# Bio-Inspired Optimization Strategies: A Survey [*]

Nuria Gómez Blas
Escuela Universitaria de Informática
Universidad Politécnica de Madrid
28031 Madrid, Spain
e-mail: ngomez@eui.upm.es

Luis Fernando de Mingo
Escuela Universitaria de Informática
Universidad Politécnica de Madrid
28031 Madrid, Spain
e-mail: lfmingo@eui.upm.es

Juan Castellanos Peñuela
Facultad de Informática
Universidad Politécnica de Madrid
28660 Madrid, Spain
e-mail: jcastellanos@fi.upm.es

## ABSTRACT

This paper provides a brief survey about optimization strategies from a biological point of view. There exists a clear difference between cooperative and competitive strategies. The former ones are based on the swarm colonies, in which all individuals share its knowledge about the goal in order to pass such information to other individuals to get optimum solution. The latter ones are based on competitive learning, that is, individuals can die and new individuals are created combining information of alive one; or are based on molecular/celular behaviour passing information from one structure to another. Some results, concerning grammatical swarm, obtained using the GEVA simulator are shown.

## Keywords

Computer science, artificial intelligence, natural computing.

## 1. INTRODUCTION

Natural sciences, and especially biology, represented a rich source of modelling paradigms. Well-defined areas of artificial intelligence (genetic algorithms, neural networks), mathematics, and theoretical computer science (L systems, DNA computing) are massively influenced by the behaviour of various biological entities and phenomena. In the last decades or so, new emerging fields of so-called natural computing identify new (unconventional) computational paradigms in different forms. There are attempts to define and investigate new mathematical or theoretical models inspired by nature, as well as investigations into defining programming paradigms that implement computational approaches suggested by biochemical phenomena. Especially since Adleman's experiment, these investigations received a new perspective. One hopes that global system-level behaviour may be translated into interactions of a myriad of components with simple behaviour and limited computing and communication capabilities that are able to express and solve, via various optimizations, complex problems otherwise hard to approach.

A number of computational paradigms, inspired or gleaned from biochemical phenomena, are becoming of growing interest building a wealth of models, called generically Molecular Computing. New advances in, on the one hand, molecular and theoretical biology, and on the other hand, mathematical and computational sciences promise to make it possible in the near future to have accurate systemic models of complex biological phenomena.

[*]Partially supported by projects CCG08-UAM TIC-4425-2009 and TEC2007-68065-C03-02.

## 2. NATURAL COMPUTATION

Natural computation [19], also called natural computing, is the field of research that works with computational techniques inspired in part by nature and natural systems. The aim of such research is to develop new computational tools (in software, hardware or wet-ware) for solving complex, usually conventionally-hard problems. This often leads to the synthesis of natural patterns, behaviours and organisms, and may result in the design of novel computing systems that use natural media with which to compute. Natural computing can be divided into three main branches:

- Computing inspired by nature (also called biologically inspired computing): This makes use of nature as inspiration for the development of problem solving techniques. The main idea of this branch is to develop computational tools (algorithms) by taking inspiration from nature for the solution of complex problems;
- The simulation and emulation of nature by means of computing: This is basically a synthetic process aimed at creating patterns, forms, behaviours, and organisms that (do not necessarily) resemble life-as-we-know-it. Its products can be used to mimic various natural phenomena, thus increasing our understanding of nature and insights about computer models; and
- Computing with natural materials: This corresponds to the use of natural materials to perform computation, thus constituting a true novel computing paradigm that comes to substitute or supplement the current silicon-based computers.

### 2.1 P Systems

P-systems represent a class of distributed and parallel computing devices of a biological type that was introduced in [5, 4] which are included in the wider field of cellular computing. Several variants of this model have been investigated and the literature on the subject is now rapidly growing. The main results in this area show that P-systems are a very powerful and efficient computational model [7, 9, 8]. There are variants that might be classified according to different criteria. They may be regarded as language generators or acceptors, working with strings or multi-sets, developing synchronous or asynchronous computation. Two main classes of P-systems can be identified in the area of membrane computing [6]: cell-like P-systems and tissue-like P-systems. The former type is inspired by the internal organization of living cells with different compartments and membranes hierarchically arranged; formally this structure is associated with a tree. Tissue P-systems have been motivated by the structure and behaviour of multi-cellular organisms where they form a multitude of different tissues performing various functions [6]; the structure of the system is instead represented as a graph where nodes are associated with the cells which are allowed to communicate alongside the edges of the graph.

