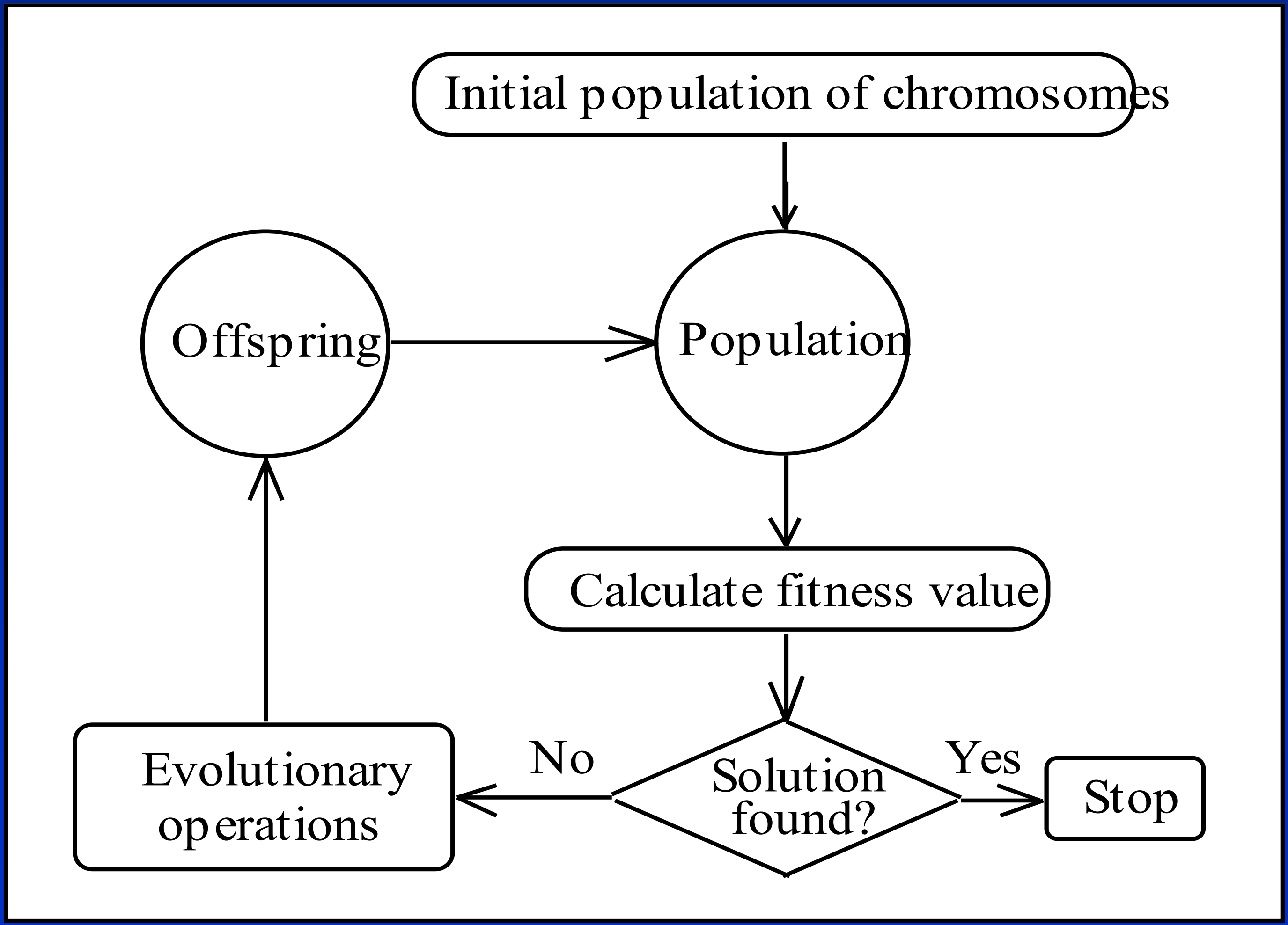
About Genetic Algorithms

Evolutionary computation is a subfield of artificial intelligence (AI) (or more particularly computational intelligence (CI)), involving combinatorial optimization problems. It may be recognised by the following criteria:

- iterative progress, growth or development
- population based
- guided random search
- parallel processing
- often biologically inspired

About Genetic Algorithms

A typical structure of an Evolutionary Algorithm (EA) :



About Genetic Algorithms

- GA begins with a **set of <u>solutions</u>** (represented by **<u>chromosomes</u>**) called **<u>population</u>**. Solutions from one population are selected according to their fitness, and form new solutions (**<u>offspring</u>**) by using genetic operators (**<u>crossover</u>**, **<u>mutation</u>**).

  This is motivated by a hope, that the new population will be better than the old one.
- This is repeated until some condition (for example number of generations or improvement of the best solution) is satisfied.

