## Self-Adapting, Multi-Parent Recombination Evolution Strategy Algorithm

Ivaylo Simeonov[1], Kalin Penev[2]
[1]Technical University, Sofia, Bulgaria, [2]Southampton Solent University, UK
[1]ivailosim@gmail.com, [2]kalin.penev@solent.ac.uk

**Abstract:** Evolution Strategies (ES) are algorithms similar to Genetic Algorithms (GA) which use the basic principles of natural evolution and adaptation as a method to solve optimization problems [1].

This paper focuses on Self-Adapting, Multi-Parent Recombination Evolution Strategy (MPR-ES) method applied to numerical optimization.

Proposed modified Evolution Strategy Algorithm utilises a new approach in generation of offspring from three randomly chosen individuals. It is also equipped with a mutation strategy in order to bring it out of local optima stuck problems and a self-adaptation replacement strategy ensuring that each generation of individuals is at least as good as the previous one, if not better.

**Keywords:** Evolution Strategy, Optimization, Self-Adaptation, Evolutionary Computation.

### 1. Evolutionary Computation Essentials

Evolution Strategy crystallizes in result of research efforts on improving optimization techniques [9][10]. The idea behind evolutionary computation and in particular ES is based on recreating the natural evolution processes. In nature, different individuals reproduce and create offspring of new individuals. The new generation has similarities with the parents but by no means is identical to them. Over time, the offspring generations adapt more and more to the surrounding environment so they become better – they evolve [1][4][6][10].

Sometimes unforeseen mutations happen, which in some cases prove to be useful to the individuals and lead again to evolution.

Also in nature even individuals with less opportunity for reproduction (weaker) get the chance to create offspring. This is not a prerequisite that their descendants will also be weak – in some cases this leads to more advanced successors, therefore weaker individuals should not be ignored in the process of reproduction.

The objective of presented in this paper investigation is to evaluate MPR-ES algorithm on maximization of well-known from the literature global, test functions (also called objective or fitness functions) somehow resistant to existing methodologies. The conducted experiments are for limited number of iterations and with limited computational resources.

### 2. Multi-Parent Recombination Evolution Strategy

The MPR-ES is from the type $\mu/\rho+\lambda$ [6]. Where $\mu$ is the number of generated parent individuals in the generation *G*. $\lambda$ is the number of children individuals in the generation *G.* And $\rho$ is the mutation and/or the recombination strategy used to produce children from the parents. The $\mu/\rho+\lambda$ mean that $\mu$ parents 'produce' $\lambda$ children using recombination and/or mutation. Each of the $\mu$ parents and $\lambda$ children is then assigned a fitness value (the value of the objective function). Those with best fitness (the fittest) both parents and children become next generation parents. This is the so called *multimembered ES* proposed by *Rechenberg* and later elaborated by *Schwefel* [2][8].

A maximum number of generations, achieving required optimal value or complex condition could be used as a termination criterion.

To design of MPR-ES is based on the concept have to make the following assumption. Let $S_p(M)$ be the parent individuals array, $S_c(M)$ – the offspring (children) individuals array and *G* – the number of the generations. *M* is the number of dimensions (arguments) of the individuals. The pseudo code for the algorithms is as follows:

```
Begin
  Generations = 1
  Initialize (Parents)
  Evaluate (Fitness (Parents))
  While (! TerminalCondition) {
    Parents = ParentSelection;
    Children = Recombination (Parents);
    If (mutatuion_probability){
      Children=Mutation (Children)
      Evaluate (Fitness (Children))}
   Tournament Selection (Parents, Children)
   Generations++}
End
```

The algorithm starts its work by randomly generating S individuals (real numbers) for the parent array in pre-defined search space boundaries. After that the generation of offspring individuals (recombination) begins. What makes this MPR-ES distinctive is that the successors are generated by use of three solutions. This improves the probability to create stronger and more successful new solution. In other words – using this strategy is more likely the offspring to be “scattered” around the search space and more effectively to reach the optimum of the function. The three parent solutions are randomly chosen for each child from the parents array:

$$S_c(M) = S_{p\eta}(M) + S_{p\varepsilon}(M) + S_{p\phi}(M)/3 \qquad (1)$$

The given formula is for how the children’s array is generated where $\eta$, $\varepsilon$ and $\phi$ are random numbers.

