Tuning Parameters of Differential Evolution: Self-adaptation, Parallel Islands, or Co-operation

Christina Brester[1,2][a] and Ivan Ryzhikov[1,2][b]
[1]Department of Environmental and Biological Sciences, University of Eastern Finland, Kuopio, Finland
[2]Institute of Computer Science and Telecommunications, Reshetnev Siberian State University of Science and Technology, Krasnoyarsk, Russia

Keywords: Optimization, Evolutionary Algorithms, Differential Evolution, Tuning Parameters, Self-adaptation, Parallel Islands, Co-operation, Performance.

Abstract: In this paper, we raise a question of tuning parameters of Evolutionary Algorithms (EAs) and consider three alternative approaches to tackle this problem. Since many different self-adaptive EAs have been proposed so far, it has led to another problem of choice. Self-adaptive modifications usually demonstrate different effectiveness on the set of test functions, therefore, an arbitrary choice of it may result in the poor performance. Moreover, self-adaptive EAs often have some other tuned parameters such as thresholds to switch between different types of genetic operators. On the other hand, nowadays, computing power allows testing several EAs with different settings in parallel. In this study, we show that running parallel islands of a conventional Differential Evolution (DE) algorithm with different CR and F enables us to find its variants that outperform advanced self-adaptive DEs. Finally, introducing interactions among parallel islands, i.e. exchanging the best solutions, helps to gain the higher performance, compared to the best DE island working in an isolated way. Thus, when it is hard to choose one particular self-adaptive algorithm from all existing modifications proposed so far, the co-operation of conventional EAs might be a good alternative to advanced self-adaptive EAs.

1 INTRODUCTION

Evolutionary algorithms (EAs) are flexible and widely applicable tools for solving optimization problems. Their effectiveness has been demonstrated in many studies and international competitions on black-box optimization problems. Nowadays, EAs are successfully used in machine learning, deep learning, and reinforcement learning. Therefore, more and more effective heuristics and meta heuristics are being developed and their beneficial properties and convergence are being investigated in the scientific community.

In spite of all positive features, EAs require tuning a set of parameters for their effective work, which is a non-trivial task even for the experts (Eiben et al., 2007). The main issue is that optimal values of the parameters tuned differ for various problems. The "No Free Lunch" theorem proves this phenomenon and claims that there is no one search algorithm working best for any kind of optimization problem (Wolpert and Macready, 1997). As a result, an idea of parameter self-adaption has been proposed to find their proper values for the problem solved during the algorithm execution (Meyer-Nieberg and Beyer, 2007).

The primary approaches to the EA self-adaptation are normally based on one of the following concepts. Firstly, it might be a deterministic scenario, according to which parameters are changed during iterations. As an example, in Daridi's work, a mutation probability is presented as a function of a generation number (Daridi, 2004). Secondly, several types of genetic operators (different implementations of selection, crossover, and mutation) may compete for resources based on their effectiveness in previous generations. For instance, Khan and Zhang (2010) used two crossover operators to produce offspring which probabilities of being applied were recalculated in each generation based on the success rate (how many times a new solution outperformed at least one of the

[a] https://orcid.org/0000-0001-8196-2954
[b] https://orcid.org/0000-0001-9231-8777

Brester, C. and Ryzhikov, I.
Tuning Parameters of Differential Evolution: Self-adaptation, Parallel Islands, or Co-operation.
DOI: 10.5220/0008495502590264
In *Proceedings of the 11th International Joint Conference on Computational Intelligence (IJCCI 2019)*, pages 259-264
ISBN: 978-989-758-384-1

parents). Alternatively, parameters of genetic operators might be included in a chromosome, evolve in iterations and be applied to generate the next candidate solution (Pellerin et al., 2004).

However, a booming interest in self-adaptation has resulted in many proposed techniques and again caused a problem of choice. Moreover, in some approaches, self-adaptive operators use a number of thresholds to switch between different types of a genetic operator. These thresholds also should be selected properly. On the other hand, due to the impressive computing power available nowadays, it became possible to test various settings of the algorithm in parallel, which might be an alternative approach to self-adaptation.

Nevertheless, in some studies, it has been shown that at different stages of optimization, certain types of genetic operators are beneficial for the search (Tanabe and Fukunaga, 2013). Self-adaptive EAs support this replacement of operators in generations, whereas EAs with diverse settings run in parallel do not provide this option. At the same time, incorporating a migration process into parallel EAs, *i.e.* the exchange of solutions, and creating a co-operation of EAs with different settings allows introducing candidate solutions generated by various EAs operators in the population.