About Genetic Algorithms

The structure of a Genetic Algorithm:

```
begin
    generate randomly the initial population of chromosomes;
    repeat
        calculate the fitness of chromosomes in population;
        repeat
                select 2 chromosomes as parents;
                apply crossover to the selected parents;
                calculate the fitness of new child chromosomes;
        until end of the number of new chromosomes
        apply mutation to the new chromosomes;
        update the population;
    until end of the number of generations
end
```

About Genetic Algorithms

Encoding of a chromosome in a GA:

Chromosome 1	1101100100110110
Chromosome 2	1101111000011110

- A chromosome should in some way contain information about solution that it represents. The most used way of encoding is a binary string.
- The encoding depends mainly on the solved problem. For example, one can encode directly integer or real numbers, sometimes it is useful to encode some permutations and so on.

About Genetic Algorithms

Crossover:

Chromosome 1	11011 | 00100110110
Chromosome 2	11011 | 11000011110
Offspring 1	11011 | 11000011110
Offspring 2	11011 | 00100110110

This example shows a crossover in a single point. There are other ways how to make crossover, for example we can choose more crossover points. Crossover can be quite complicated and depends mainly on the encoding of chromosomes.

About Genetic Algorithms

Mutation:

Original offspring 1	1101111000011110
Original offspring 2	1101100100110110
Mutated offspring 1	1100111000011110
Mutated offspring 2	1101101100110100

Mutation operation randomly changes the offspring resulted from crossover. In case of binary encoding we can switch a few randomly chosen bits from 1 to 0 or from 0 to 1. The probability of mutation is low and it prevents the falling of all solutions in the population into a local optimum.

About Genetic Algorithms

Parameters of GA:

- Population size
- Encoding (binary, permutation, value, tree)
- Selection (roulette wheel, elitism, tournament)
- Crossover type
- Crossover probability
- Mutation type
- Mutation probability

Some results

Evolvable Hardware (EHW):

☞ Popa Rustem - *Genetic Algorithms: An Overview with Applications in Evolvable Hardware* published in "Bio-Inspired Computational Algorithms and Their Applications", Dr. Shangce Gao (Ed.), ISBN: 978-953-51-0214-4, InTech - Croatia, march 2012, pp. 105-120

☞ R. Popa, V. Nicolau, S. Epure - *A New Quantum Inspired Genetic Algorithm for Evolvable Hardware*, The 3-nd International Symposium on Electrical and Electronics Engineering, ISEEE 2010, Galati, Romania, 16-18 September, pp. 64-69 (IEEE Catalog Number CFP1093K-PRT, ISBN 978-1-4244-8407-2)

☞ R. Popa, S. Epure, V. Nicolau - *FPGA Circuits for Evolvable Hardware*, The 2-nd International Symposium on Electrical and Electronics Engineering, ISEEE 2008, Galati, Romania, 12-13 September, pp. 70-73 (ISSN 1844-8054)

☞R. Popa, V. Nicolau, S. Epure - *Evolvable Hardware in Xilinx PLDs*, First International Symposium on Electrical and Electronics Engineering, ISEEE 2006, Galati, Romania, 13-14 October, pp. 108-113 (ISBN (10) 973-627-325-3, ISBN (13) 978-973-627-325-4)

Some results (EHW)

Intrinsic evolution of a modulo 5 counter on a motherboard with CMOS integrated circuits (2004)

