VOLUME 18, N° 1 2024
Journal of Automation, Mobile Robotics and Intelligent Systems

$(\mu + \lambda)$ Evolution Strategy with Socio-cognitive Mutation

Submitted: 8th February 2023; accepted: 17th July 2023

Aleksandra Urbanczyk, Krzysztof Kucaba, Mateusz Wojtulewicz, Marek Kisiel-Dorohinicki, Leszek Rutkowski, Piotr Duda, Janusz Kacprzyk, Xin Yao, Siang Yew Chong, Aleksander Byrski

DOI: 10.14313/JAMRIS/1-2024/1

Abstract:
Socio-cognitive computing is a paradigm developed for the last several years in our research group. It consists of introducing mechanisms inspired by inter-individual learning and cognition into metaheuristics. Different versions of the paradigm have been successfully applied in hybridizing Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Genetic Algorithms, Differential Evolution, and Evolutionary Multi-agent System (EMAS) metaheuristics. In this paper, we have followed our previous experiences in order to propose a novel mutation based on socio-cognitive mechanism and test it based on Evolution Strategy (ES). The newly constructed versions were applied to popular benchmarks and compared with their reference versions.

Keywords: *metaheuristics, socio-cognitive computing, global optimization*

1. Introduction

Tackling difficult optimization problems requires using metaheuristics [1], and very often it is needed to create new ones [2], i.e. by modifying or hybridizing the existing algorithms [3].

Although Sorensen has criticized the development of new metaheuristics [4], we contend that using metaphors in our daily work [5] not only fosters creativity but also may result in the discovery of truly new solutions of considered issues or novel mechanisms to solve them automatically.

Because classic metaheuritics are frequently inspired by nature, their further modifications frequently combine different phenomena observed in the real world.

One direction of such modifications comes from the very influential Social-Cognitive Theory introduced by Bandura [6]. According to this theory, some of a person's knowledge can be directly linked to observing others during their social interactions, experiences, and external media influences. [7]. Thus, despite learning only through her own trial and error, one can reach her goals sooner thanks to such observation [8].

We have already introduced dedicated mechanisms rooted in Social-Cognitive Theory to selected metaheuristics (socio-cognitive ACO [9] and socio-cognitive PSO [10]), obtaining good results compared to the reference algorithms.

Presently, we focus on the group of evolutionary metaheuristics, and by modifying chosen algorithms from this group, we aim to develop a universal mechanism for variation operators that would embody the idea of socio-cognitive learning mechanisms.

The main contribution of this paper is a socio-cognitively inspired mutation mechanism, that makes it possible to exchange the information among the individuals in evolutionary algorithms. The proof-of-concept of this mechanism was introduced in the research paper in 2021 [11] and was redesigned and reimplemented based on the results achieved. The efficiency and efficacy of the new version of the algorithms are tested using well-known high-dimensional, multimodal benchmark functions. The proposed method is based on copying certain parts of the genotypes (thus passing the knowledge) from the better ones, and avoiding the parts of solutions of the worst ones. In this paper, we consider well-known $(\mu + \lambda)$ ES, but we believe that our mutation mechanism may be used in a broader range of algorithms.

We start with the reference to state-of-the-art showing the existing modifications of metaheuristics, in particular evolution strategies. Then we show the novel method for introducing socio-cognitive mechanisms into $(\mu + \lambda)$ evolution strategy. We provide relevant experimental results and, in the end, we conclude our paper showing the summary and the future work plans.

2. Related Non-classic Evolutionary Algorithms

There are several metaheuristic discourses in which this work can be anchored. On the most general level (considering the architecture of the entire algorithm), it can be treated as a kind of hybrid algorithm [12] in the same sense that a memetic algorithm is one [13] and many other similar algorithms, developed in the research group of the Authors [14–16]. The majority of memetic algorithms are based on genetic algorithm, and have introduced some local search or heuristic learning mechanisms. Unlike them, the described algorithm is based on another metaheuristic of the evolutionary computation group, namely the evolution strategy [17, 18].