In this ESA the probability for mutation to occur is 60%. If this happens then the child individuals array is mutated. From the conducted experiments on different objective functions the most successful strategy happens to be the following one:

$$S_c(M) = S_c(M) * (X\min + random * (X\max - X\min)) \qquad (2)$$

where *Xmin* and *Xmax* are the search space lower and upper boundaries.

Similar to nature’s mutation where sometimes it helps for the faster adaptation of the individuals to the habitat, this mutation is very important, because it has the power to bring the algorithm out of local optima traps. After the recombination and the mutation (if it happens to occur) follows the *tournament selection function* (the self-adaptation strategy for this MPR-ES). This process is of great importance, because it guarantees that only the fittest (best) *S* individuals both from the parent’s or the children’s array will survive and will be able to reproduce in the next generation. The way it works is simple. Based on the fitness functions assigned to each individual only the best *S* are selected. Each of these steps is repeated in a loop until the stop criterion is met. For our case when the maximum number of generations is reached. In order to assess MPR-ES it is compared to Real-Coded Genetic Algorithm Blend Crossover alpha modification [12] with variable blend alpha [7].

## 3. Real Coded Genetic Algorithm

A computational implementation and application of Genetic Algorithms is proposed by Holland [13]. Genetic algorithms are different from other optimisation and search processes in several ways:

(1) GAs work with a coding of the parameter set, not the parameters themselves;
(2) GAs search from a population of points, not from a single point;
(3) GAs use payoff (objective function) information, not derivate or other auxiliary knowledge;
(4) GAs use probabilistic transition rules, not deterministic rules [14].

The Genetic Algorithm begins with initialisation. Initialisation is a stochastic selection and evaluation of a set of initial solutions, called initial population. The next step is to form a new population. Generation of a new population consist of the events modification, replacement and evaluation.

During the modification, individuals from the current population are taken and used for creation of an offspring. Most often for generation of a new individual, two individuals from the current population are selected. These individuals are called parents. Recombination between selected parents produces an offspring. This is motivated by expectations, that the new individual can be better than its parents. To generate a solution different from selected parents the offspring can be a subject of mutation.

Mutation generates a random value for the offspring. Mutation can appear with certain, small probability. After modification follow the events evaluation and replacement. The Genetic Algorithm uses a preliminary defined objective (fitness) function included as part of the evaluation. For any particular optimisation problem the fitness function is different. After the event replacement, follows the event termination. Unless termination criterion (for example expiration of the number of iterations or improvement of the best solution) is met, repetition of the events modification and replacement continues.

Used in this study GA is with Blend crossover modification strategy called also BLX-alpha [12]. For BLX-alpha modification strategy, the offspring is a random location within the area determined by selected parents and extended with a blend interval alpha. The mathematical description of BLX-alpha modification strategy is presented with equation below.

$$X_c = X_{p1} - \alpha + (X_{p2} - X_{p1} + 2) * random(0,1) \quad (3)$$

where $X_{p2}$ and $X_{p1}$ are selected parents, $X_{p2} > X_{p1}$, $\alpha$ is a blend around the selected parents, *random*(0,1) generates a random value between 0 and 1.

The extension of the space, between selected parents, increases the chances of the algorithm to reach the appropriate solution if it is near to the area determined by the parents. Increasing the chances to reach the optimum is an advantage. Variation of the blend $\alpha$ can be used for tuning of the convergence and divergence of the search process. Therefore, the concept for extension of the space for modification by a blend $\alpha$ is considered as valuable for improvement of the performance of the search process.