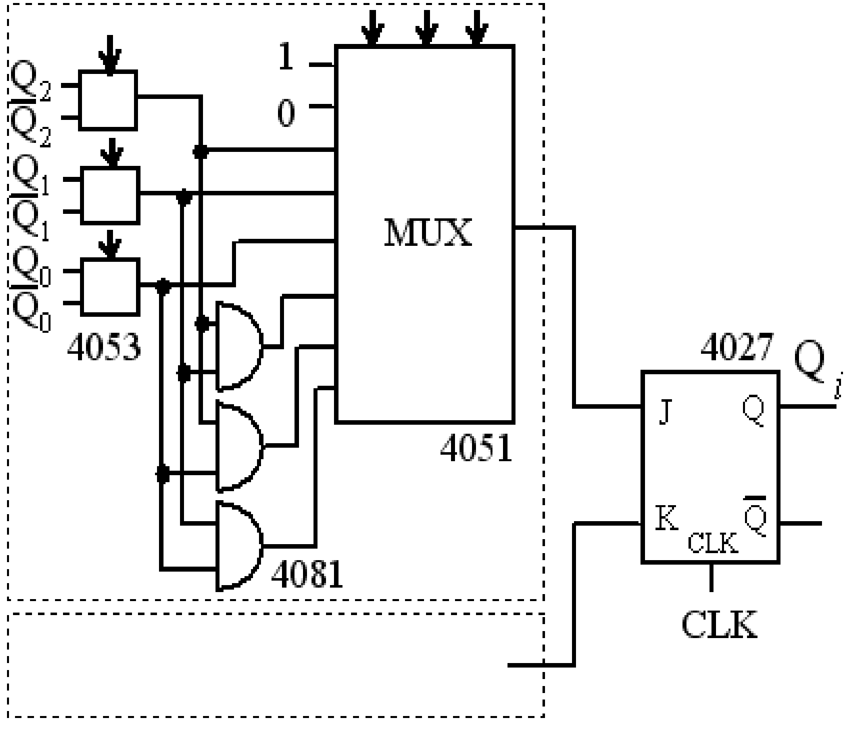


Fig. 9. The building-block for intrinsic digital EHW

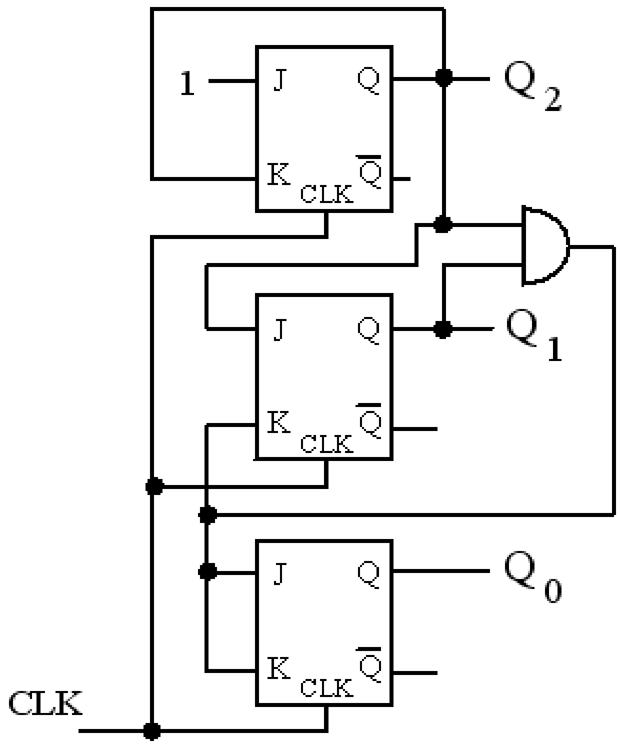


Fig. 10. An example of an evolved counter with 5 states

Some results (EHW)

Two examples of extrinsic evolution of a state machine with 8 states (2006)

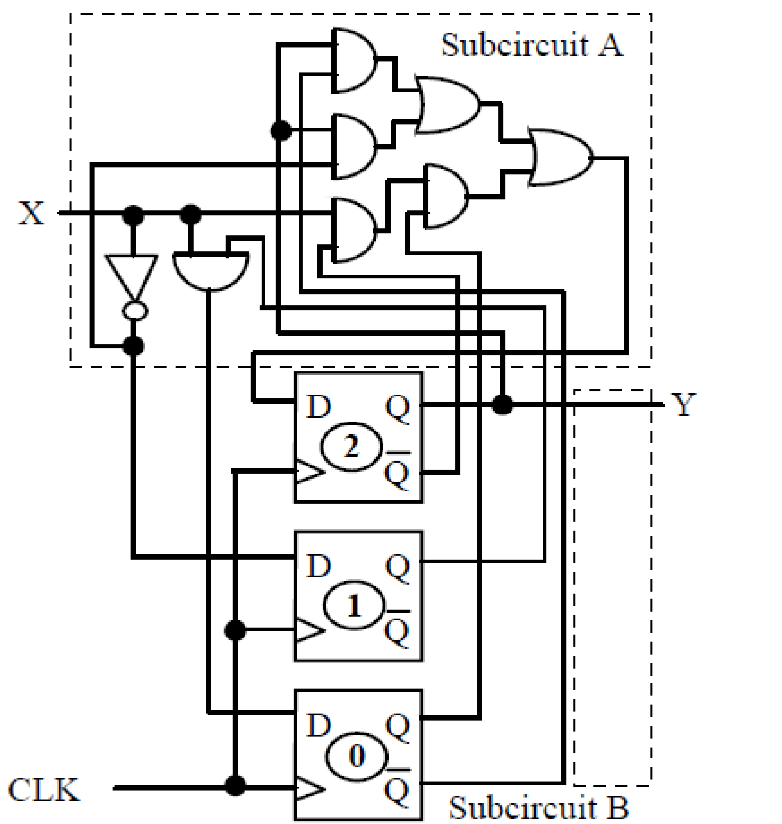


Fig. 5. Evolved optimal circuit solution of the sequence detector (first solution).

