New Evolutionary Algorithm Modifications in Non-stationary Environments

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 the evolutionary algorithm which uses new methods which allow to increase the efficiency of finding the optimum in an environment in which fitness function is time-varying. All methods base on gathering information from environments obtained with the help of individuals (treated as sensors) and use it to help to make a decision to use a particular mechanism.

These methods include *watching procedure* which provides information about changes in the environment based on using individuals as detectors, *multiply random immigrants mechanism* which investigates a particular area of the environment, *predict procedure* which tries to calculate the next location of the optimum, *memory mechanism* which cannot be classified as any existing type of memory.

1 Introduction

In [1] we analyzed the behavior of the evolutionary algorithms in the non-stationary environment. Those researches allowed us to create a new evolutionary algorithm which was introduced and described in [6]. That algorithm allows to reconcile two mutually conflicting aims: using the best available solution and the most thorough searching of the whole accessible space of solutions, and maintaining balance between exploitation and exploration. The applied solution gives a very good dispersion of the population. In addition, the described algorithm works very effectively in the stationary environment (was presented in [5]).

Because the algorithm can easily be combined with any existing techniques (for example, random immigrants, hypermutation and restart), we searched for new propositions of other techniques which would cause a significant increase in adaptability.

First of them is a memory mechanism which cannot be classified as any in existing group of memory algorithm, whose very good review was presented in [9] and [3]. Mostly, researchers have applied some version of implicit memory [10] or explicit memory [2].

The second mechanism is an algorithm that tries to predict a new location of optimum and searching area of the environment around it, using specific version of random immigrants mechanism. That algorithm was inspired by ESSS-FDM algorithm (ESSS with forced direction of mutation) presented in [8].

The main objectives of the proposed solutions were also to find such algorithm modifications which would allow to exploit knowledge of the state of population in the environment, and could use knowledge about supposed type of the environment (moving, oscillating or random peaks).

The paper is organized as follows: the description of the proposed algorithm modification are presented in section 2, our experiments, research and their results are presented in section 3, whereas section 4 contains the results and a draft of the future work.

2 Proposed algorithm

2.1 Algorithm with memory

Memory mechanism is almost always necessary when we have to deal with a periodic environment. Almost every environment with oscillating peaks is periodic and many environments with moving peaks and certain changes of path are also periodic. Because two of three kinds of the non-stationary environments often have periodic changes and a memory mechanism seems to be required.

Our memory mechanism was inspired by the algorithm presented in [4]. It is a form of a long-term elitist. Elitism forces the EA to maintain the best element of every generation so the current optimum cannot be lost.

Our mechanism marks these elitists for saving, so they can be used for exploitation and exploration in the next generation. The basic difference between our algorithm and the proposed one [4] is a kind of the structure that is used for storing the best individuals. Its approach is based on storing the individuals in the alternative external fixed structures (for example in the second small population). Our approach is based on marking the chosen individuals. These marked individuals cannot be excluded from the population (a mechanism similar to the elitist mechanism). Such approach creates a situation in which we do not need any additional structures.

The number of best solutions in the memory is flexible (in assumed percentage maximum - even all population size). Because of the memory capacity flexibility we do not have to set the size of the memory and examine its influence on the evolutionary process.

All marked individuals constantly participate in the evolutionary process, so we do not have to bother as to when or how the information in the memory will be used.

So, from among five basic questions about applying the memory we must answer only one: which element will be added to (or removed from) the memory? In our proposition, an individual is added to the memory when it represents the best solution at least two times. It is not added to the memory when there is in an equal solution in it. When there is a similar solution in the memory, the newly added individual replaces the old one stored. This mechanism assures that the solution in the memory is getting better, and worse solutions are removed from it. If there are no similar solutions in the memory and a maximum limit for a number of the individuals in the memory has been reached, the algorithm removes the worst solution. In such a case, the program checks how many times an individual stored in the memory was used to determine which solution is better and which one is worse.

2.2 Algorithm with optimum location prediction

This algorithm watches the vector of changes of the best solution locations, calculating its averaged direction and range, and basing on this, it forecasts the new location of the optimum. It may of course, occur to be useful if we have to deal with an environment with predictable changes (for example an environment with moving peaks).

Then the algorithm uses the *multiplication random immigrant mechanism.* This part of the algorithm creates new random immigrants in surrounding of predicted location. The range of the covered area is adjusted to the range of the predicted changes. The algorithm does not employ any new features - this is the same mechanism which is used in crating an initial population around a certain point.

The assumption in case of this algorithm was to try to improve the effectiveness of the evolutionary algorithm in an environment in which the optimum solution is moving along certain path, and simultaneously assures that an additional feature does not significantly limit the exploration ability.

We can achieve the above-mentioned thanks to the fact that this feature uses a part of the population which is standardly used to generate random immigrants and thanks to this that mechanism is executed only once in every cycle.

