Parallel differential evolution with variable population size for global optimization

Iztok Fister[1,3]*, Andres Iglesias[2,3], Akemi Galvez[2,3], Dušan Fister[4], and Iztok Fister Jr.[1]

[1] University of Maribor, Faculty of Electrical Engineering and Computer Science,
Koroška cesta 46, Slovenia
[2] Toho University, 2-2-1 Miyama, 274-8510, Funabashi, Japan
[3] University of Cantabria, Avenida de los Castros, s/n, 39005 Santander, Spain
[4] University of Maribor, Faculty of Economics and Business, Razlagova 14, Slovenia

Abstract. The results of evolutionary algorithms depends on population diversity that normally decreases by increasing the selection pressure from generation to generation. Usually, this can lead evolution process to get stuck in local optima. The study is focused on mechanisms to avoid this undesired phenomenon by introducing parallel differential evolution that decompose a monolithic population into more variable-sized sub-populations, which evolve independently of each other. The proposed parallel algorithm operates with individuals having some characteristics of agents, e.g., they act autonomously by selecting actions, with which they affect the state of environment. This incorporates two additional mechanisms: aging, and adaptive population growth, which direct the decision-making by individuals. The proposed parallel differential evolution was applied to the CEC'18 benchmark function suite, while the produced results were compared with some traditional stochastic nature-inspired population-based and state-of-the-art algorithms.

Keywords: Differential evolution · Variable population size · Aging mechanism · Autonomous agent

1 Introduction

Usually, researchers in Evolutionary Algorithm (EA) community are confronted with the question: How to maintain a diversity of population in the conditions of open-ended evolution, where EA must operate continuously without any breaks [8]. Unfortunately, losing the population diversity normally leads to a premature convergence. A lot of approaches have been proposed for avoiding this phenomenon, such as, for instance by Črepinšek et al. [13], and by Fister et al. [5]. The novel step in mastering the arisen problem in open-ended evolution and artificial life, obviously, is the development of Parallel Evolutionary Algorithms (PEA) [7] and Evolutionary MultyAgent Systems (EMAS) [3]. PEAs decompose the monolithic population in EAs into more sub-populations, where

[*] Corresponding Author: iztok.fister@um.si

selection and reproduction are limited to individuals inhabiting one region and a migration operator is used to move selected individual from one region to another. On the other hand, agents in EMAS, representing solution of the problem to be solved, are distributed across islands (i.e., sub-populations) and can interact directly only with their local environment. Due to suffering a lack of global knowledge, their decision-making process is limited only on local information [3].

The purpose of this study is to confront with the problem of premature convergence in Parallel Differential Evolution (PDE) decomposing the monolithic population into more sub-populations (i.e., islands) capable of exploring different regions of the search space independently of each other. Moreover, individuals in the PDE have some characteristics of agents, because they act autonomously in deciding, which action to select in order to affect the state of environment (i.e., problem). Consequently, also some elements of the classical DE algorithm are either eliminated (e.g., survivor selection), or redefined in new way (e.g., variation operators).

PDE incorporates two new mechanisms: aging, and adaptive population growth. The former influences on the size of island by action death and controls decreasing the number of individuals with regard to the feedback obtained from the last generation, similar as the latter that directs individuals by deciding, whether to apply reproduction with growing of the population or clone actions keeping the population size intact. Because both mechanisms use global information, appearing on the population level, the individuals in the proposed PDE are not pure agents.

As a result, the proposed Parallel Variable sized DE for global optimization (gPVaDE) was developed and applied to the CEC-18 benchmark function suite. The results of comparison with the classical EAs, like DE [11], and its self-adaptive variants jDE [2] and SaDE [10], and the state-of-the-art algorithms, like LShade [12] and jSO [1], showed the potential of the proposed algorithm, and encourage us to continue with the research in the same direction also in the future.

2 Mutation strategies in Differential Evolution

DE was introduced by Storn and Price in 1995 [11] and work with real-valued vectors. Although these vectors in the original DE undergo operations of variation operators, such as mutation, crossover, and selection, we borrow only DE mutation strategies in our study. These strategies describes the way, in which the operations of mutation and crossover are conducted within DE.