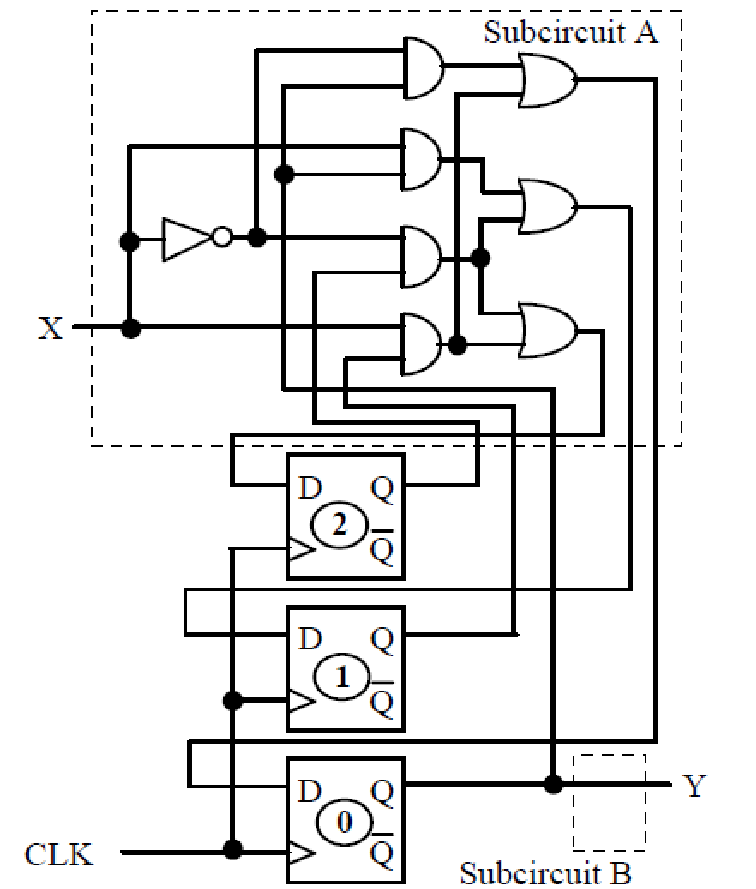
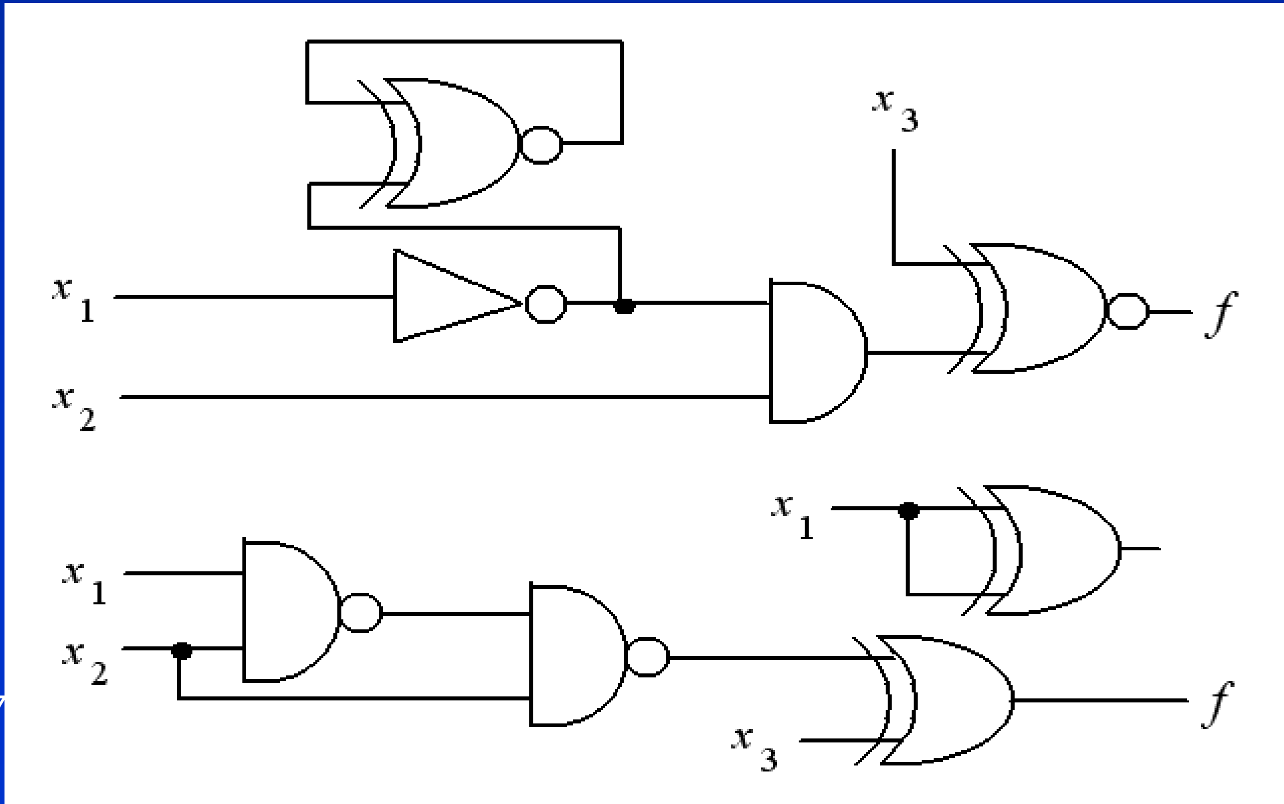


Fig. 6. Evolved optimal circuit solution of the sequence detector (second solution).