The similarity lies in the fact that a novel mechanism (i.e., socio-cognitive mutation operator) is introduced in between standard steps of the algorithm. Our work should also be placed in the context of various modified or hybrid ESs. The possible modifications of classic ESs range from simple tuning or manipulation of control parameters such as mutation strength or population size (step-size) [19–21], through covariance matrix adaptation evolution strategy (CMA-ES) [22] to heterogeneous hybrids of ES, which are often focused on particular application, e.g. vehicle routing problem [23], optimization of engineering, and construction problems [24, 25] and the number of which is apparently not very high.

Taking into account the level of the variation operators itself, our postulated operator can be compared to the one present in the differential evolution metaheuristic [26]. The characteristic trait of DE is the mutation variation operator, which operates on parameter vectors with scaled population-derived difference vectors. In this sense, it is not just a randomly performing operator, as in traditional EAs and ESs, but it utilizes the information about current population, especially in the schemes having "best" in the names, such as $DE/best/1$ and $DE/target - to - best/1$ that use the best solution to define mutation directions [27]. A similar analogy is present between classic mutation and our socio-cognitive mutation operator. The mechanics of the new operator can be related to the well-known TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) method [28]. TOPSIS is based on the idea that the chosen alternative should be the one with the shortest geometric distance from the positive ideal solution and the one with the greatest geometric distance to the negative ideal solution.

As already mentioned in the Introduction, we root our work in a discourse of socio-cognitively inspired algorithms. The first objective of introducing sociocognitive mechanism into evolution strategies served as a proof-of-concept that turned out to be promising [11], but pointed out several dimensions for major improvements. The first conclusion was that these mechanisms that operate towards better solutions give better results than operators based on moving away from the worst individuals. We decided that the core of our idea was a synergy of these two directions, and that the second part must be totally redesigned in order to work as intended. Otherwise, it would be too straightforward analogy with $DE/best/1$ and other socio-cognitive algorithms described in [29] and [30], so the novelty would be minimal. The second lesson from the previous attempt to modify ES was that the algorithm itself should have a moderate level of complexity in order to be a base for a successful socio-cognitive modification. The experiments performed on the $(1 + 1)$ version of ES, as well as the (μ, λ) version were not as successful as those based on the $(\mu + \lambda)$ version of the algorithm, which gave better results in all the benchmarks tested, in contradiction to the (μ, λ) version that was better only in one of them. So we decided that it will be the best to stick to the $(\mu + \lambda)$ version for our further purposes.

3. Socio-cognitive $(\mu + \lambda)$ Evolution Strategy

The classic algorithm of ES can be described as follows:

1) Initialize parent population $P_\mu = \{i_1, \dots, i_\mu\}$. Each of the individuals can be described as follows: $I \ni i_k = \{g_{k,1}, \dots, g_{k,d}, s_{k,1}, \dots, s_{k,d}\}, k, d \in \mathbb{N}$ stands for an individual containing a genotype $g_{k,1}, \dots, g_{k,d}$ representing objective parameters, and associated $s_{k,1}, \dots, s_{k,d}$ mutation strategy parameters that will be adapted in order to guide the search. The dimensionality of the considered problem is d. **Later, we use the notation $i_{k,l}$ to refer to $g_{k,l}$, which is l-th gene of k-th genotype.**
2) Generate λ offspring individuals forming the offspring population $P_\lambda = \{i_1, \dots, i_\lambda\}$ in the following procedure:
   - Randomly select ϱ parents from P_μ (if $\varrho = \mu$, then take all of them).
   - Recombine the ϱ selected parents (traditionally a pair) to form a recombinant individual i_r, using any possible recombination means (traditionally averaging crossover operator was used).
   - Mutate the strategy parameter set $s_{r,1}, \dots, s_{r,d}$ of the recombinant i_r (adapting e.g. the mutation diversities for the next mutation). Traditionally, mutation is realized by applying a perturbation based on, for example uniform or Gaussian random distribution or adding or subtracting a certain value to (from) a selected gene.
   - Mutate the objective parameter set $g_{r,1}, \dots, g_{r,d}$ of the recombinant i_r using the mutated strategy parameter set to control the statistical properties of the object parameter mutation.
3) Select new parent population (using deterministic truncation selection) from either the offspring population P_λ (this is referred to as comma-selection, usually denoted as "(μ, λ)-selection"), or the offspring P_λ and parent P_μ population (this is referred to as plus-selection, usually denoted as "$(\mu + \lambda)$-selection").
4) Go to 2. until termination criterion fulfilled.