Therefore, in this study, we compare several self-adaptation techniques with parallel EAs and their co-operation having three variants of its topology based on the example of a Differential Evolution (DE) algorithm, which needs tuning *CR* and *F* parameters (Storn and Price, 1997). Since DE is one of the most effective and widely used heuristics, it is essential to investigate different approaches to tune its key parameters.

2 METHODS COMPARED

The general DE scheme for a minimization problem contains the following steps:

– Randomly initialize the population of size *M*: $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$;

– Repeat the next operations until the stopping criterion is satisfied:

- For each individual $\mathbf{x}_i$, $i = \overline{1, M}$, in the population do:

1. Randomly select three different individuals from the population **a, b, c** (which are also different from $\mathbf{x}_i$);

2. Randomly initialize an index $R \in \{1, \ldots, n\}$, where *n* is the problem dimensionality;

3. Generate a mutant vector. For each $j \in \{1, \ldots, n\}$, define $r_j \sim unif(0, 1)$. Next, if $r_j < CR$ or $j = R$, then $x'_j = a_j + F \cdot (b_j - c_j)$, otherwise $x'_j = x_{ij}$. CR and F are the DE parameters.

4. If $f(\mathbf{x}') < f(\mathbf{x}_i)$, then replace $\mathbf{x}_i = \mathbf{x}'$.

As a basis of this work, we used algorithms implemented in the PyGMO library (Biscani et al., 2018). There are two self-adaptive versions of DE called SaDE and DE1220, wherein two variants of *CR* and *F* control and adaptation are available, particularly, jDE (Brest et al., 2006) and iDE (Elsayed et al., 2011). In SaDE, a mutant vector is produced using a DE/rand/1/exp strategy (by default and in our experiments too), whereas in DE1220 the mutation type is coded in a chromosome and also adapted.

In addition to self-adaptive algorithms, we applied a conventional DE with different values of *CR* and *F* parameters: $CR = [0.3, 0.5, 0.7, 0.9]$ and $F = [0.3, 0.5, 0.7, 0.9]$. Using the *island* class of PyGMO, we could run DEs with different settings in parallel threads to save computational time.

Next, we extended the PyGMO library with a set of functions implementing the migration process among the parallel islands. In this study, three topologies of the island co-operation are investigated: *Ring*, *Random*, and *Fully Connected*. After each T_m generations, N_{best} individuals with the highest fitness from every population are sent to other islands to substitute N_{worst} solutions having the lowest fitness there.

In the *Ring* topology (Figure 1), at every migration stage, solutions are sent along the same route, *i.e.* from the *i*-th island to the (*i*+1)-th one. Island numbers keep constant during the search. Every (*i*+1)-th accepts $\min(N_{best}^{\,i}, N_{worst}^{\,i+1})$ solutions to replace the worst individuals in its population.

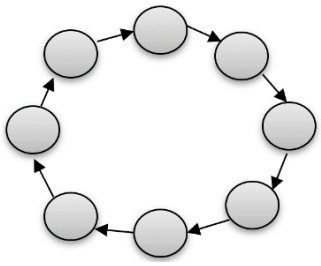

Figure 1: Ring topology.

In the *Random* topology (Figure 2), at each migration stage, for every *j*-th island, where $j = \overline{1, M}$ and *M* is the total number of islands in the co-operation, the *i*-th island, sending the best individuals to it, is chosen randomly so that $i \neq j$. The *j*-th island accepts $\min(N_{best}^{\,i}, N_{worst}^{\,j})$ solutions.

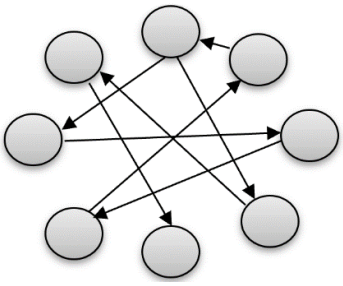

Figure 2: Random topology.

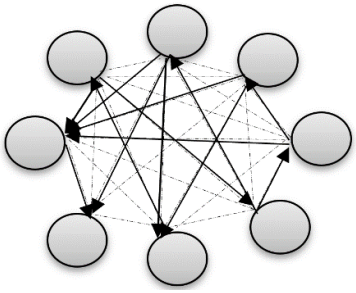

Figure 3: Fully Connected topology.

