A New Optimization Algorithm based on a Paradigm Inspired by Nature

Leslie Astudillo, Patricia Melin, Oscar Castillo

Abstract:

In this paper, we propose a new optimization algorithm for soft computing problems, which is inspired on a nature paradigm: the reaction methods existing on chemistry, and the way the elements combine with each other to form compounds, in other words, quantum chemistry. This paper is the first approach for the proposed method, and it presents the background, main ideas, desired goals and preliminary results in optimization.

Keywords: *natural computing, novel optimization method, chemical reactions paradigm.*

1. Introduction

Several works have proved the relevance of computing techniques to solve diverse kinds of problems, including forecasting, control and pattern recognition among others [1], [2], [3].

These techniques not only comply with their objective, but they also pro mote the creation of new ways to give solutions and improve the actual methods as well [4],[5], [6].

One of the main difficulties when designing the structure of a solution method is the tuning of the parameters; which are the key to the success of these applications. These parameters will vary depending on the complexity of the problem and the method used to find the solution; and in some cases, they stem from our own ability to conceptualize the problem itself, taking in account, the inputs of the system and the expected output values.

Due to these facts, several optimization strategies based on nature paradigms have arisen. From Ant Colony Optimization, to Particle Swarm Optimization among others, these strategies had emerged as an alternative way to solve problems [7], [8], [9], [10], [11], [12], [13].

For this work, we will be observing the process in which the different elements existing in nature are created, behave and interact with each other to form chemical compounds.

The structure of this paper is the following. Section 2 shows a brief description of the chemical method that inspired this investigation; section 3 describes the proposed method and first approach; section 4 shows the preliminary experiment results; in section 5 we describe the current and future work and section 6 shows some references.

2. Chemical Paradigm

In order to have a better understanding of the process that we intend to model, we present some general definitions [14],[15].

Chemistry is the study of matter and energy and the interaction between them, including the composition, the properties, the structure, the changes which it undergoes, and the laws governing those changes. A substance is a form of matter that has a defined composition and characteristic properties. There are two kinds of substances: elements and compounds.

An element is a substance that cannot be broken down into simpler substances by ordinary means. It is apparent from the wide variety of different materials in the world that there are a great many ways to combine elements.

Compounds are substances formed by two or more elements combined in definite proportions through a chemical reaction. There are millions of known compounds, and thousands of new ones are discovered or synthesized each year.

A chemical reaction is a change in which at least one substance changes its composition and its sets of properties; they are classified into 4 types.

Type 1: combination reactions: (B+C → BC).

A combination reaction is a reaction of two reactants to produce one product. The simplest combination reactions are the reactions of two elements to form a compound. After all, if two elements are treated with each other, they can either react or not.

Type 2: decomposition reactions: (BC → B+C).

The second type of simple reaction is decomposition. This reaction is also easy to recognize. Typically, only one reactant is given. A type of energy, such as heat or electricity, may also be indicated. The reactant usually decomposes to its elements, to an element and a simpler compound, or to two simpler compounds.

Binary compounds may yield two elements or an element and a simpler compound. Ternary (three-element) compounds may yield an element and a compound or two simpler compounds. These possibilities are shown in the Figure 1.

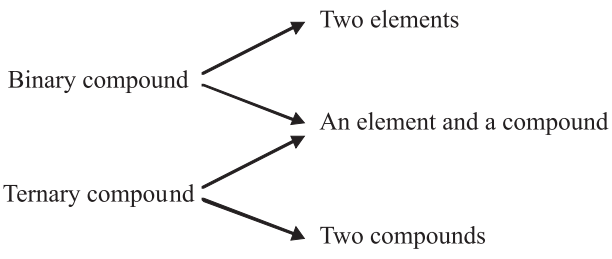

Fig. 1. Decomposition possibilities.

Type 3: substitution reactions: (C + AB → AC + B).