In the basic mutation, two solutions are selected randomly, and their scaled difference is added to the third solution, as follows:

$$\mathbf{u}_i^{(t)} = \mathbf{x}_{r0}^{(t)} + F\cdot(\mathbf{x}_{r1}^{(t)} - \mathbf{x}_{r2}^{(t)}), \quad \text{for } i = 1, \ldots, Np, \tag{1}$$

where $F \in [0.1, 1.0]$ denotes the scaling factor that scales the rate of modification, while Np represents the population size and $r0,\ r1,\ r2$ are randomly selected values in the interval $1, \ldots, Np$.

The mentioned mutation strategy is capable for exploring a search space. When the exploitation of the search space is needed, the following mutation strategy is more appropriately:

$$\mathbf{u}_i^{(t)} = \mathbf{x}_{best}^{(t)} + F \cdot (\mathbf{x}_{r1}^{(t)} - \mathbf{x}_{r2}^{(t)}), \quad \text{for } i = 1, \ldots, Np, \tag{2}$$

where $\mathbf{x}_{best}^{(t)}$ is the current best individual, and $r1,\ r2$ are randomly selected values in the interval $1, \ldots, Np$. Let us emphasize that a balancing between exploration and exploitation can be achieved by mixing both strategies [13].

In our study, we employ a binomial crossover [11]. This crossover is uniform in the sense that each parameter, regardless of its location in the trial vector, has the same probability of inheriting its value from a given vector. Thus, the trial vector is built from parameter values copied from either the mutant vector generated by Eq. (1) or parent at the same index position laid i-th vector. Mathematically, this crossover can be expressed as follows:

$$w_{i,j}^{(t)} = \begin{cases} u_{i,j}^{(t)} & \text{rand}_j(0,1) \leq CR \vee j = j_{rand}, \\ x_{i,j}^{(t)} & \text{otherwise}, \end{cases} \tag{3}$$

where $CR \in [0.0, 1.0]$ controls the fraction of parameters that are copied to the trial solution. The condition $j = j_{rand}$ ensures that the trial vector differs from the original solution $\mathbf{x}_i^{(t)}$ in at least one element.

3 Proposed gPVaDE for global optimization

The proposed gPVaDE consists of more islands connected in ring topology that evolve in parallel. Each island hosts individuals located in some positions on a grid. They are capable of changing these positions randomly in each generation. On the other hand, these individuals can perform following autonomous actions: reproduction, clone, death, migration, and rebirth. Reproduction generates the trial solution using the exploration DE-mutation strategy (Eq. (1)), and keep it within the island beside its parent to evolve in the next generation. With new individual, the reproduction contribute to the island growth. Clone that generates the trial solution using the exploitation DE-mutation strategy (Eq. (2)) selects the better between parent and trial solutions for the next generation, and therefore keeps the island size unchanged. Death eliminate the individual from the island. Migration enable some individuals in one island to move in the other island according to a migration probability p_m. The action rebirth is launched, when the diversity in the island is lost.

Additionally, two mechanisms are incorporated within the algorithm, i.e., aging and adaptive population growth. The former introduces the concept of individual's age replacing the classical selection and changes this with more natural paradigm asserting, when people are old enough, they must die. The latter directs individual by making decision, whether replacement or clone should be performed in particular situation. In summary, the aging mechanism takes case about reducing the island size, while the adaptive population growth enriches the island with new individuals.

3.1 Aging mechanism

An aging mechanism presents one of the more popular concepts of adapting the population size during the evolutionary cycle in the EA community, and was used in the Genetic Algorithm with varying population size (GAVaPS) [9]. This mechanism introduced the concept of an individual's "age", which counts the number of generations the individual stays "alive".