Some results (EHW)

Two examples of extrinsic evolution of a boolean function with 3 variables in a network with 4 gates, using 3 GAs: (2010)

- CGA (Canonical Genetic Algorithm)
- SCQGA (Single Chromosome Quantum Genetic Algorithm)
- QIGA (Quantum Inspired Genetic Algorithm)



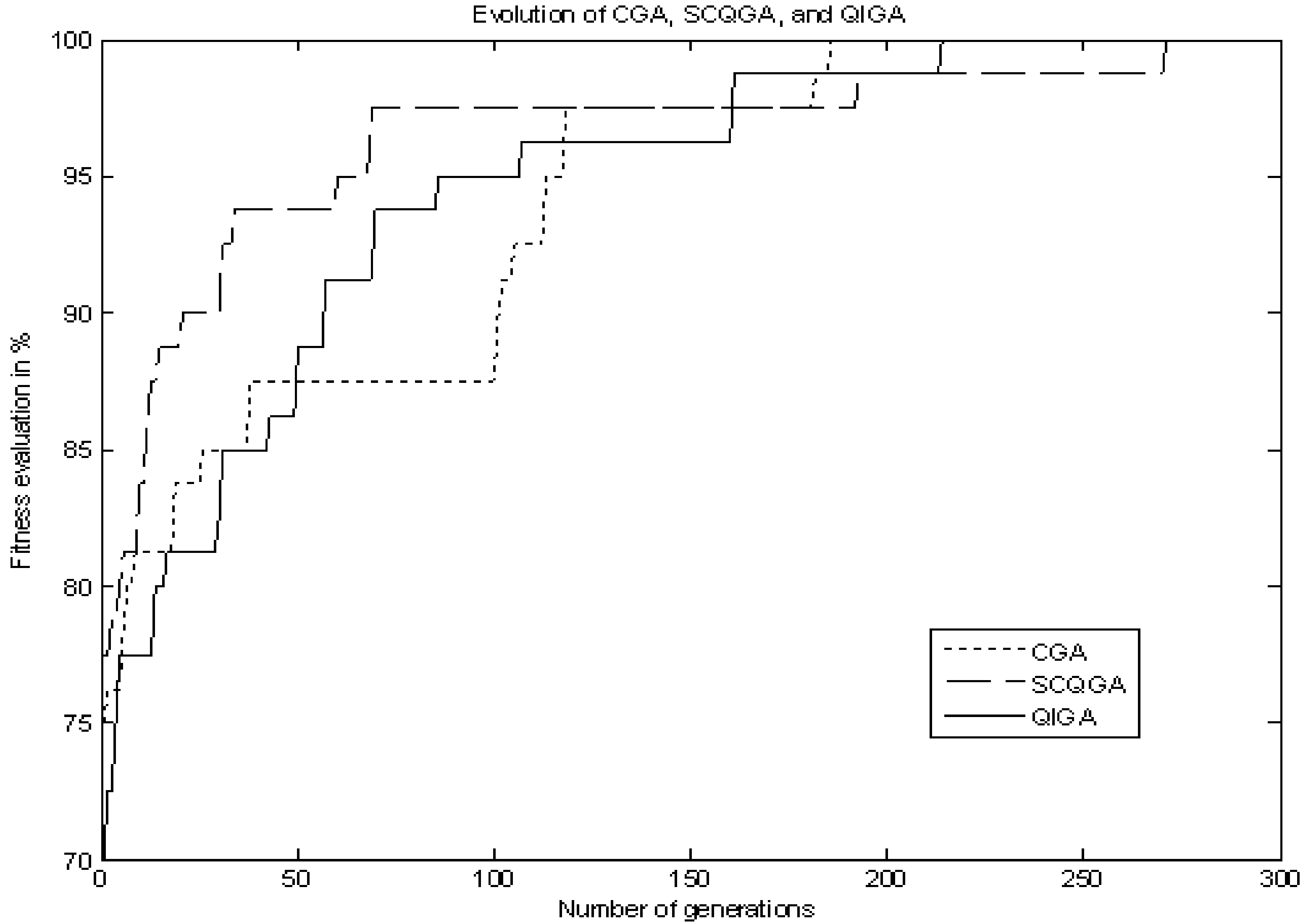


Fig. 4. The evolutions of CGA, SCQGA and QIGA in attempt to synthesize the function f given in (12). It was represented the fitness evaluation of the best chromosome in population after 10 successful runs of each algorithm on 300 generations.