In the *Fully Connected* topology (Figure 3), every *j*-th island randomly selects $N_{worst}{}^{j}$ solutions from the united set of the best individuals from all other islands. Thus, at each migration stage, every island is likely to get solutions from several other islands.

These three groups of DE (self-adaptive, conventional with different CR and F, and co-operative) were tested using the generalized *n*-dimensional Rosenbrock function (n = 10, 30, 100):

$$F(x_1, \dots, x_n) = \sum_{i=1}^{n-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2],$$

$$x_i \in [-5, 10]. \quad (1)$$

The global optimum is $x_i = 1, F(1, \dots, 1) = 0$.

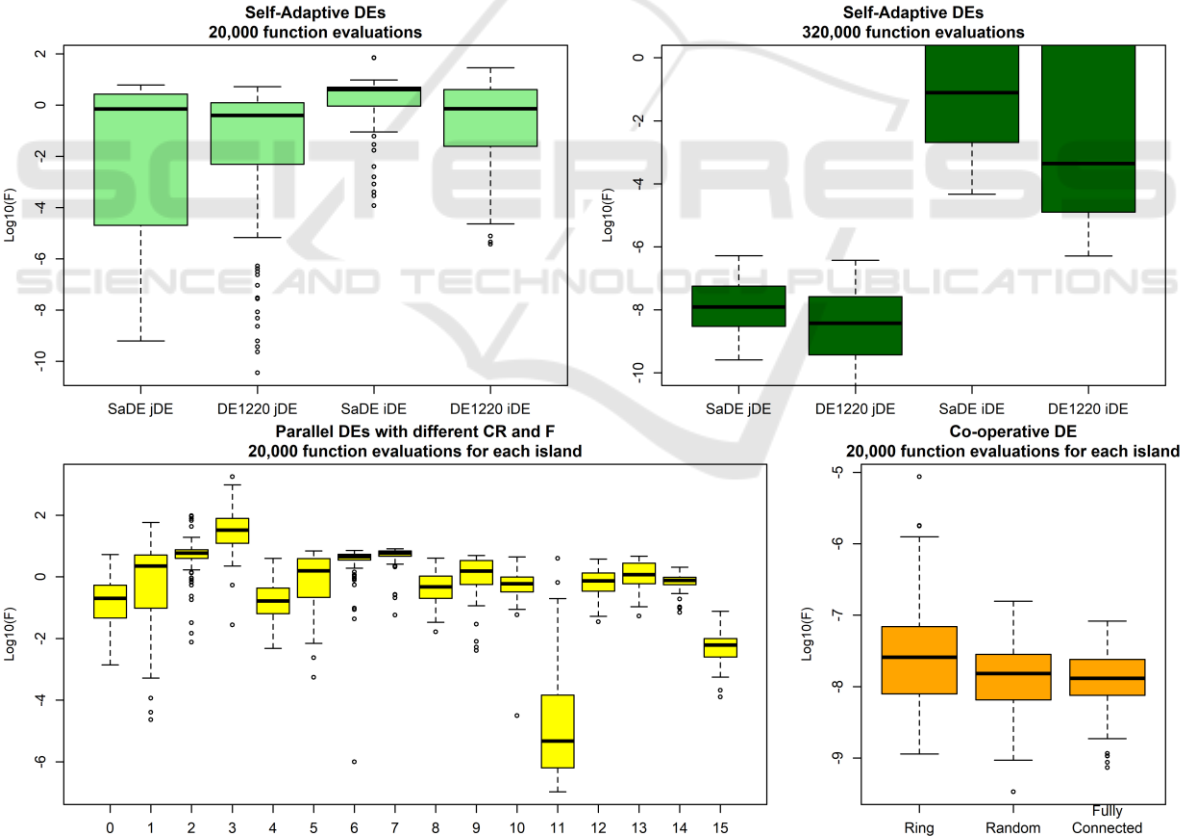

Figure 4: Experimental results for the Rosenbrock function (n = 10).