2.3 Algorithm with changes monitoring and random immigrant multiplication

These kinds of algorithms use *changes watching procedure.* This procedure deals with situations in which the last generations of the evolutionary algorithm in every cycle (number of generation between consecutive information acquisition from the environments) does not execute exploitation and exploration (for example mutation), only the evaluation process is performed. In case of a change in the environment, every individual may gather knowledge about its fitness before and after a change. Of course, if a change does not occur, the algorithm works standardly. Basing of the gathered information from each individual, we can analyze what kind of changes occurred and try to predict where the existing optima would move and where new optima would appear etc.

In that paper we tested two kinds of changes monitoring process.

- **WC** - *watching* maximum positive *changes* in the individuals.
- **WV** - *watching* maximum *values* in changed individuals.

In both cases we sorted the population according to fitness function changes in the individuals. During the sorting procedure, first, we check if both individuals notice a positive change. In case this is false, an individual with positive change wins of course. In case this is true, we compare next parameters. In case of NEW-WC, we choose an individual with better value of fitness function *change.* In case of NEW-WV we chose an individual with better value of fitness function.

It must be added that using individuals as changes detectors is easy because of the earlier research which was focused on finding an algorithm which can assure optimal dispersion in the environment. The ESSS-FV algorithm is an excellent base for gathering information from dynamic environments.

For each kind of changes monitoring process we apply two kinds random immigrants creations.

- **OTO** - *One* random immigrant for *one* individual.
- **BTP** - *Population* of random immigrants for one *best* individual.

In the first case, we create a certain number (a parameter in the algorithm) of random immigrants so that we create one random immigrant for all best solutions (defined by changes monitoring mechanism) in the population. In the second case, we create all random immigrants around location of the best solutions. In each case, we used *multiplication random immigrant mechanism* which was mentioned above.

3 Illustrative examples

3.1 Experiments

We used a dynamic problem generator similar to DF1 [9] which can generate a wide variety of complex dynamic environments thorough application of more than one of the available types of dynamics. Our version uses a random generator instead of logistic function.

The algorithm was examined for 2-dimensional dynamic problems. Three types of environments (presented in table 1) were tested: moving, oscillating and random peaks.

Table 1. Environment description.

Type	Cones	Parameter's changes	Changes type	Fig.
Moving	1	Location	Periodic	1a
Oscillating	4	Height	Periodic	1b
Random	4	Location, Height, Slope	Random	1c

For each environment we used ten different levels of severity (the range of changes of parameters which describe functions). Thanks to this, we get environments with adiabatic, indirect and turbulent changes. Between each successive change in the environment we performed 50 generations of the evolutionary algorithm.

The basic goal of the searching process in the non-stationary environments is to keep solutions close to the optimum as much as possible. The closeness to the optimum during the search process is an interesting value which seems to be helpful in comparisons between the applications, and is easy to control in experiments. The following quality rate (tracing measure) was used:

$$\mu_{tr} = \frac{1}{t_{max}} \sum_{t=1}^{t_{max}} \rho\big(\vec{x}_{opt}(t), \vec{x}_0(t)\big), \tag{1}$$

where $\vec{x}_0(t)$ is the best point of population in the time t and $\vec{x}_{opt}(t) = \arg\max_{\vec{x}\in\mathcal{D}} \Phi(\vec{x}, t)$, $\rho(\vec{a}, \vec{b})$ is a distance measure in $\mathcal{D}$, e.g. if $\mathcal{D} \subset \mathbb{R}^n$ then $\rho(\vec{a}, \vec{b}) = \|\vec{a} - \vec{b}\|$.

The following kinds of algorithms were compared:

- **STD** - ESSS with forced variation (ESSS-FV)[6].
- **MEM** - ESSS-FV with memory.
- **PBTP** - ESSS-FV with optimum prediction mechanism.
- **WCBTP** - ESSS-FV with WC and BTP mechanisms.
- **WCOTO** - ESSS-FV with WC and OTO mechanisms.
- **WVBTP** - ESSS-FV with WV and BTP mechanisms.