Elements have varying abilities to combine. Among the most reactive metals are the alkali metals and the alkaline earth metals. On the opposite end of the scale of reactivities, among the least active metals or the most stable metals are silver and gold, prized for their lack of reactivity. Reactive means the opposite of stable, but means the same as active.

When a free element reacts with a compound of different elements, the free element will replace one of the elements in the compound if the free element is more reactive than the element it replaces. In general, a free metal will replace the metal in the compound, or a free nonmetal will replace the nonmetal in the compound. A new compound and a new free element are produced.

Type 4: double-substitution reactions: (AB + CD → CB + AD).

Double-substitution or double-replacement reactions, also called double-decomposition reactions or metathesis reactions, involve two ionic compounds, most often in aqueous solution. In this type of reaction, the cations simply swap anions. The reaction proceeds if a solid or a covalent compound is formed from ions in solution. All gases at room temperature are covalent. Some reactions of ionic solids plus ions in solution also occur. Otherwise, no reaction takes place.

Just as with replacement reactions, double-replacement reactions may or may not proceed. They need a driving force. In replacement reactions the driving force is reactivity; here it is insolubility or co-valence.

3. Modeling the Chemical Paradigm

Now that we have described the natural paradigm that we intent to mimic, the next step is to define the general structure of our optimization algorithm; which, initially will be developed in 5 phases: a combination algorithm, a decomposition algorithm, a substitution algorithm, a double-substitution algorithm and the final algorithm, which will be the combination of all the previous four.

The steps to consider in this optimization method will be as follows:

1. First, we need to generate an initial pool of elements/compounds.
2. Once we have the initial pool, we have to evaluate it.
3. Based on the previous evaluation, we will select some elements/compounds to "induce" a reaction.
4. Given the result of the reaction, we will evaluate the obtained elements/compounds.
5. Repeat the steps until the algorithm meets the criteria (desired result or maximum number of iterations is reached).

In order to start testing the phases of the algorithm, we will be applying these to the following (but not restricted to) functions: De Jong's and Rosenberg's functions [16], [12].

4. Preliminary experimental results

Figure 2 shows the De Jong's first function also called the sphere model, which is continuous, convex, unimodal and is represented by the equation:

$$f_1(x) = \sum_{i=1}^{n} x_i^2 \tag{1}$$

The domain is given by:

$$-\infty \le x_i \le \infty \tag{2}$$

And has a global minimum represented by:

$$f(x) = 0;\ x(i) = 0;\ i = 0 : n. \tag{3}$$

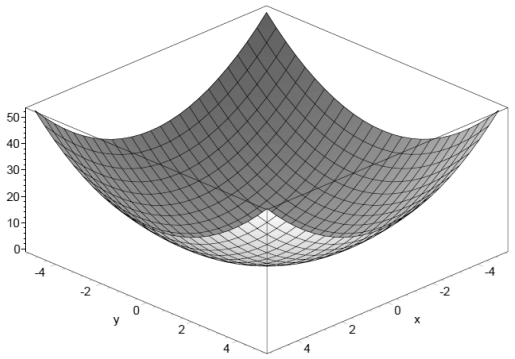

Fig. 2. De Jong's First function in 2D.

The fist approach to solve this equation is given by applying a minimization algorithm based on the decomposition reactions.

The main idea of this particular algorithm is, given a random set of initial numbers, decompose each one into smaller numbers in a way that can be represented by a binary tree, where each ensuing node will be decomposed as well into smaller numbers, to lead the result into the minimum of the function.

To start from the simplest option, in these early experiments all decomposed elements are considered to have the same value, and they are given by:

$$\text{Decomposed\_Element}(n) = \text{Element}/m \tag{4}$$

where n is the element index and m is the number of decomposed elements generated. Because the resulting values are the same for each decomposed element, only one will be selected to be evaluated in the function.