We have decided to introduce the socio-cognitive mechanisms to the $(\mu + \lambda)$ version of ES. This follows from the apparent potential of such mechanisms developed earlier in [11]. We have studied the updating part of the operators applied therein, and introduced modifications in order to increase their efficacy.

In particular, we have aimed at increasing the exchange rate of information between the individuals in current population with the goal of accelerating the learning rate of algorithm. In order to achieve this, we split a single mutation step into multiple independent sequential mutations. The first mutation is always the classical operator meant to introduce perturbation to the solution's genome. The following operator or multiple operators are meant to introduce further modifications to that solution that are guided by the current state of population.

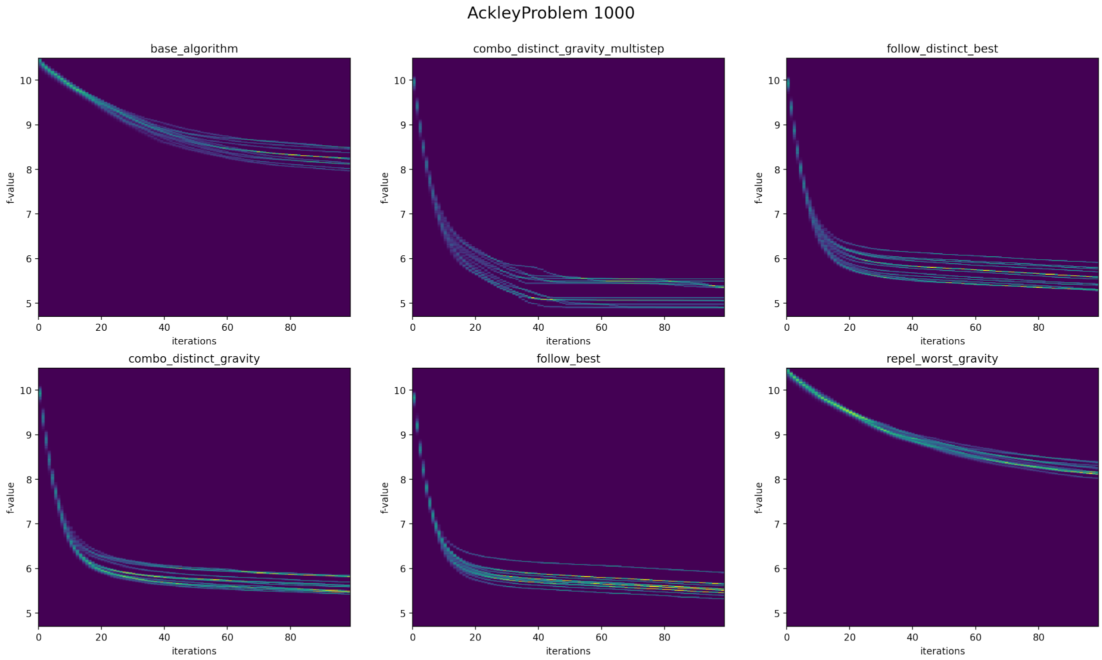

Figure 1. Population trajectory for each algorithm on AckleyProblem 1000. Each vertical slice at given step represents histogram of joined populations over all evaluation runs with color depicting histogram box count

In our experiments we test and evaluate the following social mutations:

1) Follow Best:
   Out of the top n individuals $B_1, \ldots, B_n$ in current population randomly select one that will be now called teacher T. With probability p_f, for each of the currently operated on solution's S genes s_i, assign new value $s_i \leftarrow s_i + \alpha_f(t_i - s_i)$ where t_i is the corresponding gene of T and α_f is follow rate.