## 2.2 Networks of Evolutionary Processors

The origin of networks of evolutionary processors [10] is a basic architecture for parallel and distributed symbolic processing, related to the Connection Machine [3] as well as the Logic Flow paradigm [1] which consists of several processors, each of them being placed in a node of a virtual complete graph, which are able to handle data associated with the respective node. Each node processor acts on the local data in accordance with some predefined rules, and then local data becomes a mobile agent which can navigate in the network following a given protocol. Only such data can be communicated which can pass a filtering process. This filtering process may require to satisfy some conditions imposed by the sending processor, by the receiving processor or by both of them. All the nodes send simultaneously their data and the receiving nodes handle also simultaneously all the arriving messages, according to some strategies, see, e.g., [11, 12, 2, 3].

## 2.3 Genetic Algorithms

Genetic Algorithms (GAs) are adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection and genetic. The basic concept of GAs is designed to simulate processes in natural system necessary for evolution, specifically those that follow the principles first laid down by Charles Darwin of survival of the fittest. As such they represent an intelligent exploitation of a random search within a defined search space to solve a problem.

GAs were introduced as a computational analogy of adaptive systems. They are modelled loosely on the principles of the evolution via natural selection, employing a population of individuals that undergo selection in the presence of variation-inducing operators such as mutation and recombination (crossover). A fitness function is used to evaluate individuals, and reproductive success varies with fitness.

- Randomly generate an initial population $M(0)$.
- Compute and save the fitness $u(m)$ for each individual $m$ in the current population $M(t)$.
- Define selection probabilities $p(m)$ for each individual $m$ in $M(t)$ so that $p(m)$ is proportional to $u(m)$.
- Generate $M(t+1)$ by probabilistically selecting individuals from $M(t)$ to produce offspring via genetic operators.
- Repeat step 2 until satisfying solution is obtained.

The paradigm of GAs descibed above is usually the one applied to solving most of the problems presented to GAs. Though it might not find the best solution. more often than not, it would come up with a partially optimal solution.

## 3. SOCIAL INTELLIGENCE

This section shows some new computational paradigms that are based on the co-operation of individuals instead on the competition of individuals (typically modelled by genetic algorithms). Such social intelligence makes individuals to evolve towards the best solution using information from other individuals but none of them disappear. This is a new approach taken from the biology, in essence, social behaviour helps individuals to adapt to their environment, as it ensures that they obtain access to more information than that captured by their own senses.

Two popular variants of swarm models exist, those inspired by studies of social insects such as ant colonies, and those inspired by studies of the flocking behaviour of birds and fish.

## 3.1 Ant Colony Optimization

Ant colony optimization (ACO) is a class of optimization algorithms modelled on the actions of an ant colony. ACO methods are useful in problems that need to find paths to goals. Artificial 'ants' - simulation agents - locate optimal solutions by moving through a parameter space representing all possible solutions. Real ants lay down pheromones directing each other to resources while exploring their environment. The simulated 'ants' similarly record their positions and the quality of their solutions, so that in later simulation iterations more ants locate better solutions.[2] One variation on this approach is the bees algorithm, which is more analogous to the foraging patterns of the honey bee.

## 3.2 Particle Swarm Optimization

Particle swarm optimization (PSO) is a global optimization algorithm for dealing with problems in which a best solution can be represented as a point or surface in an n-dimensional space. Hypotheses are plotted in this space and seeded with an initial velocity, as well as a communication channel between the particles [17, 18]. Particles then move through the solution space, and are evaluated according to some fitness criterion after each timestep. Over time, particles are accelerated towards those particles within their communication grouping which have better fitness values. The main advantage of such an approach over other global minimization strategies such as simulated annealing is that the large number of members that make up the particle swarm make the technique impressively resilient to the problem of local minima.