Let's consider an initial pool of 5 elements (randomly generated); each one will be decomposed in 3 sub elements throughout 10 iterations. Table 1 shows the final and average values of the best and worst result reached by the algorithm throughout 30 experiments.

Table 1. Worst and best results throughout 30 experiments evaluating the first De Jong's Function.

Experiment number	Minimum Value	Average value	Comments
1	3.799e-14	1.656e-06	Best result
13	7.829e-09	0.341	Worst result

Figure 3 shows the minimized values trough the 10 iterations of experiment number 1, which reached the minimum value overall the 30 experiments.

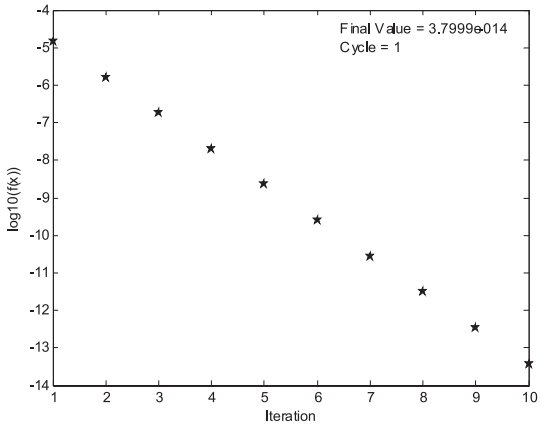

Fig. 3. Minimum value reached in experiment no. 1.

In Figure 4 we can see the behavior of the algorithm along the 30 experiments, where every experiment is represented by "Cycles" of 10 iterations each.

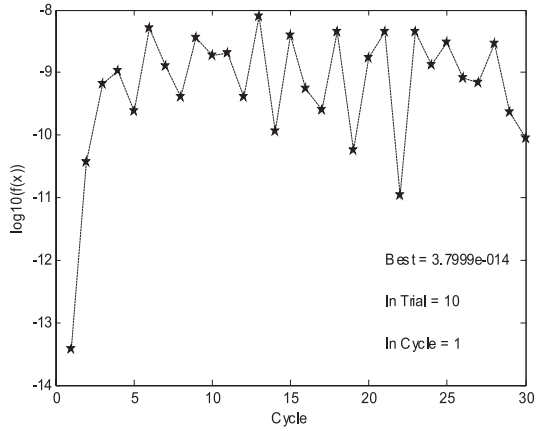

Fig. 4. Minimum values reached 30 experiments.

Table 2 shows the standard deviation calculated by iteration throughout 30 experiments.

Table 2. Standard deviation per trial in 30 experiments evaluating the first De Jong's Function

Trial	Standard Deviation
1	0.769404299
2	0.085489367
3	0.009498819
4	0.001055424
5	0.000117269
6	1.30E-05
7	1.45E-06
8	1.61E-07
9	1.79E-08
10	1.99E-09

Figure 5 shows the plot of the average and standard deviations calculated per iteration in 30 experiments.

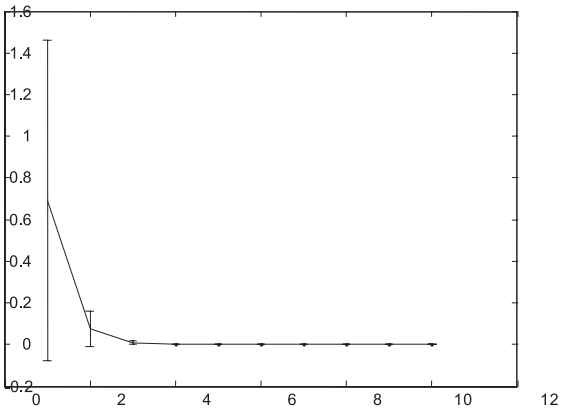

Fig. 5. Plot of the average and standard deviations per iteration in 30 experiments evaluating the De Jong's first function: the sphere model.

