# New Approach to Fast and Precise Optimization with Evolutionary Algorithms

Dariusz Wawrzyniak[1,3] and Andrzej Obuchowicz[2,3]

[1] Faculty of Electrical Engineering, Computer Science and Telecommunications, email: d.wawrzyniak@weit.uz.zgora.pl
[2] Institute of Control and Computation Engineering, e-mail: a.obuchowicz@issi.uz.zgora.pl
[3] University of Zielona Góra, Poland

**Abstract.** The paper deals with an evolutionary algorithm which uses new methods to control the range of mutation. In order to significantly increase the efficiency of finding the optimum, it discovers and exploits knowledge about the state of population in environment in every generation. It allows to be found a solution both quickly and precisely. Due to the division of population into objects dealing with different functions of optimization, it can simultaneously explore as well as exploit solution space. The algorithm works in such a way that it is unable to undergo a premature convergence. There is no possibility of falling into a trap of local optimum. Therefore, it is possible to increase a selective pressure safely, for example with the help of elitist succession.

## 1 Introduction

One of the basic problems connected with using evolutionary algorithms is reconciling two mutually conflicting aims: using the best available solution and the most thorough searching of the whole accessible space of solutions, and maintaining the balance between exploitation and exploration [1, 2, 3].

Researchers have tried to solve this problem in various ways. Many different techniques were and are being applied, including: Domination and Diploidy mechanisms, adding adaptation or self-adaptation of parameters [4], maintenance of populations diversity [6], co-evolutionary genetic algorithm, Learnable Evolution Model [5], etc.

The cardinal assumptions of the new way of solving the problem:

- Division of population. Assigning the abilities of exploitation or exploration to individual objects.
- Discovering and exploiting knowledge of the state of population in an environment.

The main objective of the proposed solution is to enable an algorithm to be used to reconcile the two mutually conflicting aims, exploitation and exploration. Thanks to this, the optimum can be found as quickly as possible, fulfilling at the same time requirements concerning accuracy.

The paper is organized as follows. The description of the proposed evolutionary algorithm is presented in section 2. Our experiments and research and their results are presented in section 3. Section 4 concludes the results. The last section includes a draft of future work.

## 2 Proposed algorithm

Standard evolutionary algorithms use historically acquired knowledge. They use the features of the best solutions which have been found so far. In opposite to that, our proposal uses knowledge about every current individual location in the solution space in every generation.

Considering the standard evolutionary algorithm ESSS, the basic parameter controlling an evolutionary process is the one which controls the range of changes taking place during recombination (in the ESSS case - during mutation). Small values of the parameter cause the increase of the exploitation function and the large values increase the exploration function.

Our idea is to cause one part of the population to deal with exploitation and the other part to deal with exploration. The question is, how and to what extent these functions should be assigned to individual objects. The maximum effect can be reached, if objects which are in the neighborhood of local or global optima deal with exploitation, and objects located in less promising regions of solution space deal with exploration. The next problem is how to obtain the necessary information about current location of individual, in solution space. Such information, delivered during every generation, is the value of the fitness function of particular individuals.

After the evaluation stage, in most kinds of evolutionary algorithms, population is sorted according to fitness function. Thanks to this, in later stages, any kind of selection can be performed easier. Considering population sorted in such a way, we can notice, that the best adapted chromosomes are located closer to global or local optima. It can be also assumed (in all probability), that worse adapted objects are located further from optima.

It seems obvious, that individuals which are located closer to optimum should try to find better solutions in their near neighborhood (exploitation), and objects which are located far from optimum have much less chances to find the best solution. So, it is reasonable, to use them in order to find other optima (exploration). Moreover, research clearly shows that even small populations find optimum effectively and it would be wasteful to use the whole population to exploit (probably one of many) optimum.

We can achieve above mentioned with the help of a simple modification of algorithm. In every consecutive generation individual parameter the $\delta$ determining range of mutation [eq.(1)] must be assigned to every individual depending on its current fitness.