## 3.3 Stochastic Diffusion Search

Stochastic Diffusion Search (SDS) is an agent-based probabilistic global search and optimization technique best suited to problems where the objective function can be decomposed into multiple independent partial-functions. Each agent maintains a hypothesis which is iteratively tested by evaluating a randomly selected partial objective function parameterised by the agent's current hypothesis. In the standard version of SDS such partial function evaluations are binary, resulting in each agent becoming active or inactive. Information on hypotheses is diffused across the population via inter-agent communication. Unlike the stigmergic communication used in ACO, in SDS agents communicate hypotheses via a one-to-one communication strategy analogous to the tandem running procedure observed in some species of ant. A positive feedback mechanism ensures that, over time, a population of agents stabilise around the global-best solution. SDS is both an efficient and robust search and optimization algorithm, which has been extensively mathematically described.

## 3.4 Grammatical Swarm

Grammatical Swarm (GS) adopts a Particle Swarm learning algorithm coupled to a Grammatical Evolution (GE) genotype-phenotype mapping to generate programs in an arbitrary language. Grammatical Evolution (GE) is an evolutionary algorithm that can evolve computer programs in any language [13, 14], and can be considered a form of grammar-based genetic programming. Rather than representing the programs as parse trees, as in GP [15, 16], a linear genome representation is used. A genotype-phenotype mapping is employed such that each individuals variable length binary string, contains in its codons (groups of 8 bits) the information to select production rules from a Backus Naur Form (BNF) grammar. The grammar allows the generation of programs in an arbitrary language that are guaranteed to be syntactically correct, and as such it is used as a generative grammar, as opposed to the classical use of grammars in compilers to check syntactic correctness of sentences. The user can tailor the grammar to produce solutions that are purely syntactically constrained, or they may incorporate domain knowledge by biasing the grammar to produce very specific forms of sentences. BNF is a notation that represents a language in the form of production rules.

Let us suppose the following BNF grammar:

```
<expr> :: = <expr><op><expr>
          | <var>
<op> :: = +
         | -
         | *
<var> :: = x
          | y
```

And the following genotype:

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| 14 | 8 | 27 | 254 | 5 | 17 | 12 |

In the example individual (see figure 1), the left-most `<expr>` in `<expr> <op> <expr>` is mapped by reading the next codon integer value 240 and used in 240%2 = 0 to become another `<expr> <op> <expr>`. The developing program now looks like `<expr> <op> <expr> <op> <expr>`. Continuing to read subsequent codons and always mapping the left-most non-terminal the individual finally generates the expression `y * x - x - x + x`, leaving a number of unused codons at the end of the individual, which are deemed to be introns and simply ignored.

This is the classic benchmark problem in which evolution attempts to find the five input even-parity boolean function [20]. The grammar adopted here is:

```
<prog> ::= <expr>
<expr> ::= <expr> <op> <expr>
         | ( <expr> <op> <expr> )
         | <var> | <pre-op> ( <var> )
<pre-op> ::= not
<op> ::= "|" | & | ^
<var> ::= d0 | d1 | d2 | d3 | d4
```

The result is given by the best individual, see transcript bellow. Figure 2 shows a graphic with the best, average and variance of the swarm population. This figure has been obtained using the *GEVA* simulator [20].

```
( not ( d1 ) | d2 ^ d4 ) &
not ( d3 ) ^ ( not ( d1 ) &
( not ( d2 ) &
( not ( d2 ) |
( d1 ^ not ( d3 ) ^ not ( d1 ) ^
( not ( d1 ) ^ ( d0 | not ( d4 ) ) ) ) ) ) ^ d4 )
 ^ not ( d0 ) ) ^ d1
```

## 4. CONCLUSIONS

This paper has reviewed some natural computation strategies as a survey concerning optimization strategies. Some competitive and collaborative models has been exposed in order to understand the ability to extract some biological concepts and apply them in computational models as described along the paper. Such bio-inspired models have proof to be a powerful tool in order to solve non common problems in a collaborative/competitive way.