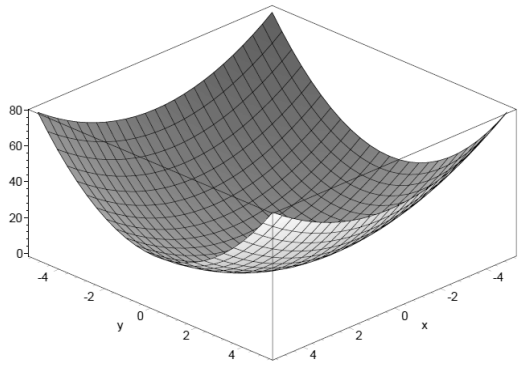

Fig. 6. The weighted sphere model in 2D.

Figure 6 shows the axis parallel hyper-ellipsoid, also known as the weighted sphere model. It a continuous, convex and unimodal function and is represented by the equation:

$$f_1(x) = \sum_{i=1}^{n} x * x_i^2 \tag{5}$$

The domain is given by:

$$-\infty \le x_i \le \infty \tag{6}$$

And has a global minimum represented by:

$$f(x) = 0;\ x(i) = 0;\ i = 0 : n. \tag{7}$$

The Table 3 shows the final and average values of the best and worst result reached by the algorithm throughout 30 experiments.

Table 3. Worst and best results throughout 30 experiments evaluating the first De Jong's Function.

Experiment number	Minimum Value	Average value	Comments
28	8.85E-17	7.00-05	Best result
23	7.91-13	0.62	Worst result

Figure 7 shows the minimized values trough the 10 iterations of experiment number 28, which reached the minimum value overall the 30 experiments.

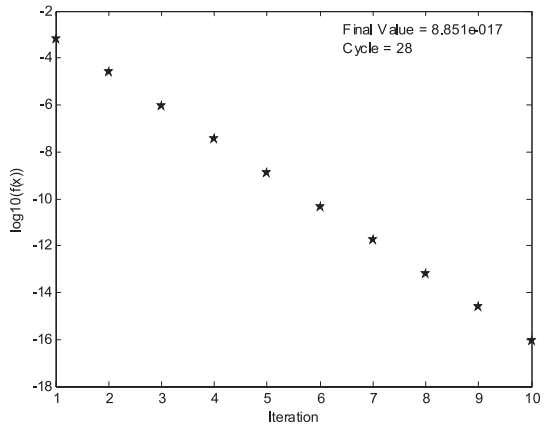

Fig. 7. Minimum value reached in experiment no. 28, evaluating the weighted sphere model.

In Figure 8 we can see the behavior of the algorithm along the 30 experiments, where every experiment is represented by "Cycles" of 10 iterations each.

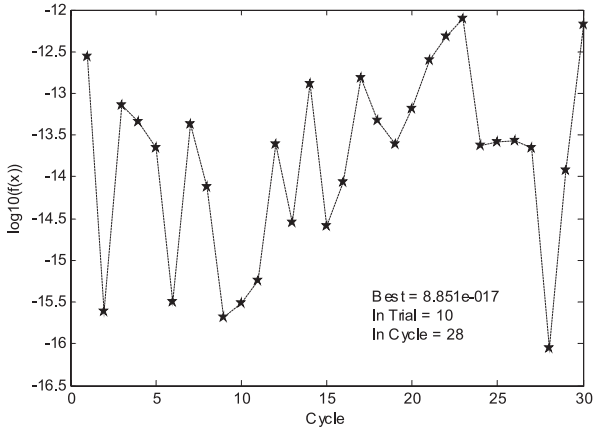

Fig. 8. Minimum values reached 30 experiments evaluating weighted sphere model.

Table 4 shows the standard deviation calculated by iteration throughout 30 experiments.

Table 4. Standard deviation per trial in 30 experiments evaluating the weighted sphere model.