The first tested solution is a linear assignment of parameters to populations sorted in descending order of fitness in range from $\delta_{min}$ to $\delta_{max}$.

$$\delta_{(i)} = \delta_{min} + i * \frac{\delta_{max} - \delta_{min}}{\eta - 1} \tag{1}$$

Where: $\eta$ - population size.
i - (0,...,$\eta$-1) object index in sorted population.

Thanks to this simple operation the mutation range is strongly related to current individual location. The information is transferred from the environment to the population directly and immediately.

The proposed solution causes progressive dispersion of the population around certain optima, whereas the population performed by standard algorithms is clustered (Fig.1).

One should draw attention to the fact, that the computational cost of the modification

(a) (b)

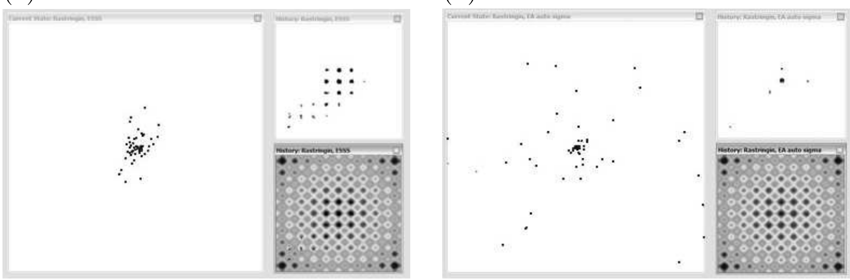

**Figure 1.** Algorithm's dispersion (a), ESSS (b) ESSS-FV.

of this algorithm is very small.

We also tested two other versions of ESSS-FV algorithm. The first of them is a version with a logistic function and the second - a simple version in which one part of population obtains $\delta_{min}$ value and the other part obtains $\delta_{max}$ value. Examples of these three kinds of distribution are presented in Fig.2. Of course we undertook coniderable research dealing with different values of parameters, which control distribution in the methods described above.

Motivation to consider these versions was to enable simultaneous increase of pressure on exploitation for that part of the population which is located next to the optimum and on exploration for that part of population which is located far from the optimum. One might raise an objection to the fact, that it is a necessary to set up parameters

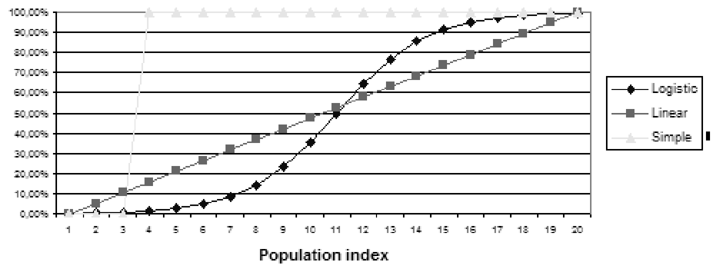


**Figure 2.** Distribution of mutation range.

$\delta_{min}$ and $\delta_{max}$. It might appear, that we brought in the necessity of setting up two parameters instead of one parameter. However, during our research it was proved that there is no necessity change these parameters in the progress of evolutionary process in addition to setting them up in depending on the kinds or nature of the environment. There is no need to choose values of these parameters precisely. A small change of the values of parameters doesn't affect the search result significantly. The parameter $\delta_{min}$

has an influence on the accuracy of found solutions. We should set it up at specific level, depending on how accurate we want to be in finding the optimal solution. Parameter $\delta_{max}$ should be set up in such a way that a chromosome would be able to obtain any value from an acceptable solution space during a mutation. It is also possible, to set up this parameter like that of a standard algorithm - at such a level which allows crossing the saddle in the local optimum trap.

## 3 Illustrative examples

Despite the fact that the presented algorithm has been created for a dynamic environment, we also tested it in a stationary one. The obtained results are presented in this paper.