The aging mechanism operates as follows: Each individual in a population lives the number of generations (ages) determined by its parameter *lifetime*. This parameter depends on the fitness of the corresponding individual, i.e., the higher the fitness of an individual, the higher a *lifetime* granted to it. Mathematically, the *lifetime* is defined as:

$$\mathit{lifetime} = \begin{cases} \mathit{MinLT} + K \cdot \frac{f_i - \mathit{MinFit}}{\mathit{AvgFit} - \mathit{MinFit}}, & \text{if } \mathit{AvgFit} \geq f_i, \\ \frac{1}{2}(\mathit{MinLT} + \mathit{MaxLT}) + K \cdot \frac{f_i - \mathit{AvgFit}}{\mathit{MaxFit} - \mathit{AvgFit}}, & \text{if } \mathit{AvgFit} < f_i, \end{cases} \tag{4}$$

where *MinLT* and *MaxLT* denotes the minimum and maximum available *lifetime* values, respectively, *AvgFit*, *MinFit*, and *MaxFit* are average, minimum, and maximum values of fitness in the current population, while the coefficient is expressed as $K = \frac{1}{2}(\mathit{MaxLT} - \mathit{MinLT})$.

3.2 Adaptive population growth

The adaptive population growth implements so-called Non-Linear population Size Reduction (NLSR) mechanism, where the population size is adapted following the population dynamics. In population dynamics, measure of the uncertainty in population size is expressed as:

$$\Delta H^{(t+1)} = \log \frac{2 \cdot S^{(t)}}{N^{(t+1)}}, \tag{5}$$

where ΔH denotes a change in the evolutionary entropy [4], $S^{(t)}$ is the number of positive variations in the last population, and $N^{(t+1)}$ is the effective population size in the next evolutionary cycle. Entropy influences increasing/decreasing the current population size regarding the following relations:

$$\Delta_{\max} = \begin{cases} -\text{rand}(1, 2 \cdot (R^{(t+1)} - N^{(t+1)})), & \text{if } \Delta H^{(t+1)} > 0, \\ +\text{rand}(1, 2 \cdot (R^{(t+1)} - N^{(t+1)})), & \text{if } \Delta H^{(t+1)} < 0, \\ 0, & \text{if } \Delta H^{(t+1)} = 0, \end{cases} \tag{6}$$

where $\Delta_{\max}$ denotes a modification in the population size, and $R^{(t+1)}$ is a decreased, linear, reference function that reduce the population size according to:

$$R^{(t+1)} = \left(1 - \frac{t+1}{t_{\max}}\right) \cdot (\mathit{MAX\_VAL} - \mathit{MIN\_VAL}) + (t+1), \tag{7}$$

where t is the generation number, $t_{\max}$ the maximum number of generation, *MAX_VAL* and *MIN_VAL* are the maximum and minimum population sizes, respectively.

3.3 Design and implementation of gPVaDE

Each individual in gPVaDE has its own time of birth and time of death, and lives as long as permitted by its quality of behavior. It is defined as a tuple:

$$X_i = \langle \mathbf{x}_i, f_i, M_i, age_i, lt_i, act_i \rangle, \quad \text{for } i = 1, \ldots, Np, \tag{8}$$

where $\mathbf{x}_i$ denotes vector with elements $\{x_{i,j}\}$ for $i = 1, \ldots, D$, f_i is a fitness function of the problem in question, M_i location of the individual within the island, age_i current age of agent, lt_i calculated lifetime, and act_i action to be performed by the individual.

Three main algorithms need to be implemented for covering the proposed three level program architecture of the gPVaDE, i.e.: algorithm's, island's, and individual's. The first algorithm takes care about: creation, termination and parallel evolving of islands, and synchronization between them. The second one provides global functions for individuals, like: positioning on random locations within grid, aging management, and adaptive population growth. The last one addresses tasks with which an individual is confronted, as: decision-making, and executing actions.

Due to the paper limitation, only the algorithm Evolve coping the second level is illustrated here (Algorithm 1). As can be seen, the Evolve algorithm launched in each generation is divided into two parts: In the first part, the evolving of the particular individuals is performed, and the potential emigrants are collected, while in the second, at first, the target island is selected between two island local neighbors in ring randomly and then migration is performed.