Some results (EHW)

TABLE II

A COMPARISON BETWEEN CGA, SCQGA AND QIGA

Parameter	CGA	SCQGA	QIGA
Global time	73.990 s	38.599 s	19.263 s
Self time	2.447 s	1.417 s	1.390 s
Evaluation time	59.561 s	31.536 s	11.750 s
Calls of eval. func.	25200	19200	4836
Ratio between eval. and global time	80.5 %	81.7 %	60.9 %
Nr. of generations	300	300	300
Successful runs in 10 attempts (with fitness of 100%)	7	6	6

Some results

The Hybridisation of GAs:

☞R. Popa - *The Hybridization of the Selfish Gene Algorithm*, 2002 IEEE International Conference on Artificial Intelligence Systems, ICAIS 2002, Divnomorskoe, Russia, 5-10 September, 2002, pp. 345-350 (ISBN 0-7695-1733-1)

☞R. Popa, D. Aiordachioaie, V. Nicolau - *Multiple Hybridization in Genetic Algorithms*, The 16-th European Meeting on Cybernetics and Systems Research EMCSR'2002, Vienna, Austria, April 2-5, 2002, pp.536-541 (ISBN 3 85206 160 1)

Some results

The Hybridisation of the Selfish Gene Algorithm:

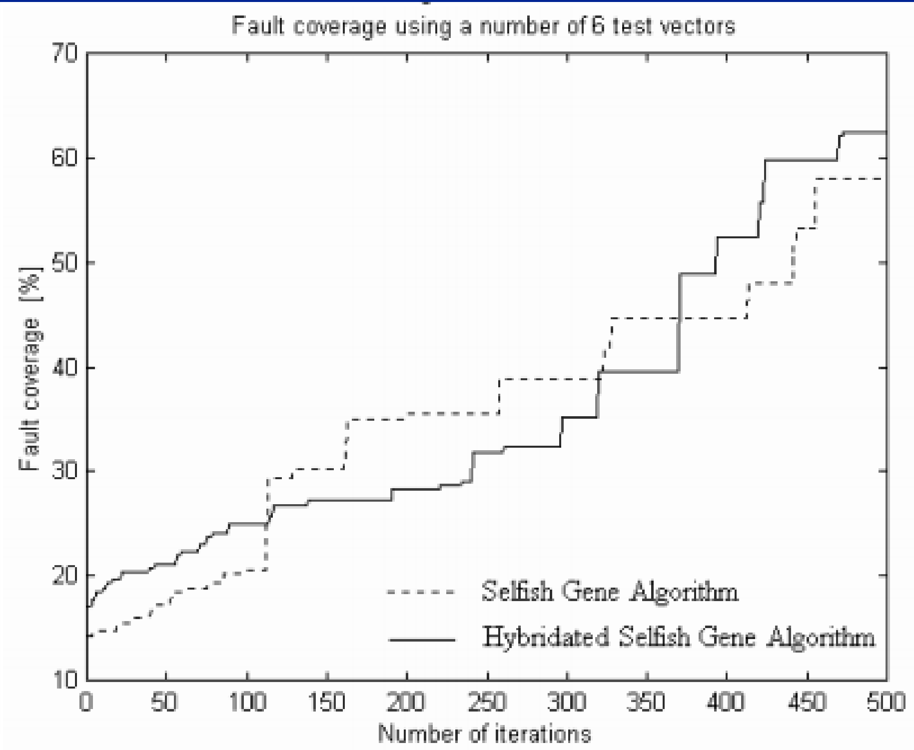


Figure 5. Runs of the two algorithms based on the Selfish Gene Theory on a fault coverage problem of 50 possible faults

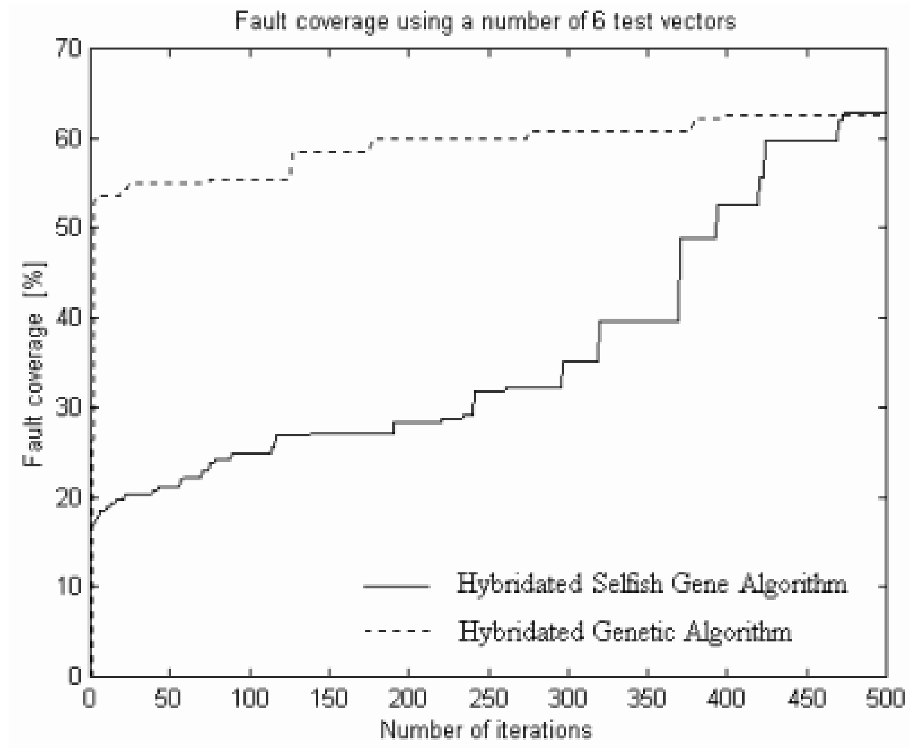


Figure 6. A comparison between two hybridated algorithms on a fault coverage problem of 50 possible faults