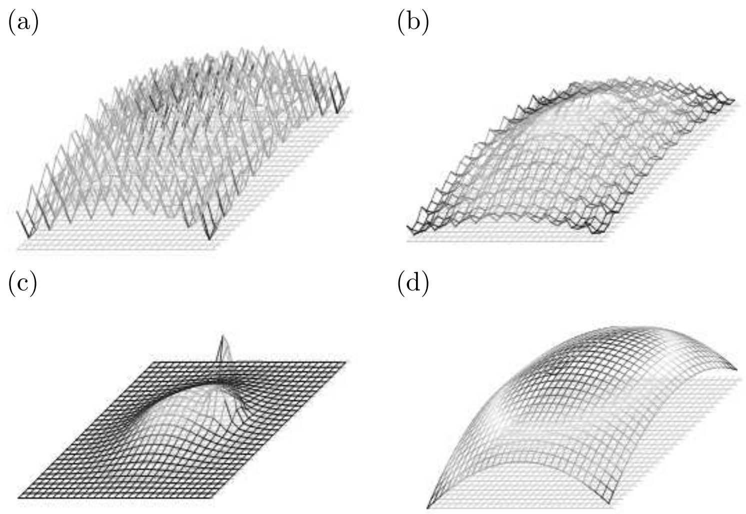


**Figure 3.** Testing functions. (a) Rastringin (b) Ackley (c) Gausian's two peaks (d) Sphere.

### 3.1 Experiments.

Algorithm was examined for three basic problems:

- To find the global optimum **(Main task)**; executed for all the functions (Fig.3).
- To find the most accurate solution **(Exploitation ability)**; executed for sphere function (Fig.3.d).
- To cross the saddle **(Exploration ability)**; executed for Gaussian function (Fig.3.c).

The following quality rates were used:

- for **Main task** - number of generations after which the global optimum can be found with certain precision (0.0001 for the space of solutions [0..1]),
- for **Exploitation ability** - Euclidean distance from the best solution to the global optimum after certain (10,20,30,100,200,300) number of generations,

- for **Exploration ability** - number of generations after which the average location of the population crosses the saddle. We assume that it is done, after the weight mean of the population is located at the higher peak.

The following kinds of algorithms were compared:

- **ESSS** - Evolutionary Search with Soft Selection [4].
- **ESSS-SVA** - ESSS with simple variation adaptation [4].
- **ESSS-ERI** - version of ESSS with elitist succession and random immigrants mechanism.
- **ESSS-FV** - ESSS with forced variation - the proposed algorithm.
- **ESSS-FV-ERI** - version of ESSS-FV with elitist succession and random immigrants mechanism.

All experiments were repeated 500 times. At the beginning for every algorithm the best parameters ($\delta, \delta_{min}, \delta_{max}$, elite group percent, random immigrants group percent) for every function were found and only after this were the algorithms compared.

The initial population (size=20) was created by adding $\eta$ times a normally-distributed random vector (with $\delta$ range) to a given initial point $x_0 \in \mathbb{R}^n$, where $x_0 = [0.5, 0.5]$ for the saddle cross problem and $x_0 = [0.1, 0.1]$ for other cases.

In ESSS, ESSS-ERI, ESSS-ERI the roulette selection method was used ,in ESSS-ERI, ESSS-ERI 'RandomRestWithRepetitions' [2] selection method was used.

### 3.2 Results.

The results for finding the global optimum are shown in Table.1 and their averages in Figure.4.a. The basic version of ESSS-FV is more effective than any older compared algorithm. ESSS-FV version with elitism and random immigrants mechanism show larger effectiveness. This improvement results mainly from the elitist succession. The results

**Table 1.** Global optimum finding.

| Functions | ESSS | ESSS-SVA | ESSS-ERI | ESSS-FV | ESSS-FV-ERI |
|---|---|---|---|---|---|
| Sphere | 9168 | 2516 | 6563 | 143 | 44 |
| Gaussian2P | 8577 | 3102 | 6099 | 5457 | 43 |
| Ackley | 6840 | 2322 | 6285 | 136 | 47 |
| Rastringin | 23466 | 3843 | 5677 | 693 | 378 |
| Average | 12013 | 2946 | 6156 | 1607 | 128 |

for the exploration ability test (without ESSS which has achieved 1765 generations) have been shown in Figure 4.b. In this test, ESSS-FV was better than ESSS-FV-ERI, because of elitism of course.