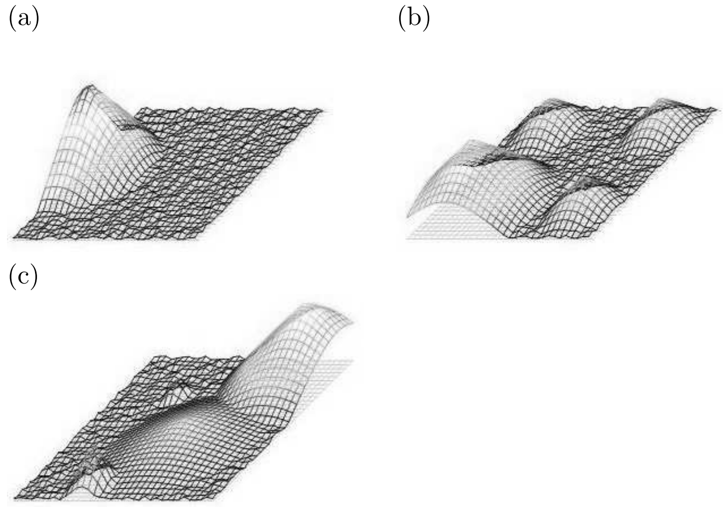

Figure 1. Testing environments. (a) Moving (b) Oscillating (c) Random peaks

- **WVOTO** - ESSS-FV with WV and OTO mechanisms.

All algorithms are based on ESSS algorithm (Evolutionary Search with Soft Selection).

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

The initial population (size=50) was created by adding η times a normally-distributed random vector (with δ range) to a given initial point. In every algorithm 'RandomRestWithRepetitions' [7] selection method was used.

3.2 Results

The averaged results are shown in figure 2a. Only the algorithm with memory are significantly better than the standard algorithm in every environment. Also the WCBTP algorithm is better than the standard algorithm, but the difference is very small. The results for an individual environment has been shown in next figures (aggregated by type, averaged by severity). The figure 2b presents results for an environment with moving peaks. In that case, almost all proposed algorithms have a higher ability to track optima than a standard one. Only an algorithm in which a method of creating random immigrants was implemented, in the form of the version OTO, is worse than the standard algorithm. Generally, this idea did not occur to be true in every type of dynamic environment and severity levels. The best results were shown by the algorithm with memory. The algorithm with prediction of new optimum location has much worse efficiency. The PTP algorithm version is also more effective than the standard version; it was true for both versions of changes monitoring mechanisms but the difference between them is very small.

The results for the environment with oscillating peaks have been shown in figure

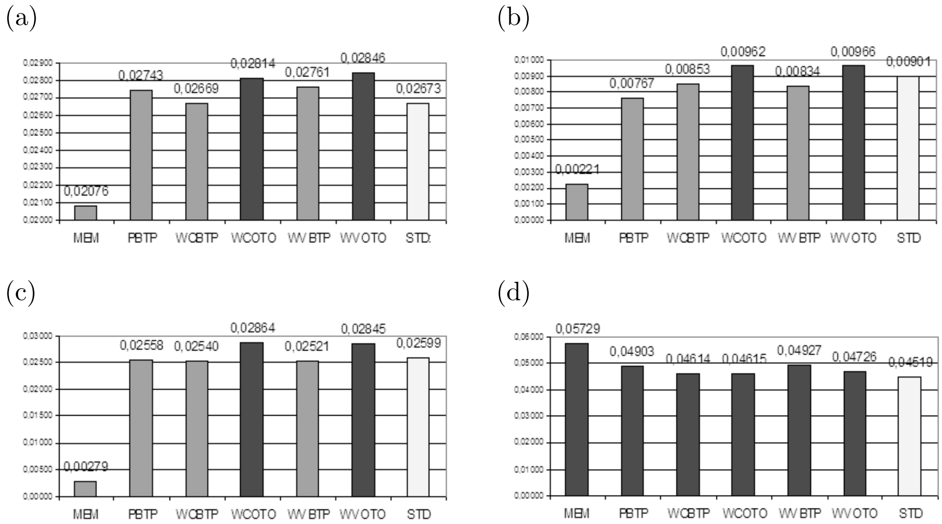

Figure 2. Tracing process (a) averaged results (b) for moving peaks (c) for oscillating peaks (d) for random peaks.

2c. In that case, the situation is very similar, but the difference between the standard algorithm and PBTP, PBTP, WVBTP is very small. Similarly to the previous case, the algorithm with memory has shown significant efficiency.

Figure 2d presents results for the environment with random peaks. Unfortunately, in that case neither of the new algorithms has got a higher ability to track optima than the standard algorithm. Only in the random environment with adiabatic changes some algorithms are better than the standard one (figure 3a). According to the expectations, the algorithm with memory has notably worse efficiency than any other methods.

The next figures (3b,3c,3d) show results grouped by severity of changes (averaged by an environment type).

These examples show that a memory mechanism works better as the severity increases. It happens because a higher severity level means a lower number of optimum locations.

4 Conclusions

The assumed objectives have been partially achieved in the presented work. It has been shown that the new algorithm modifications can be successfully applied to improve effectiveness of solving the non-stationary problems, but specific algorithms have a good efficiency in particular environment types and severity of changes value.

Summarizing, the proposed algorithms are good in the following environment:

- **MEM** - Moving and oscillating environments with any severity of changes.
- **PBTP** - Moving environment with any severity of changes, some oscillating environment with indirect severity of changes.
- **WCOTO, WVOTO** - Only moving environments with adiabatic severity of changes.