2) Follow Best Distinct:
   Let each individual B_j be a sequence of d genes $B_j = (B_{j,1}, \ldots, B_{j,d})$. Out of the top n individuals $B_1, \ldots, B_n$ in current population randomly select one that will be now called teacher T. Across the $B_1, \ldots, B_n$ individuals calculate the standard deviation for each of the gene positions $1, \ldots, d$ resulting in $g^1_{std}, \ldots, g^d_{std}$ where $g^i_{std} = std(B_{1,i}, \ldots, B_{n,i})$. Choose k gene positions performing weighted random selection across $1, \ldots, d$ using $softmax(g^1_{std}, \ldots, g^d_{std})$ as vector of probabilities. For each of k chosen gene positions of the currently operated on solution's S genes s_i assign new value $s_i \leftarrow s_i + \alpha_f(t_i - s_i)$ where t_i is the corresponding gene of T and α_f is follow rate.

3) Repel Worst Gravity:
   Out of n worst individuals in the current population randomly select one individual Bi_w. While operating on an individual Bi_m, with probability p_f, perform the following assignment for every gene g: $i_{m,g} \leftarrow i_{m,g} + \alpha_r \cdot \frac{sgn(d_g)}{d_g^2}$, where $d_g = (i_{m,g} - i_{w,g})$ is called a distance in gene g, sgn is a sign function and α_r is a repel rate. That way the repel magnitude is inversely proportional to the squared distance for a given gene, and with a direction away from the chosen worst individual.

4) Repel Worst Gravity Multistep:
   For every individual B_w from n worst individuals in the current population perform the assignments described above. That way the repel effect is stronger and more versatile.

4. Experiments

The main aim of the experiments is to verify the efficacy of global optimization (minimization) of the novel algorithms for the selected benchmark functions (Ackley, De Jong, Rastrigin, and Griewank [31]) of dimensions $d \in \{100, 500, 1000\}$. Both the value obtained in the last iteration, and the trajectory of the fitness functions improvements are considered – in certain situations it is desirable to have a relatively fast convergence earlier, in other situations the focus is placed on the final result. The equations used for the benchmark functions are as follows:

- Ackley: $f(x) = -ae^{-b\sqrt{1/n\sum_{i=1}^{n}(x_i^2)}} - e^{1/n\sum_{i=1}^{n}\cos(cx_i)} + a + e; a = 20; b = 0.2; c = 2\pi; i \in [1:n]; -32.768 \le x(i) \le 32.768$. $f(x^{\text{opt}}) = 0$, $x_i^{\text{opt}} = 0$.
- De Jong: $f(x) = \sum_{i=1}^{n} x_i^2, i \in [1, n]; -5.12 \le x_i \le 5.12$. $f(x^{\text{opt}}) = 0$, $x_i^{\text{opt}} = 0$.
- Rastrigin: $f(x) = 10n + \sum_{i=1}^{n}(x_i^2 - 10\cos(2\pi x_i)), i \in [1, n]; -5.12 \le x_i \le 5.12$. $f(x^{\text{opt}}) = 0$, $x_i^{\text{opt}} = 0$.
- Griewank: $f(x) = \sum_{x=1}^{n} x_i^2/4000 - \prod\cos(x_i/\sqrt{i}) + 1, i \in [1, n]; -600 \le x_i \le 600$, $f(x^{\text{opt}}) = 0$, $x_i^{\text{opt}} = 0$.