For the purposes of the investigation the Genetic Algorithms BLX-$\alpha$ is modified and implemented with a variable blend $\alpha$ [7].

## 4. Black Box Model for Test Definition

For this study in order to guarantee fair and equal condition for exploration or any search task a model for Black box definition of optimisation test is applied for both Multi-Parent

Recombination Evolution Strategy and Real Coded Blend Crossover Genetic Algorithm is applied.

The main characteristic of used Black box model is that the objective and constraint functions are explicitly not known for the search method [15].

Appling the Black box concept to numerical problems involves implementation of an interface between data space defined by the objective function and the optimisation process. If optimisation methods have no abilities for adaptation the black box may not be feasible. Both used methods are implemented in a manner, which guarantee sufficient for black box search level of adaptation. An ability of the algorithm to explore, to learn and to cope with unknown problems usually leads to the improvement of performance and effectiveness of the optimisation process.

Presented study uses four test functions for testing MPR-ES and comparing it to GA BLX alpha.

All tests are implemented in Black Box model and do not require modification of the methods or tuning their parameters for each particular test. MPR-ES and GA BLX alpha have to adapt to explored tasks.

**5. Test Functions**

The numerical test functions used for experiments are presented in this section; these are Ackley test [3], Rosenbrock test [9], Rastrigin test [11], and Griewank test [5].

The objective function of Ackley test [3] is:

$$f\ (x) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos\left(2\pi x_i\right)\right) + 20 + \exp(1) \qquad (4)$$

The maximal value is *f* (0.0, 0.0) = 0.0.

The Rosenbrock test function [11] is:

$$f\ (x) = \sum_{i=1}^{D-1}\left(100\left(x_i^2 - x_{i+1}\right)^2 + \left(x_i - 1\right)^2\right) \qquad (5)$$

The maximal value is *f*(1.0, 1.0) = 0.0.

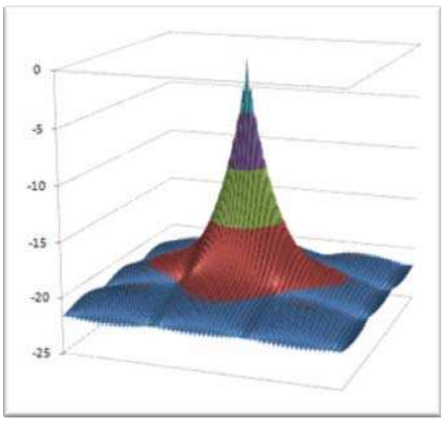


**Figure 1:** Ackley function

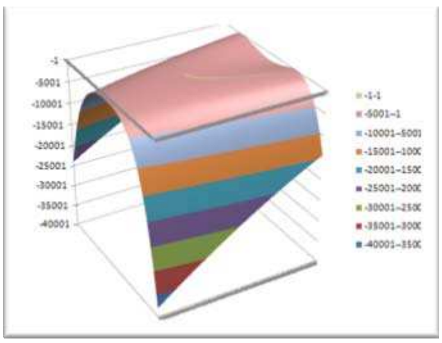


**Figure 2:** The Rosenbrock function

The objective function Rastigin test [9] is:

$$f(x)=\sum_{i=1}^{D}\left(x_i^2-10\cos\left(2\pi x_i\right)\right)+10*D \tag{6}$$

The maximal value is *f*(0.0, 0.0) = 0.0.

The Griewank test objective function [5] is:

$$f(x)=\sum_{i=1}^{D}\frac{x_i^2}{4000}-\prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1 \tag{7}$$

The maximal value is *f*(0.0, 0.0) = 0.0.