3 EXPERIMENTAL RESULTS

Four self-adaptive DEs, namely, SaDE and DE1220 with the jDE variant of self-adaptation, SaDE and DE1220 with the iDE variant of self-adaptation, were applied to solve the Rosenbrock problem with 10 (Figure 4), 30 (Figure 5), and 100 (Figure 6) variables. In the first experiment, the amount of resources for each algorithm was 20,000 function evaluations. Every algorithm was run 100 times.

As one can see, these self-adaptive DEs demonstrate different effectiveness on the problems solved, which implies that there is also a necessity to choose an effective self-adaptive modification of the algorithm from various existing variants.

Next, we ran 16 variants of the conventional DE algorithm in parallel (again 100 times), with different values of CR and F. Each island was provided with the same amount of resources: 20,000 function evaluations. In Figures 4–6, we denoted the following pairs of (CR; F) with numbers 0…15:

Table 1: Values of CR and F parameters tested.

ID	CR	F		ID	CR	F
0	0.3	0.3		8	0.3	0.7
1	0.5	0.3		9	0.5	0.7
2	0.7	0.3		10	0.7	0.7
3	0.9	0.3		11	0.9	0.7
4	0.3	0.5		12	0.3	0.9
5	0.5	0.5		13	0.5	0.9
6	0.7	0.5		14	0.7	0.9
7	0.9	0.5		15	0.9	0.9

Based on the results obtained, we may conclude that the conventional DE with many variants of the tested combinations of CR and F is able to compete with its self-adaptive modifications and even outperforms them in some cases. Moreover, due to available computing resources and possible parallelization, several conventional DE with various settings might be easily compared. Therefore, it is arguable and really depends on the situation, what to choose, multiple tests or one of self-adaptive modifications of the algorithm.