Some results

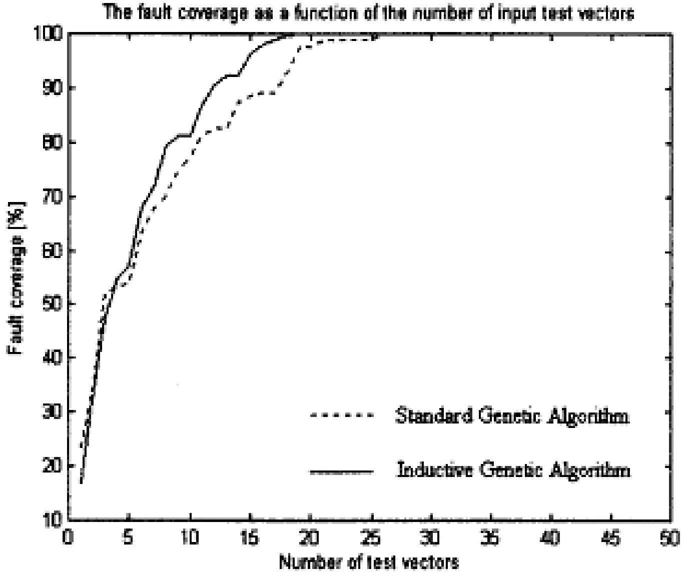


Figure 5. The fault coverage as a function of the number of input test vectors

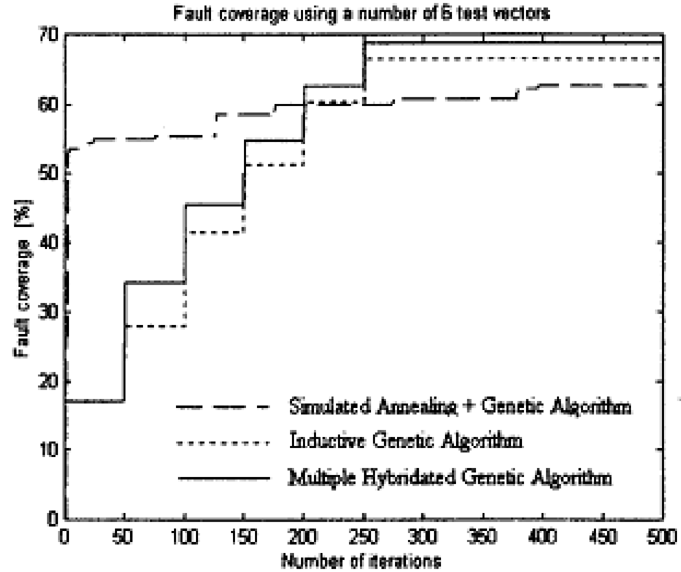


Figure 7. Runs of the three hybridated algorithms on a fault coverage problem of 50 possible faults

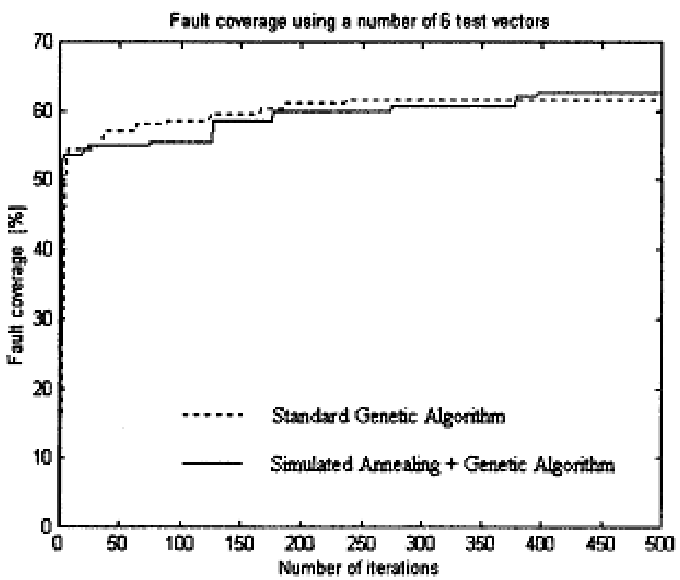


Figure 6. Runs of the two genetic algorithms on a fault coverage problem of 50 possible faults