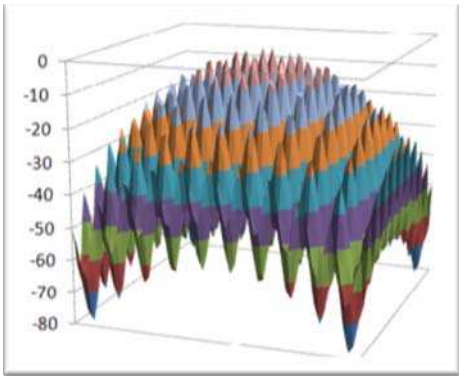


**Figure 3:** The Rastrigin function

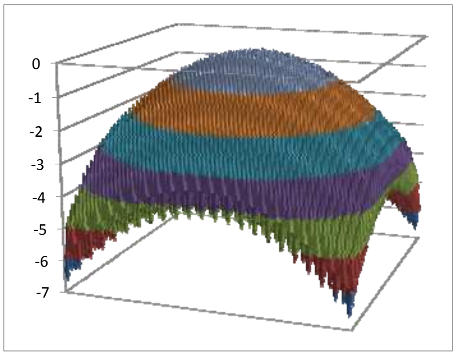


**Figure 4:** The Griewank function

## 6. Experimental Results

All experiments on the objective functions for optimization with Multi-Parent Recombination Evolution Strategy and Real Coded Genetic Algorithm Blend Crossover alpha with variable blend are conducted with the same initial parameters for both algorithms. The parameters are as follows:

- The number of individuals in a generation is 10. This is due to demonstrate that even a small number of random generated individuals is capable of reaching the optimum by recombination, mutation and adaptation.
- All tests are used in their two dimensional *M*=2 variants.
- The number of maximum generations for MPR ES limited to 2000 and for GA BLX alpha to 20000 accordingly. The reason for this is that for one iteration MPR ES produces 10 new individuals and GA BLX alpha only one. These numbers of iterations guarantee equal number of fitness function calculations required for fair comparison.
- Probability for mutation to occur for MPR ES is 60% and for GA BLX alpha 40%. The percentages are relatively high but since mutation is of a great importance for global optimization they are justified. Blend alpha for GA BLX varies from 0.5 to 1.5 with step 0.1. With variable blend GA BLX alpha produces 320 results per test. For comparison is used best achieved result.

Although the results are shown for the 2000-th generation in some cases the MPR ES leads to the presented result before the 500-th generation.

Experimental results are presented in the tables below.

| **Table 1:** Ackley test | | |
|---|---|---|
| Results for MPR ES | | |
| Search range | Optimum | 2000G Result |
| [-32; 32] | 0 | -0.000004 |
| Results for GA BLX alpha | | |
| [-32; 32] | 0 | -0.000202 |

| **Table 2:** Rosenbrock test | | |
|---|---|---|
| Results for MPR ES | | |
| Search range | Optimum | 2000G Result |
| [-1000; 1000] | 0 | -0.000000 |
| Results for GA BLX alpha | | |
| [-1000; 1000] | 0 | -0.000000 |

| **Table 3:** Rastrigin test | | |
|---|---|---|
| Results for MPR ES | | |
| Search range | Optimum | 2000G Result |
| [-1000; 1000] | 0 | -0.000000 |
| Results for GA BLX alpha | | |
| [-1000; 1000] | 0 | -0.000101 |

| **Table 4:** Griewank test | | |
|---|---|---|
| Results for MPR ES | | |
| Search range | Optimum | 2000G Result |
| [-1000; 1000] | 0 | -0.000000 |
| Results for GA BLX alpha | | |
| [-1000; 1000] | 0 | -0.040255 |

Experimental results confirm the expectations that Multi-Parent Recombination Evolution Strategy will adapt to explored numerical tests.

MPR ES resolves Ackley test, Rosenbrock test, Rastrigin test, and Griewank test without resetting or retuning of its optimisation parameters with precision above 0.0000001.

For comparison and Real Coded Genetic Algorithm Blend Crossover alpha resolves Rosenbrock test with the same precision, Ackley test and Rastrigin test with precision 0.0001, and for Griewank test needs additional iterations or perhaps retuning of mutation probability.