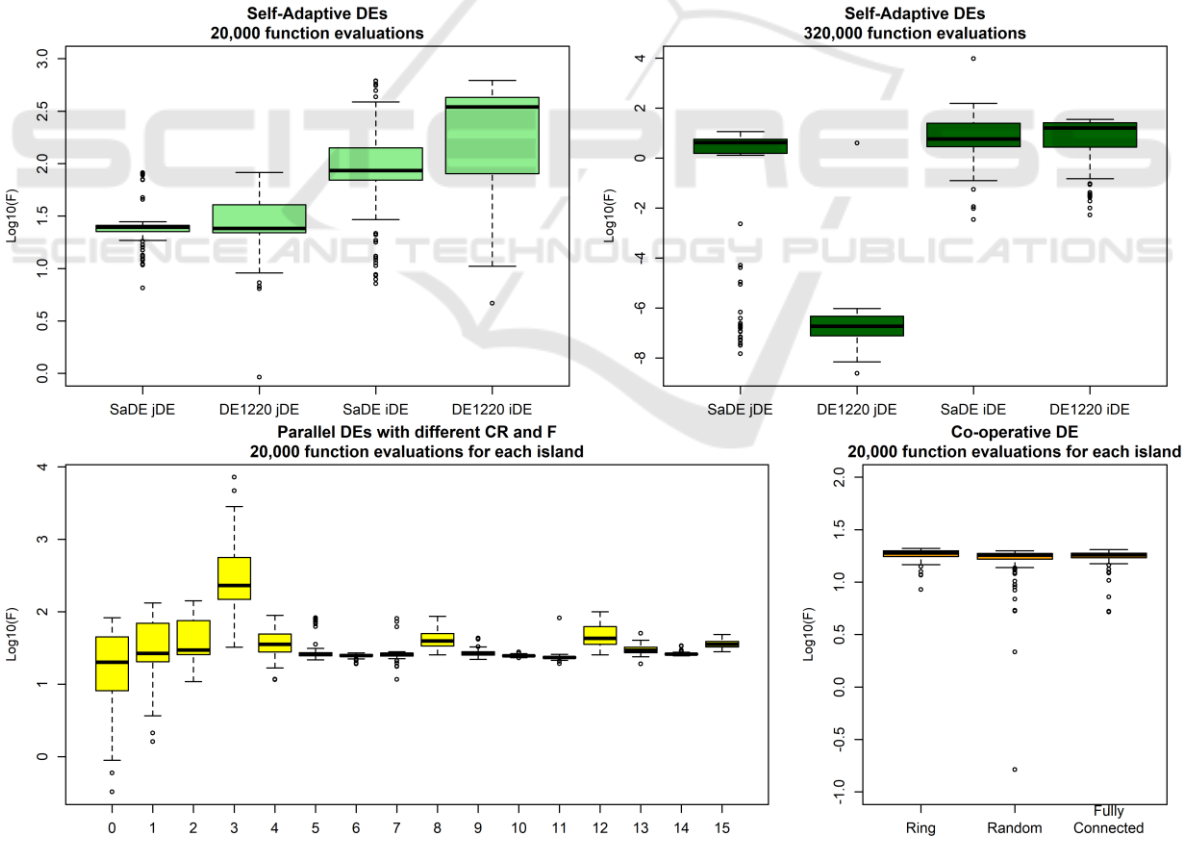

Figure 5: Experimental results for the Rosenbrock function (n = 30).

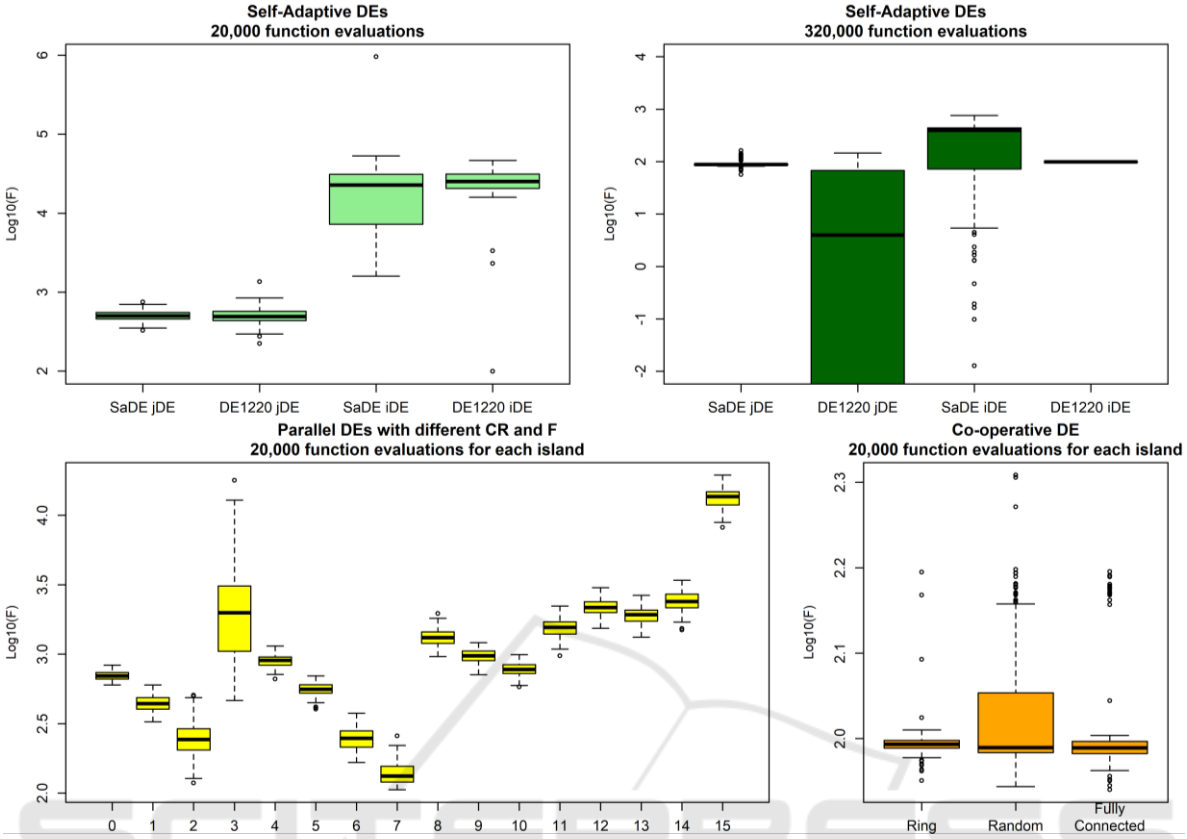

Figure 6: Experimental results for the Rosenbrock function (n = 100).

Then, we allowed the 16 parallel islands with different values of CR and F to exchange the best solutions among each other. We investigated three schemes of the migration process, in particular, *Ring*, *Random*, and *Fully Connected Topology*. Every island had the population of 20 individuals evolving for 1000 generations. The migration parameters were defined as follows: T_m = 20, N_{best} = N_{worst} = 4. According to the experimental results, we may point out that interactions among parallel islands are favorable for the performance of the whole co-operation since it typically outperforms the best island evolving in an isolated way. In the experiments conducted, the fully connected topology has shown the high performance steadily.