Algorithm 1 The Evolve algorithm.

```
procedure Evolve(island)
    emigrants = $\emptyset$;
    island.CalcLifeTimes;                                        ▷ update aging data
    $\mathbf{M}$ = island.GenerateTopology;              ▷ set new topology for individuals
    $\Delta_{\max}$ = island.EvolutionEntropy;        ▷ calculate the population dynamics
    for all individual $\in$ island do
        individual.Decision_Making(island);                   ▷ decision-making process
        individual.Do_Action(island);                             ▷ execution of actions
        emigrant = emigrant $\cup$ CollectEmigrants(island);        ▷ tag migration
    end for
    while emigrant $\neq \emptyset$ do                       ▷ performing migration actions
        emg_item = Head(emigrant);
        emigrant = Tail(emigrant);
        target_island =Rand(predecessor, successor);
        do_migrate(target_island, island,$emg_i tem$);              ▷ launch migration
    end while
end procedure
```

4 Experiments and results

The goal of this study can be condensed into three hypotheses, whose evidence can be shown that gPVaDE: (1) is capable of solving the global optimization problems introduced by the CEC-18 benchmark function suite, (2) can improve the behavior of classical linear population size reduction, and (3) achieves the results comparable with the results of some traditional EAs. During the experiments, the gPVaDE applied the parameter setup as illustrated in Table 1. Let us emphasize that the initial number of individuals resident on the island depends on the number of islands (normally $100/n$, where 100 is the maximum initial population size). On the other hand, all configurations use the same number of fitness function evaluations that serves also as a termination condition for the proposed algorithm, and $F = 0.5$ and $CR = 0.9$.

Table 1: Parameter setup of the proposed gPVaDE.

Parameter	Abbreviation	Value/Interval
Dimension of the problem	D	10
Island size	Np	$[10, 100]$
Number of islands	n	$[1, 10]$
Number of fitness function evaluations	$nFEs$	$1,000 \times D$
Probability of migration	p_m	0.001
Life time	*lifetime*	$[1, 24]$

The results obtained by the algorithms were evaluated according to five standard statistical measures: *Best, Worst, Mean, Median,* and *StDev* values. Friedman's non-parametric statistical test [6] was conducted in order to estimate the quality of the results obtained by various nature-inspired algorithms for global optimization. This test is a two-way analysis of variances by ranks, where the null hypothesis is stated assuming that medians between the ranks of all algorithms are equal.The second step is performed only if a null hypothesis of a Friedman test is rejected. In this step, the post-hoc tests are conducted using the calculated ranks. Indeed, a Wilcoxon two paired non-parametric test was applied in our study as a post-hoc test after determining the control method (i.e., the algorithm with the lowest rank) by using the Friedman test, while the Nemenyi post-hoc test is used for graphical presentation of the results. Both post-hoc tests were conducted using a significance level of 0.05.

The CEC'18 test suite consists of 30 benchmark functions that are divided into four classes:(1) unimodal functions (1–3), (2) simple multimodal functions (4–10), (3) hybrid functions (11–20), and (4) composition functions (21–30). Unimodal functions have a single global optimum and no local optima. Unimodal functions in this suite are non-separable and rotated. Multi-modal functions are either separable or non-separable. In addition, they are also rotated and/or shifted. To develop the hybrid functions, the variables are divided randomly into some sub-components and then different basic functions are used for different sub-components. Composition functions consist of a sum of two or more basic functions. In this suite, hybrid functions are used as the basic functions to construct composition functions. The characteristics of these hybrid and composition functions depend on the characteristics of the basic functions. The functions of dimensions D = 10 were used in our experiments due to a limitation of the paper length, while the search range of the problem variables was limited to $x_{i,j} \in [-100, 100]$.

4.1 Results

Influence of the adaptive population growth The purpose of the experiment was to establish a behavior of the adaptive population growth introducing the NLSR mechanism, and to compare this with the famous Linear population Size Reduction (LSR) used in many popular stochastic nature-inspired population-based algorithms. Actually, while the LSR is capable of uniform decreasing of population size with maturing the search process, the population size can also be increased by the NLSR depending on the number of positive individual's variations.