Trial	Standard Deviation
1	1.525
2	0.056
3	0.0020
4	7.75e-05
5	2.87e-06
6	1.06e-07
7	3.93e-09
8	1.45e-10
9	5.40e-12
10	2.00e-13

Figure 9 shows the plot of the average and standard deviations calculated per iteration in 30 experiments, evaluating the weighted sphere model.

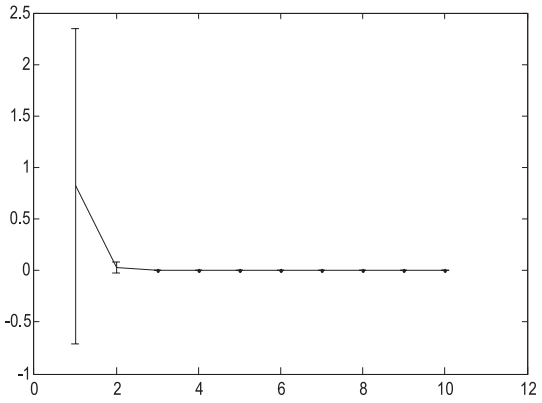

Fig. 9. Plot of the average and standard deviations per iteration in 30 experiments evaluating the weighted sphere model.

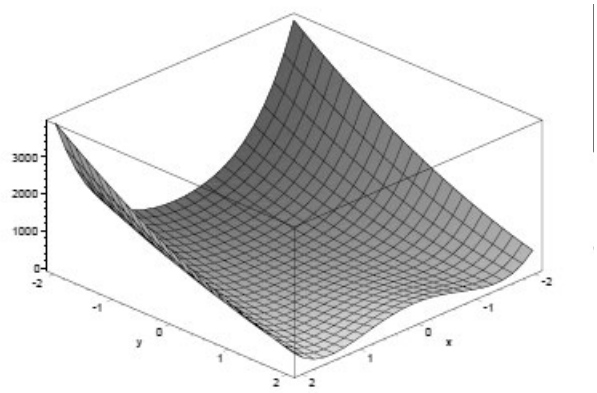

Fig. 10. The Rosembrock's Valley in 2D.

Figure 10 shows the Rosembrock's Valley function, also known as banana function or the second function of De Jong. The global optimum lies inside a long, narrow, parabolic shaped flat valley. It is represented by the equation:

$$f_1(x) = \sum_{i=1}^{n-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + \left(1 - x_i\right)^2 \right]. \quad (8)$$

The test area is usually restricted to hypercube:

$$-2.048 \le x_i \le 2.048 \quad (9)$$

And has a global minimum represented by:

$$f(x) = 0;\ x(i) = 1;\ i = 1:n. \quad (10)$$

The Table 5 shows the final and average values of the best and worst result reached by the algorithm throughout 30 experiments.

Table 5. Worst and best results throughout 30 experiments evaluating the Rosembrock's Valley Function.

Experiment number	Minimum Value	Average Value	Comments
24	0.99966	1.053	Best final result
30	0.99997	1.013	Worst final result
17 (Iteration 2)	6.88e-06	1.23	Best result (Minimum value)

Figure 11 shows the minimized values trough the 10 iterations of experiment number 24, which reached the minimum final value overall the 30 experiments..

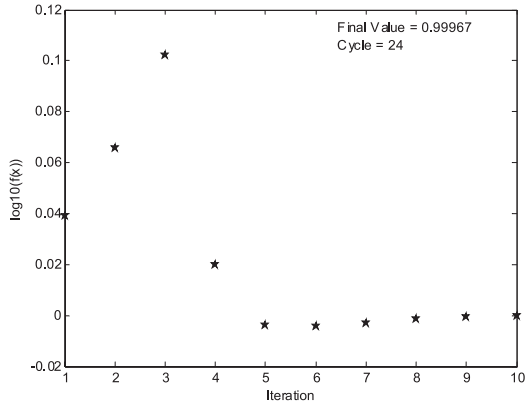

Fig. 11. Minimum value of experiment no. 24, evaluating the Rosembrock's Valley Function.