Moreover, the effectiveness of the presented co-operation is often comparable with the results of self-adaptive modifications having the same amount of resources as the whole co-operation (16 x 20,000). In comparison with self-adaptive DEs, the co-operation of conventional DEs has a parallel structure and requires less computational time for the algorithm execution. Furthermore, it is quite interesting that using just primitive versions of DE and applying migration, we can achieve the same effectiveness as some recently developed and advanced self-adaptive modifications demonstrate.

4 CONCLUSIONS

In this study, we raised a question of tuning parameters of EAs and compared three approaches, which were self-adaptation, testing different settings in parallel islands, and co-operation. To ponder this question and investigate these approaches, we chose one of the most popular EAs nowadays, namely, Differential Evolution.

Firstly, the self-adaptive modifications of DE showed different performances on the test problems. This implies that the solution quality depends on our choice of the self-adaptation strategy, which typically has some tuned parameters too.

Next, due to possible parallelization, various settings of DE might be checked in parallel threads and the best combination of them is likely to provide better results than some self-adaptive modifications. However, without parallelization, it takes more time, and sometimes it is not the option.

Finally, we showed that the migration process among parallel islands of conventional DEs with different CR and F helped to achieve the higher performance than the best of these islands showed while working in an isolated way. The co-operation of conventional DEs has a parallel structure and allows reducing computational time, whereas some self-adaptive DEs do not show better results even having the same amount of resources as the whole co-operation uses. Therefore, the co-operation of conventional DEs might be considered as an alternative to advanced self-adaptive DEs in some cases.

In the future work, we are planning, firstly, to test co-operations of advanced self-adaptive DEs and, secondly, make islands compete for resources in the co-operation. More test problems will be used for that comparative analysis.

ACKNOWLEDGEMENTS

The reported study was funded by Russian Foundation for Basic Research, Government of Krasnoyarsk Territory, Krasnoyarsk Regional Fund of Science, to the research project: 18-41-242011 «Multi-objective design of predictive models with compact interpretable strictures in epidemiology».

REFERENCES

Biscani, F., Izzo, D., Märtens, M., 2018. Pagmo 2.7 URL https://doi.org/10.5281/zenodo.1217831

Brest, J., Greiner, S., Bošković, B., Mernik, M., Zumer, V., 2006. Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646-657.

Daridi, F., Kharma, N., Salik, J., 2004. Parameterless genetic algorithms: review and innovation. *IEEE Canadian Review*, (47), pp. 19–23.

Eiben, A., Michalewicz, Z., Schoenauer, M., Smith, J., 2007. Parameter Control in Evolutionary Algorithms. In: Lobo F., Lima C., Michalewicz Z. (eds.) Parameter Setting in Evolutionary Algorithms, Springer, pp. 19–46.

Elsayed, S.M., Sarker, R.A., Essam, D.L., 2011. Differential evolution with multiple strategies for solving CEC2011 real-world numerical optimization problems. *In Evolutionary Computation (CEC)*, 2011 IEEE Congress on, pp. 1041-1048.

Meyer-Nieberg, S., Beyer, H.G., 2007. Self-Adaptation in Evolutionary Algorithms. In: Lobo F.G., Lima C.F., Michalewicz Z. (eds) Parameter Setting in Evolutionary Algorithms. Studies in Computational Intelligence, vol. 54. Springer, Berlin, Heidelberg.
DOI https://doi.org/10.1007/978-3-540-69432-8_3

Pellerin, E., Pigeon, L., Delisle, S., 2004. Self-adaptive parameters in genetic algorithms. *Proc. SPIE 5433, Data Mining and Knowledge Discovery: Theory, Tools, and Technology VI.* DOI: 10.1117/12.542156

Storn R. and Price K., 1997. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization, vol. 11, no. 4, pp. 341–359.

Tanabe R. and Fukunaga A., 2013. Success-history based parameter adaptation for Differential Evolution. 2013 IEEE Congress on Evolutionary Computation, Cancun, 2013, pp. 71-78. doi: 10.1109/CEC.2013.6557555

Wolpert, D.H. and Macready, W.G., 1997. No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82.