The results for the exploitation ability test have been shown in Figure.5. Figure (a) shows average results for a small number of generations, Figure (b) shows average results for larger number of generations. These examples show that more numbers of generations allow the algorithm to find a more accurate solution, even in the version without elitist mechanism. Figure 6 show the comparison of exploitation and exploration abilities in pareto form (logarithmic scale). Figure (a) shows results for a small number of generations in exploitation case, Figure (b) shows average results for larger number of generations. The obtained results confirm, that the presented algorithm has both

(a) (b)

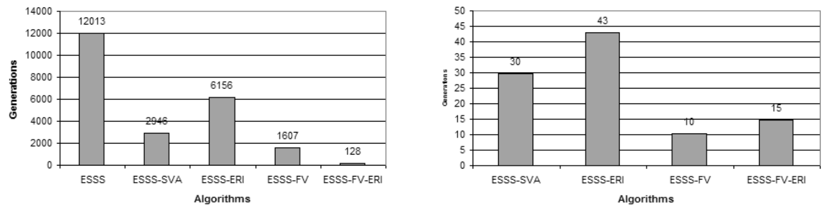


**Figure 4.** Experiments results (a) optimum finding (b) cross saddle - explorations.

(a) (b)

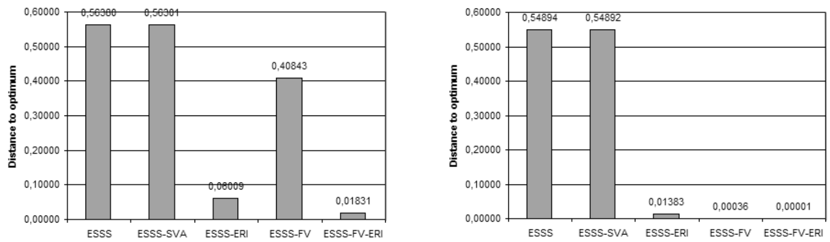


**Figure 5.** Experiments results - exploitation (a), for 10,20,30 generations (b) for 100,200,300 generations.

exploitation and exploration abilities. The ESSS-FV version in each case has the highest ability of exploration. The exploitation ability with ESSS-FV-ERI is lost in every case and with ESSS-ERI in case of a small number of generations. However, in the case of a bigger number of generations the proposed algorithm is the best. The results for finding the global optimum have been shown in Fig.7.a and for exploration test in Fig.7.b. Both algorithm modifications achieve better results than the basic version. The results for the exploitation ability test have been shown in Figure.7. Figure (c) shows results for a small number of generations, Figure (d) shows results for a larger number of generations. These examples show, that in the case of a larger number of generations, the logistic version is better than the simple version, and in case of a smaller number of generations, the logistic version is worse than the simple one. However, in both cases the algorithm modifications achieve better results than the basic version.

## 4 Conclusions

The assumed objectives have been achieved in the presented work. It has been shown that the new algorithm can be successfully applied to solve the stationary problems. The invented model of the evolutionary algorithm works very well. Its high efficiency, especially when there are a lot of local optima, is owed first of all to the fact, that it is