In Figure 12 we can see the behavior of the algorithm along the 30 experiments, where every experiment is represented by "Cycles" of 10 iterations each.

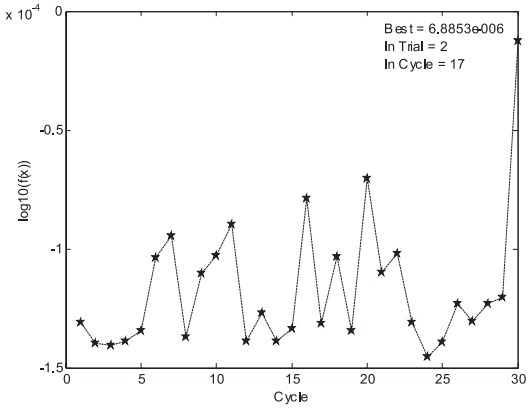

Fig. 12. Minimum values reached in 30 experiments evaluating Rosembrock's Valley Function.

As we can see in Table 5 and Figure 12, the last value reached in each experiment, may not be the "best value" found by the algorithm.

Figure 13 shows the behavior of trial with the minimum result through the 30 experiments. In this experiment, the final value was 0.9997, but the minimum value (6.88e-06) was found in iteration no. 2.

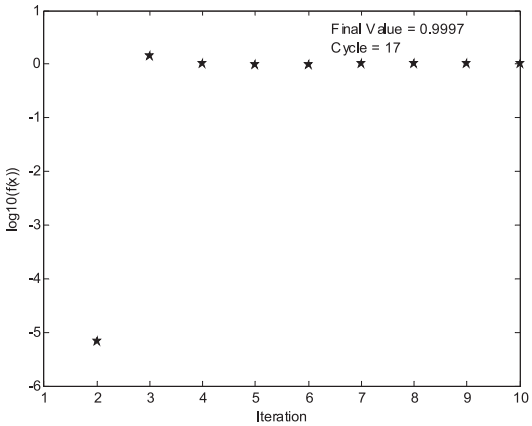

Fig. 13. Experiment 17, which had the minimum value in 30 experiments.

Table 6 shows the standard deviation calculated by iteration throughout 30 experiments.

Table 6. Standard deviation per trial in 30 experiments evaluating the Rosembrock's Valley Function.

Trial	Standard Deviation
1	13.443
2	0.796
3	0.205
4	0.024
5	0.004
6	0.0014
7	0.0012
8	0.00053
9	0.00018
10	6.44e-05

Figure 14 shows the plot of the average and standard deviations calculated per iteration in 30 experiments, evaluating the Rosembrock's Valley Function.

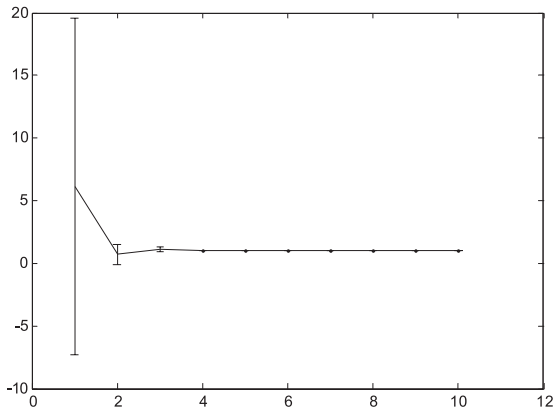

Fig. 14. Plot of the average and standard deviations per iteration in 30 experiments evaluating the Rosembrock's Valley Function.

5. Conclusions

In this paper, we introduced the first stage of a new optimization method that tries to mimic the chemical reactions. The *Decomposition Reaction Method* was applied in 3 benchmark functions to evaluate the first development phase of the optimization algorithm. This *Decomposition Reaction Method* by itself finds or guides the result to a certain minimum value, but, due the nature of some functions, it is necessary to introduce the second phase of this optimization method: The *Combination Reactions Method*, which will be able to guide the algorithm to find an optimum value when it is not necessarily the "smallest" one. At the time, more functions are being evaluated to pursue the tuning of the algorithm itself.