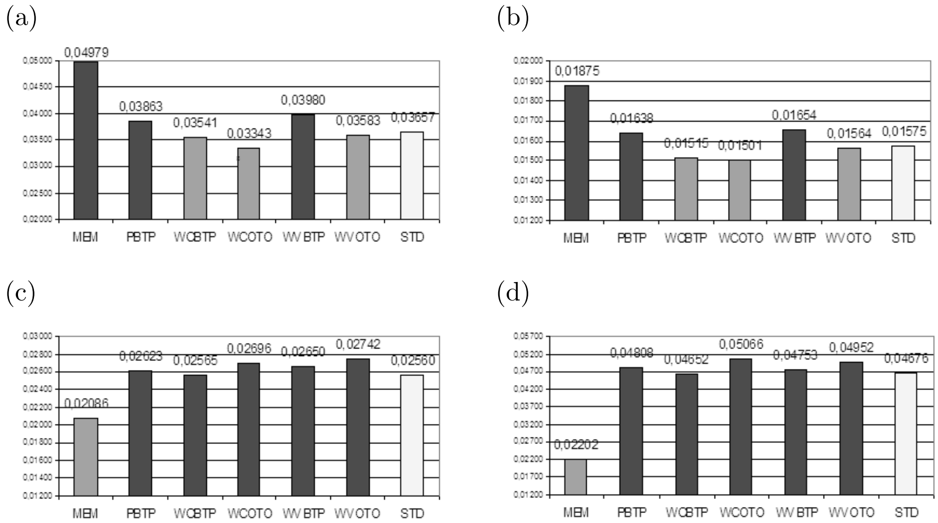

Figure 3. Tracing process (a) random peaks with adiabatic changes (b) for adiabatic changes (c) for indirect changes (d) turbulent changes.

- **WVBTP** - Moving and oscillating environments with any severity of changes.
- **WCBTP** - Moving and oscillating environments with any severity of changes and random environment with adiabatic severity of changes.

The proposed algorithms might be characterized by the following features:

- Algorithm with memory.
  - Flexible capacity of memory mechanism.
  - Information in the memory is used all the time.
  - Few modification of the algorithm to implement the memory. Using existing elitist mechanism.
- Algorithm with optimum location prediction.
  - Might be applied only in an environment with moving peaks.
- Algorithm with changes monitoring and random immigrants multiplication.
  - Might be applied in an environment with moving and oscillating peaks.
  - Watching-maximum-positive-changes mechanism works better in the environment with indirect and turbulent changes and watching-maximum-values-in-changes mechanism works better in the environment with adiabatic changes.
  - Population-of-random-immigrants-for-one-best-individual mechanism works better than 'One-random-immigrants-for-one-individual mechanism in almost any case.

Because all described methods can work simultaneously in one algorithm, our future experiments will be concentrated around a mechanism of choosing a particular kind of a mechanism depending on the kind of an environment and its changes. Of course, we must deliver a mechanism which can try to guess what the current type of the environment is. For this, we also use information which is collected within the individuals working as sensors.

Bibliography

[1] A. Obuchowicz, D. Wawrzyniak, *Evolutionary Adaptation in Non-stationary Environments: a Case Study.* In: R. Wyrzykowski, J. Dongarra, N. Meyer, J. Waśniewski (Eds.) *Parallel Processing And Applied Mathematics*, Springer (LNSC 3911) 2005, pp.439–446.

[2] C. N. Bendtsen. *Optimization of Non-Stationary Problems with Evolutionary Algorithms and Dynamic Memory.* Aarhus Universitet, 2001.

[3] J. Branke. *Memory-enhanced evolutionary algorithms for dynamic optimization problems.* Proc. IEEE Congress on *Evolutionary Computation, CEC 1999*,volume 3 pp.1875–1882.

[4] T. Drink C. N. Bendtsen. *Dynamic Memory Model for Non-Stationary Optimization.*

[5] D. Wawrzyniak, A. Obuchowicz. *New Approach To Fast And Precise Optimization With Evolutionary Algorithm.* In: J. Arabas (Ed.) *Evolutionary Computation and Global Optimization 2006*, Warsaw University of Technology Press (series: Electronics 156) 2006, pp.397–404.

[6] D. Wawrzyniak, A. Obuchowicz. *New Approach To Optimization With Evolutionary Algorithm in Dynamic Envirnment.* Proceedings of Artificial Intelligence Studies. University of Podlasie Press 2006, Vol. 3, pp.187–196 .

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

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

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

[10] A. C. Rosa V. Ramos, C. Fernandes. *Societal Implicit Memory and his Speed on Tracking Extrema in Dynamic Environments using Self-Regulatory Swarms.* Technical University of Lisbon, 2006.