## 5. ACKNOWLEDGEMENT

This work has been done by the Natural Computing Group at the Universidad Politécnica de Madrid.

## REFERENCES

[1] L. Errico and C. Jesshope, "Towards a new architecture for symbolic processing", *Artificial Intelligence and Information-Control Systems of Robots*. 94, pp. 31-40, 1994.

[2] S. Fahlman, G. Hinton, and T. Seijnowski, "Massively parallel architectures for AI: NETL, THISTLE and Boltzmann machines", *Proc. AAAI National Conf. on AI*, pp. 109-113, 1983.

[3] W. Hillis, "The Connection Machine", MIT Press, Cambridge, 1985.

[4] G. Ciobanu, R. Desai, and A. Kumar, "Membrane systems and distributed computing", *Pre-Proceedings of Second Workshop on Membrane Computing*, Curtea de Arges, Romania, August 2002.

[5] G.Păun and G. Rozenberg,"A guide to membrane computing", *Theoretical Computer Science*, 287(1), pp. 73-100, 2002.

[6] C. Martín-Vide, G. Păun, J. Pazos, and A. Rodríguez-Patón, "Tissue P systems", *Theoretical Computer Science*, 296(2), pp. 295-326, 2003.

[7] A. Romero-Jiménez and M. J. Pérez-Jiménez, "Simulating Turing machines by P systems with external output", *Fundamenta Informaticae*, 49(1-3), pp. 273-278, 2002.

[8] A. Alhazov, R. Freund, and M. Oswald, "Cell/symbol complexity of tissue P systems with symport/antiport rules", *International Journal of Foundations of Computer Science*, 17(1), pp. 3-25, 2006.

[9] L.Cardelli and G. Paun, "An universality result for a (mem)brane calculus based on mate/drip operations", *International Journal of Foundations of Computer Science*, 17(1), pp. 49-68, 2006.

[10] J. Castellanos, C. Martín-Vide, V. Mitrana, and J. Sempere, "Networks of evolutionary processors", *Acta Informática*, 39, pp. 517-529, 2003.

[11] C. Martin-Vide, V. Mitrana, M. Perez-Jimenez, and F. S. Caparrini, "Hybrid networks of evolutionary processors", *Lecture Notes in Computer Science*, 2723, pp. 401-412, 2003.