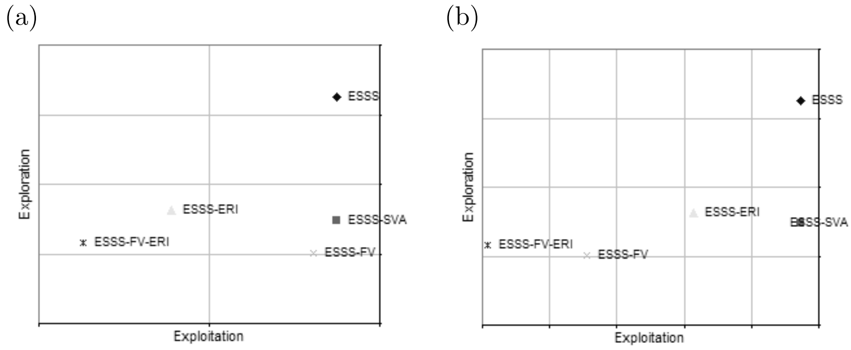


**Figure 6.** Pareto graph, exploitation vs exploration (a), for 10,20,30 generations (b) for 100,200,300 generations.

allows knowledge about the current state of population in environment to be exploited. The second very important reason for the algorithm's efficiency, is the connection of the exploitation and exploration abilities in every generation of the evolutionary algorithm. Including the elitist mechanism in the presented algorithm causes a large increase of the exploitation ability without the risk of falling into the local optimum trap. Including the random immigrants method to our algorithm doesn't cause an increase of the exploration ability (we didn't show these results), because this mechanism works worse than our method and its use causes a smaller part of population to participate in our mechanism. During the research on parameters which control $\delta$ distribution in a population we noticed that the algorithm is more effective when a larger part of population deals with the exploration rather than the exploitation ability. It must be added that ESSS-FV algorithm is unable to undergo a premature convergence and does't fall into the trap of the local optimum. One should also draw attention to the low computational cost of this algorithm modification.

Summarizing, the proposed algorithm is characterized by following features:

- Ability to discover and exploit knowledge about the state of population in environment.
- Connection of the exploitation and exploration abilities in every generation.
- Higher exploitation and exploration abilities. The global can be found faster and more precisely.
- The algorithm doesn't undergo premature convergence.
- There is no possibility of falling into the trap of the local optimum. There is always some part of population which explores the space of solutions.
- Thanks to mentioned above, we are able to increase the pressure of selection without the risk of falling into the local optimum trap.

Our future experiments will be concentrated around the dimensionality problems and study of the algorithm behavior in a dynamic environment. It seems to be interesting that there is a possibility to control the exploitation and exploration features with the help of (dynamic or adaptive) modifications rates in these algorithms.

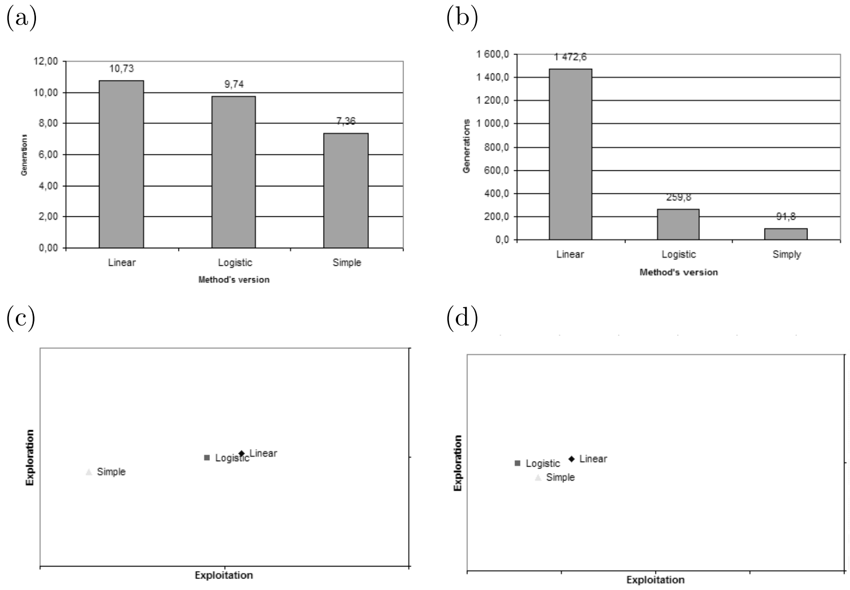


**Figure 7.** Results for ESSS-FV modification (a) global optimum search (b) exploration test, exploitation test (c), for 10,20,30 generations (d) for 100,200,300 generations.

## Bibliography

[1] J. Arabas. *Wykady z algorytmw ewolucyjnych.* Wydawnictwo Naukowo Techniczne, 2001.

[2] D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* – Addison–Wesley, Reading, MA., 1989.

[3] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs.* Springer–Verlag, 1996.

[4] A. Obuchowicz. *Evolutionary Algorithms for Global Optimization and Dynamic System Diagnosis.* Lubuskie Scientific Society Press, 2003.

[5] G. Cervone R. S. Michalski and K. Kaufman. *Speeding Up Evolution through Learning: LEM.* Proc. Ninth Int. Symp. *Intelligent Information Systems*, 2000.

[6] K. Trojanowski. *Evolutionary Algorithm with Redundant Genetic Material for Nonstatinary Environments.* Polish Academy of Science, 2003.