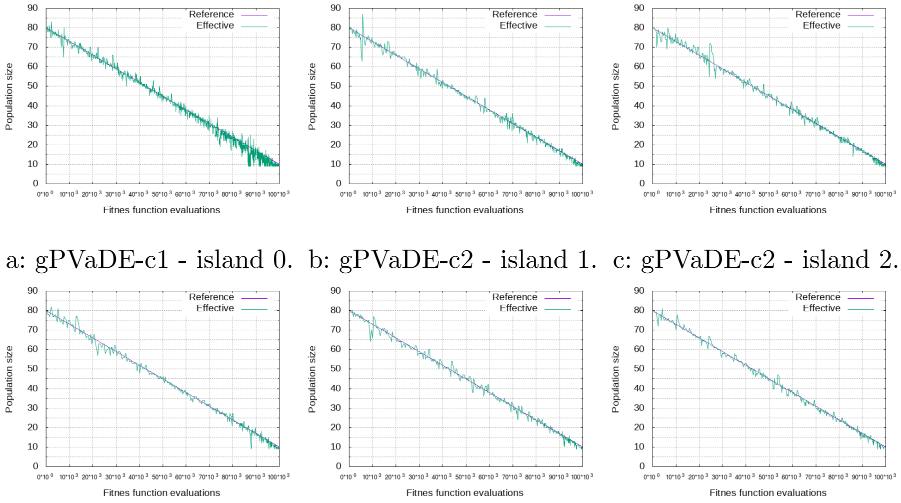
In order to show the advantage of NLSR, the six configurations of gPVaDE with the number of islands varying from one to six (denoted as gPVaDE-c1 to gPVaDE-c6) using the LSR, were compared with their counterparts using the NLSR. The results of the tests are depicted in Table 2, where the achievements obtained by gPVaDE using different features are compared in the sense of the Wilcoxon 2-paired non-parametric statistical test. From the table, it can be seen that, in summary, the gPVaDE using NLSR were significantly better than their counterparts using LSR even tree times, i.e., in most, especially, if we assume that monolithic population is not relevant.

Table 2: Comparison between two different models of population size reduction in various configuration of gPVaDE ($D = 10$).

Method	gPVaDE						Summary
	c1	c2	c3	c4	c5	c6	
LSR	+	−	−	−	+	=	2
NLSR	−	+	+	−	+	=	3

In the next test, the behavior of the NLSR feature was established, where multimodal function f_4 of dimension $D = 10$ was taken into consideration. The function was optimized by the proposed gPVaDE algorithm with three different gEMAS configurations, i.e., using one (gPVaDE-c1), using two (gPVaDE-c1), and using three islands (gPVaDE-c3). Actually, one typical run of the function optimization was selected for each algorithm in question, where no optimal solution was found. In line with this, the behavior of these can be monitored during the whole run. The result of the test is illustrated in Fig. 1 that is divided into six diagrams. As can be seen from Fig. 1, the population size oscillates around the reference line representing the traditional LSR. Typically, the population size increasing is followed by size reduction. The increasing is launched either by the decision-making process imposing by the reproduction, while reducing the population size is caused by the aging mechanism. Sometimes, this mechanism can introduce such a high selection pressure that can eliminate the major part of individuals from the island in only one cycle. Indeed, imposing the reproduction in the next cycle causes replacing the vacant places in the island. On the other hand, the diversity of island can be lost, when its size becomes lower. In this case, the new genetic material can be supplied using action rebirth.

Comparative analysis The goal of this test was to show that the results of the proposed gPVaDE algorithms are comparable with the results of the traditional



a: gPVaDE-c1 - island 0. b: gPVaDE-c2 - island 1. c: gPVaDE-c2 - island 2.

d: gPVaDE-c3 - island 1. e: gPVaDE-c3 - island 2. f: gPVaDE-c3 - island 3.