## 7. Conclusions

The Multi-Parent Recombination Evolution Strategy reviewed in this paper presents a distinctive approach in the offspring generation from three different parent individuals. The conducted experiments showed that this strategy, along with the mutation strategy successfully leads the algorithms to the optimal value of most of the objective functions even before it has reached the present maximum number of iterations (generations). Also equipped with the self-adaptation strategy and supported by it guarantees that each generation is at least as good as the previous one if not better.

The conducted experiments and the results from them show that there is no need of a large number of individuals in a generation to achieve the objective – in our case reaching the optimum.

Moreover the small number of individuals in the population combined with the relatively simple strategies for recombination, mutation and self-adaptation leads to very fast computation of the solutions.

This in turn opens possibilities before the MPR ES for use in dynamic real-time applications such as transmit power minimisation in wireless communications, in Quality of Service (QoS) related to IP telephony where specific requirements on signal-to-noise ratio and loudness levels are needed. Also it has great pottential in the field of pattern recognition in broadcast signals (wireless networking).

The results from the experiments show that this Multi-Parent Recombination Evolution Strategy based on its simplicity can turn in a powerful tool for optimization.

Further research could focus on evaluation of MPR ES on multidimensional, constrained and time dependent tests.

It will be a challenge to see how Evolution Strategies perform on numerical tests where the optimism is unknown and to compare their performance to other search algorithms.

**References**

1. Beck, T., 1996, "*Evolutionary Algorithms in Theory and Practice: Evolution strategies, Evolutionary programming, genetic algorithms"*, Oxford.
2. Beck, T., and Hoffmeister, F., and Schwefel, H.-P., 1991, "*A survey of evolution strategies*", University of Dortmund, Dpartament of computer science.
3. D. H. Ackley., 1987 "*A connectionist machine for genetic hillclimbing*". Boston: Kluwer Academic Publishers.
4. Fogel, D., B., 2002, "*Evolutionary Computing*", IEEE Press.
5. Griewank, A. O., 1981 "*Generalized Decent for Global Optimization*." *J. Opt. Th. Appl.*, Vol. 34, pp. 11-39.
6. Beyer H.G., Schwefel, H.P., Evolution Strategies: A Comprehensive Introduction, In: Natural Computing: an international journal, Vol. 1, No. 1 Hingham, MA, USA: Kluwer Academic Publishers, 2002, pp. 3-52.
7. Penev, K., Littlefair, G., 2003, "*Free search a novel heuristic method*", Solent university Southampton, pp. 133-134.
8. Rechenberg, I., 1973, "*Evolutionsstrategie: Optimierung technischer Systeme und Prinzipien der biologischen Evolution"*, Frommann-Holzboog, Stuttgart.
9. Rosenbrock, H. H., 1960, "*An automatic method for finding the greatest or least value of a function*", *The Computer Journal*, Vol. 3, pp. 175-184.
10. Schwefel, H.-P., 1995, "*Evolution and Optimum Seeking*", Wiley.
11. Törn and A. Zilinskas., 1989 "*Global Optimization*". Lecture Notes in Computer Science, Vol. 350, Springer-Verlag, Berlin.
12. L. J. Eshelman, J. D. Schaffer, Real-coded genetic algorithms and interval-schemata, Foundations of Genetic Algorithms 2, Morgan Kaufman, pp 187-202, San Mateo, 1993.
13. Holland J., 1975, Adaptation in Natural and Artificial Systems, University of Michigan Press.
14. Goldberg D.E., 1989, Genetic Algorithms in Search, Optimisation, and Machine Learning, Addison Wesley Longman Inc. ISBN 0-201-15767-5.
15. Brekelmans R., Driessen L., Hamers H., Hertog D., 2001, A new Sequential Optimisation Approach to Product and Process Design Involving Expensive Simulations, Third ASMO UK / ISSMO Conference on Engineering Design Optimisation.