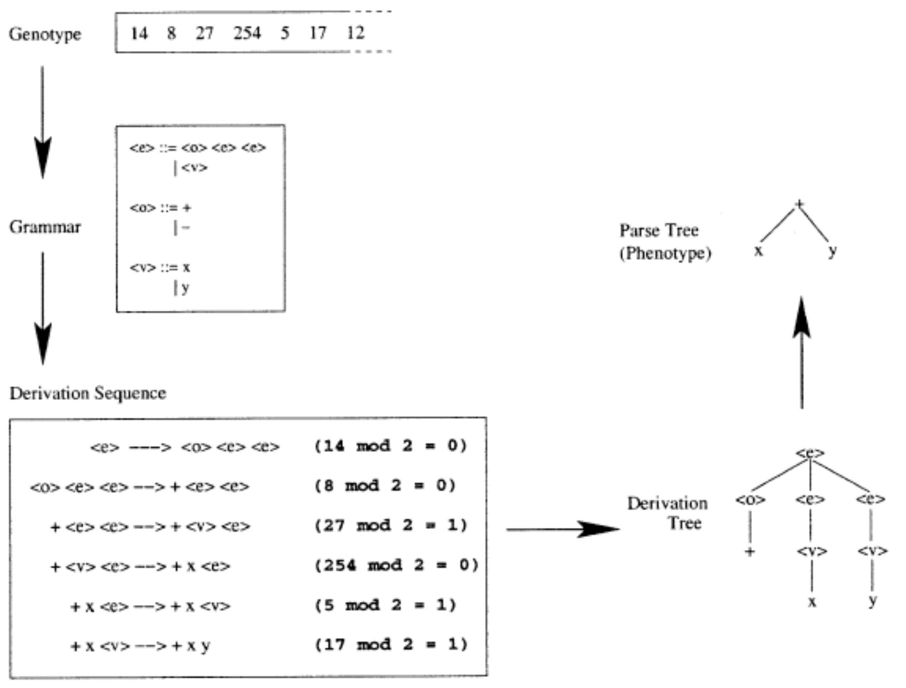
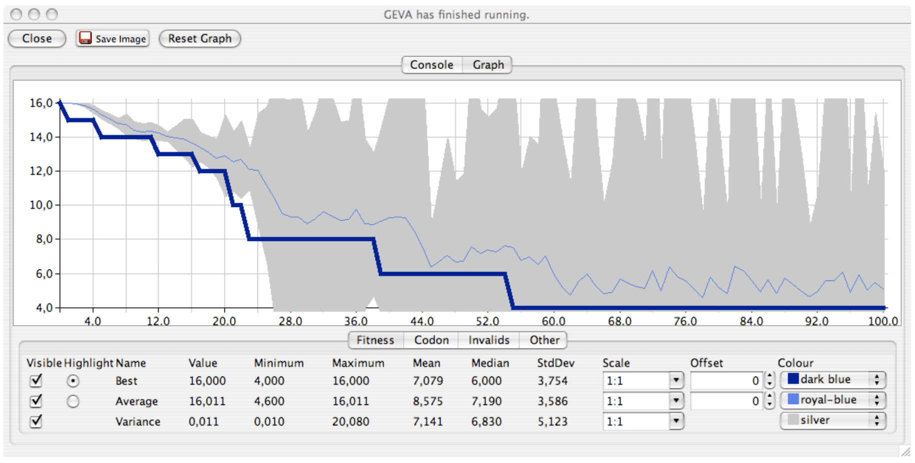


Figure 1. Grammatical Swarm concepts.



| Visible | Highlight | Name | Value | Minimum | Maximum | Mean | Median | StdDev | Scale | Offset | Colour |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ✓ | ● | Best | 16,000 | 4,000 | 16,000 | 7,079 | 6,000 | 3,754 | 1:1 | 0 | dark blue |
| ✓ | ○ | Average | 16,011 | 4,600 | 16,011 | 8,575 | 7,190 | 3,586 | 1:1 | 0 | royal-blue |
| ✓ | | Variance | 0,011 | 0,010 | 20,080 | 7,141 | 6,830 | 5,123 | 1:1 | | silver |

Figure 2. Results of *even-5-parity* problem simulated with GEVA.

[12] E. C. Varju and V. Mitrana, "Evolutionary systems, a language generating device inpired by evolving communities of cells", *Acta Informatica*, 36, pp. 913-926, 2000.

[13] M. ONeill and C. Ryan, "Grammatical Evolution", *IEEE Trans. Evolutionary Computation*, Vol. 5, No.4, 2001.

[14] M. ONeill, C. Ryan, M. Keijzer and M. Cattolico, "Crossover in Grammatical Evolution", *Genetic Programming and Evolvable Machines*, Vol. 4 No. 1. Kluwer Academic Publishers, 2003.

[15] J.R. Koza, D. Andre, F.H. Bennett III and M. Keane "Genetic Programming 3: Darwinian Invention and Problem Solving", Morgan Kaufmann, 1999.

[16] J.R. Koza, M. Keane, M.J. Streeter, W. Mydlowec, J. Yu, and G. Lanza, "Genetic Programming IV: Routine Human-Competitive Machine Intelligence". Kluwer Academic Publishers, 2003.

[17] J. Kennedy, R. Eberhart and Y. Shi, "Swarm Intelligence", Morgan Kauff-man, San Mateo, California, 2001.

[18] E. Bonabeau, M. Dorigo and G. Theraulaz, "Swarm Intelligence: From Natural to Artificial Systems", Oxford University Press, Oxford, 1999.

[19] http://en.wikipedia.org

[20] http://www.grammatical-evolution.org/