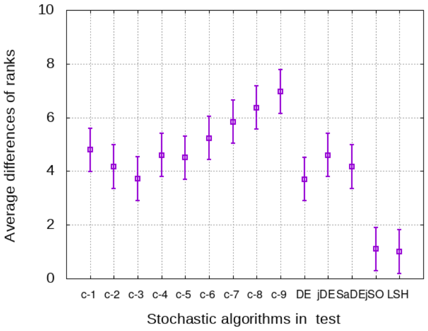
Fig. 1: Influence of the adaptive population growth on function f_4 of dimension $D = 10$.

stochastic nature-inspired population-based algorithms, like DE, jDE, and SaDE, although these do not achieve those obtained by the state-of-the-art algorithms, like jSO and LShade, at the moment. Thus, even nine configurations of gPVaDE algorithms were taken into consideration with varying the number of islands from one to nine, denoted as gPVaDE-c1 to gPVaDE-c9.

The results obtained by the particular algorithms were compared using the Friedman non-parametric statistical tests, and refined by a Nemenyi and Wilxocon post-hoc statistical tests. These are depicted in Fig. 2 that is divided into two parts, where the first presents the results in numerical and second in graphical form. As can be seen from the figure, the result quality of the proposed gPVaDE algorithms depends on the number of islands. It turns out that the smaller number of islands is better than the higher. However, the gPVaDE using monolithic population is distinguished as not the preferable configuration.

In summary, the more islands in an algorithm demand more small-sized populations. This is very inefficient for the search process due to suffering a lack of population diversity. On the other hand, the algorithm using a monolithic population maintains the higher population diversity, but suffers a lack of selection pressure. As a result, the proper bias between the population diversity and selection pressure ensure the optimal results for the configuration. In our case, the reasonable number of agents must be higher than or equal to two, but lower than or equal than six.

Algorithm	Fri.	Nemenyi		Wilcoxon	
		CD	S.	p-value	S.
gPVaDE-c1	4.83	[4.02,5.64]	†	> 0.05	†
gPVaDE-c2	4.26	[3.45,5.07]	†	> 0.05	†
gPVaDE-c3	3.77	[2.96,4.58]	†	> 0.05	†
gPVaDE-c4	4.42	[3.61,5.23]	†	> 0.05	†
gPVaDE-c5	4.69	[3.88,5.5]	†	> 0.05	†
gPVaDE-c6	5.42	[4.61,6.23]	†	> 0.05	†
gPVaDE-c7	5.43	[4.62,6.24]	†	$\gg 0.05$	†
gPVaDE-c8	6.61	[5.80,7.42]	†	$\gg 0.05$	†
gPVaDE-c9	6.71	[5.90,7.52]	†	$\gg 0.05$	†
DE	3.78	[2.97,4.59]	†	> 0.05	†
jDE	4.72	[3.91,5.53]	†	> 0.05	†
SaDE	4.15	[3.34,4.96]	†	> 0.05	†
jSO	1.12	[0.31,1.93]		$\ll 0.05$	
LShade	1.00	[0.19,1.81]	‡	∞	‡



a: $D = 10$, $p_m = 0.001$.

Fig. 2: The results of comparative analysis using Nemenyi post-hoc statistical test.

5 Conclusion

The question how to maintain a diversity of population has been followed researchers in evolutionary community from the same beginning. In the study, this problem is addressed by introducing the parallel gPVaDE algorithm, where the individuals are capable of executing some actions, with which they modify their programming environment and thus have more autonomy as those in the classical EAs. Additionally, the algorithm incorporates two mechanisms: aging, and adaptive population growth. While the former controls the action death, the latter navigates between actions reproduction and clone implemented as different DE mutation strategies.

The proposed gPVaDE was applied for solving the CEC'18 benchmark function suite representing a test-bed for global optimization. During huge experimental work, the influence of the new adaptive population growth mechanism was discovered in detail. Finally, the results produced by various gPVaDE configurations were compared with some traditional stochastic nature-inspired population-based algorithms, like DE, jDE, and SaDE, and some state-of-the-art algorithms, like jSO, and L-Shade. From the comparative analysis, it can be seen that the results of the proposed gPVaDE are comparable with those of the traditional algorithms in question, while needing some improvements to become more competitive also with the state-of-the-art ones.