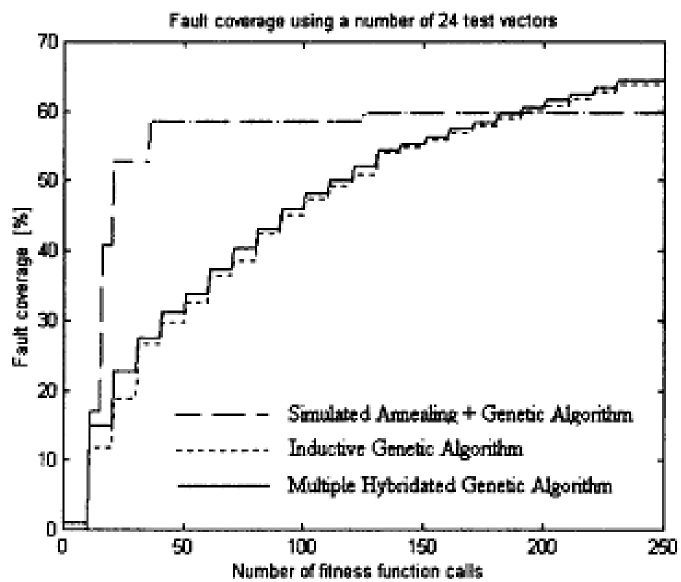


Figure 8. Runs of the three hybridated algorithms on a fault coverage problem of 200 possible faults

Multiple Hybridi-sation in GAs:

Some results

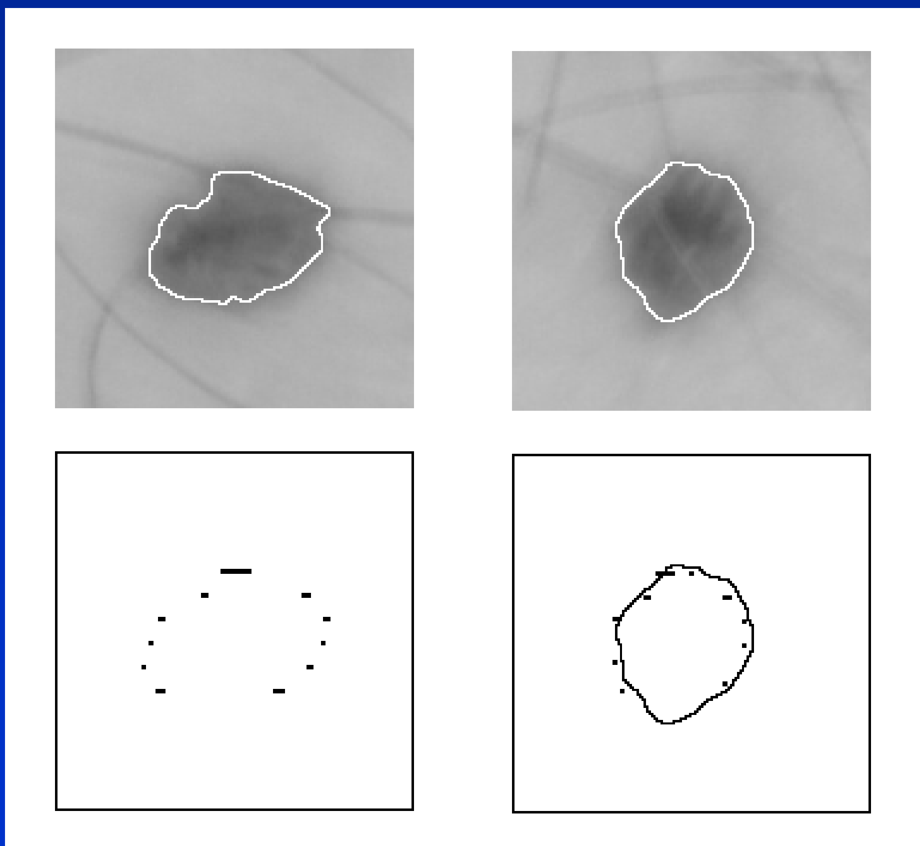
Image Processing with GAs:

☞Popa Rustem - *Melanocytic Lesions Screening through Particle Swarm Optimization* published in "Handbook of Research on Novel Soft Computing Intelligent Algorithms: Theory and Practical Applications“ (2 vol.), Dr. Pandian M. Vasant, IGI Global, 2013, ISBN13: 9781466644502, ISBN10: 1466644508, DOI: 10.4018/978-1-4666-4450-2.ch012, pp. 355-384

☞R. Popa, D. Aiordachioaie - *Genetic Recognition of Changes in Melanocytic Lesions*, The 8th International Symposium on Automatic Control and Computer Science, SACCS 2004, Iasi, Romania, 22-23 October, 2004, CD-ROM Proceedings - abstract page 69 (ISBN 973-621-083-9)

Some results

Two Different Images of the Same Melanocytic Lesion, an Affine Transformation and Two Different Lesions:



$$\begin{bmatrix} x_1' & y_1' & 1 \\ x_2' & y_2' & 1 \\ \cdots & \cdots & \cdots \\ x_N' & y_N' & 1 \end{bmatrix} \cdot \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{bmatrix} = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \cdots & \cdots \\ x_N & y_N \end{bmatrix},$$