ACKNOWLEDGMENTS

The authors would like to thanks CONACYT and Tijuana Institute of Technology for the facilities and resources granted for the development of this research.

AUTHORS
Leslie Astudillo*, Patricia Melin, Oscar Castillo - Tijuana Institute of Technology, Tijuana México, E-mails: leslie.astudillo@suntroncorp.com, epmelin@hafsamx.org, ocastillo@tectijuana.mx.
* Corresponding author

References

[1] Hidalgo D., Melin P., Licea, G., "Optimization of Modular Neural Networks with Interval Type-2 Fuzzy Logic Integration Using an Evolutionary Method with Application to Multimodal Biometry", *Bio-inspired Hybrid Intelligent Systems for Image Analysis and Pattern Recognition*, 2009, pp. 111-121.

[2] Astudillo L., Castillo O., Aguilar L., "Hybrid Control for an Autonomous Wheeled Mobile Robot Under Perturbed Torques", *Foundations of Fuzzy Logic and Soft Computing, IFSA Proceedings*, 2007, pp. 594-603.

[3] Melin P., Mancilla A., Lopez M., Solano D., Soto M., Castillo O., "Pattern Recognition for Industrial Security Using the Fuzzy Sugeno Integral and Modular Neural Networks", *Advances in Soft Computing*, vol. 39, no. 1, 2007, pp. 105-114.

[4] Kennedy J., Eberhart R. C., "Particle swarm optimization", *Proceedings of IEEE International Conference on Neural Networks*, Piscataway, NJ, 1995, pp. 1942-1948.

[5] Valdez F., Melin P., "Parallel Evolutionary Computing using a cluster for Mathematical Function Optimization", *Nafips*. San Diego CA, USA, June 2007, pp. 598-602.

[6] Fogel D.B., "An introduction to simulated evolutionary optimization", *IEEE transactions on neural networks*, vol. 5, no. 1, 1994, pp. 3-14.

[7] Man K.F., Tang K.S., Kwong S., "Genetic Algorithms: Concepts and Designs", *Springer Verlag*, 1999.

[8] Eberhart R. C., Kennedy J., "A new optimizer using particle swarm theory". In: *Proceedings of the 6th International Symposium on Micromachine and Human Science*, Nagoya, Japan, 1995, pp. 39-43.

[9] Goldberg D., *Genetic Algorithms*, Addison Wesley, 1988.

[10] Angeline P. J., "Using Selection to Improve Particle Swarm Optimization". In: *Proceedings of the World Congress on Computational Intelligence*, Anchorage, Alaska, IEEE, 1998, pp. 84-89.

[11] Montiel O., Castillo O., Melin P., Rodriguez, A., Sepulveda, R. "Human evolutionary model: A new approach to optimization", *Inf. Sci*, no. 177(10), 2007, pp. 2075-2098.

[12] Haupt R.L., Haupt S.E., "*Practical Genetic Algorithms*", 2nd Edition, Wiley Interscience, 2004.

[13] Rotar C., "A New Evolutionary Algorithm for Multiobjective Optimization Based on the Endocrine System". In: *Proceedings of the International Conference on Theory and Applications of Mathematics and Informatics ICTAMI*, Alba Iulia, 2003.

[14] Chang R., "*General Chemistry*", 5th edition, McGraw-Hill, 2004.

[15] Goldberg D., "*Schaum's Outline of Beginning Chemistry*", 3rd edition, Schaum's Outline Series, McGraw-Hill, 2009.

[16] *GEATbx: Example Functions* (single and multi-objective functions), http://www.geatbx.com/docu/fcnindex-01.html#P89_3085.