In the future, we would like to build the gPVaDE with pure agents capable of decision-making at the individual's level. Applying the algorithm for solving the other hard optimization problems seem to be a very challenging for the future.

Acknowledgment

Iztok Fister thanks the financial support from the Slovenian Research Agency (Research Core Funding No. P2-0042 - Digital twin). Iztok Fister Jr. thanks the financial support from the Slovenian Research Agency (Research Core Funding No. P2-0057). Andres Iglesias and Akemi Galvez thank the Computer Science National Program of the Spanish Research Agency and European Funds, Project #TIN2017-89275-R. (AEI/FEDER, UE), and the PDE-GIR project of the European Union's Horizon 2020 programme, Marie Sklodowska-Curie Actions grant agreement #778035.

References

1. Brest, J., Maučec, M.S., Bošković, B.: Single objective real-parameter optimization: Algorithm jso. In: 2017 IEEE Congress on Evolutionary Computation (CEC). pp. 1311–1318 (June 2017). https://doi.org/10.1109/CEC.2017.7969456
2. Brest, J., Greiner, S., Bošković, B., Mernik, M., Žumer, V.: Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. IEEE Trans. Evolutionary Computation **10**(6), 646–657 (2006). https://doi.org/10.1109/TEVC.2006.872133
3. Byrski, A., Drezewski, R., Siwik, L., Kisiel-Dorohinicki, M.: Evolutionary multi-agent systems. Knowledge Eng. Review **30**(2), 171–186 (2015). https://doi.org/10.1017/S0269888914000289
4. Demetrius, L., Legendre, S., Harremöes, P.: Evolutionary entropy: A predictor of body size, metabolic rate and maximal life span. Bulletin of Mathematical Biology **71**(4), 800–818 (May 2009). https://doi.org/10.1007/s11538-008-9382-6
5. Fister, I., Iglesias, A., Galvez, A., Del Ser, J., Osaba, E., Fister Jr., I., Perc, M., Slavinec, M.: Novelty search for global optimization. Applied Mathematics and Computation **347**, 865–881 (2019)
6. Friedman, M.: A comparison of alternative tests of significance for the problem of m rankings. Ann. Math. Statist. **11**(1), 86–92 (03 1940). https://doi.org/10.1214/aoms/1177731944
7. Luque, G., Alba, E.: Parallel Genetic Algorithms: Theory and Real World Applications. Springer Publishing Company, Incorporated (2013)
8. Lynch, M.: The evolution of genetic networks by non-adaptive processes. Nature Reviews Genetics **8**, 803–813 (2007). https://doi.org/10.1038/nrg2192
9. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer Verlag, Berlin, 2st edn. (1996)
10. Qin, A.K., Suganthan, P.N.: Self-adaptive differential evolution algorithm for numerical optimization. In: 2005 IEEE Congress on Evolutionary Computation. vol. 2, pp. 1785–1791 (Sep 2005). https://doi.org/10.1109/CEC.2005.1554904
11. Storn, R., Price, K.: Differential evolution &ndash; a simple and efficient heuristic for global optimization over continuous spaces. J. of Global Optimization **11**(4), 341–359 (Dec 1997). https://doi.org/10.1023/A:1008202821328
12. Tanabe, R., Fukunaga, A.S.: Improving the search performance of shade using linear population size reduction. In: 2014 IEEE Congress on Evolutionary Computation (CEC). pp. 1658–1665 (July 2014). https://doi.org/10.1109/CEC.2014.6900380
13. Črepinšek, M., Liu, S.H., Mernik, M.: Exploration and exploitation in evolutionary algorithms: A survey. ACM Comput. Surv. **45**(3), 35:1–35:33 (Jul 2013). https://doi.org/10.1145/2480741.2480752