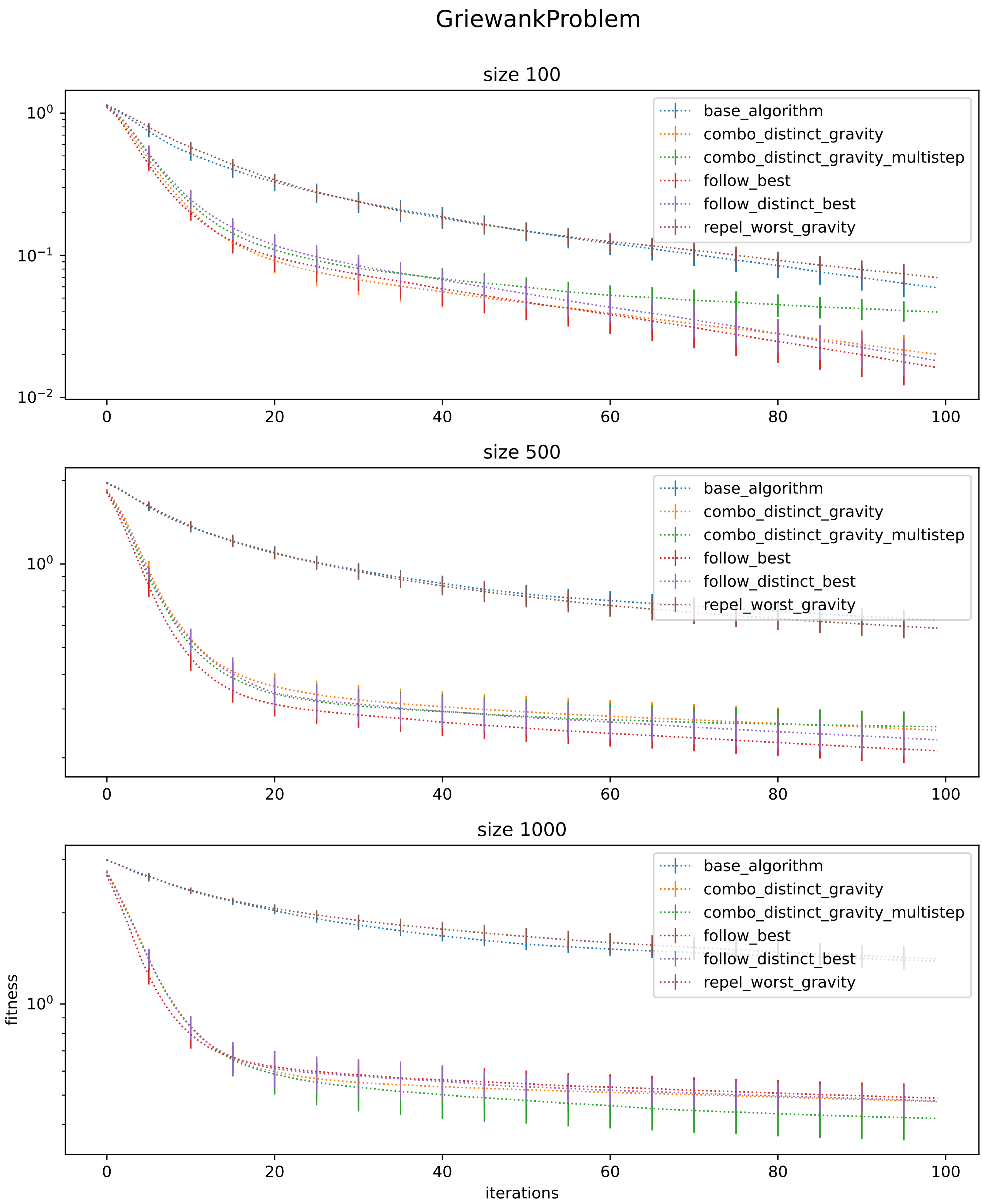

Figure 2. Trajectory of changes of mean fitness function value for Griewank problem and $(\mu + \lambda)$ Evolutionary Strategy, depending on the number of iterations

The following algorithms have been benchmarked:

- Original $(\mu + \lambda)$ ES,
- Follow Best ES – with the Follow Best mutation,
- Follow Best Distinct ES – with the Follow Best Distinct mutation,
- Repel Worst Gravity Multistep ES – with the Repel Worst Gravity Multistep mutation,
- Combo Distinct Gravity ES – with the Follow Best Distinct and Repel Worst Gravity mutations,
- Combo Distinct Gravity Multistep ES – with the Follow Best Distinct and Repel Worst Gravity Multistep mutations.

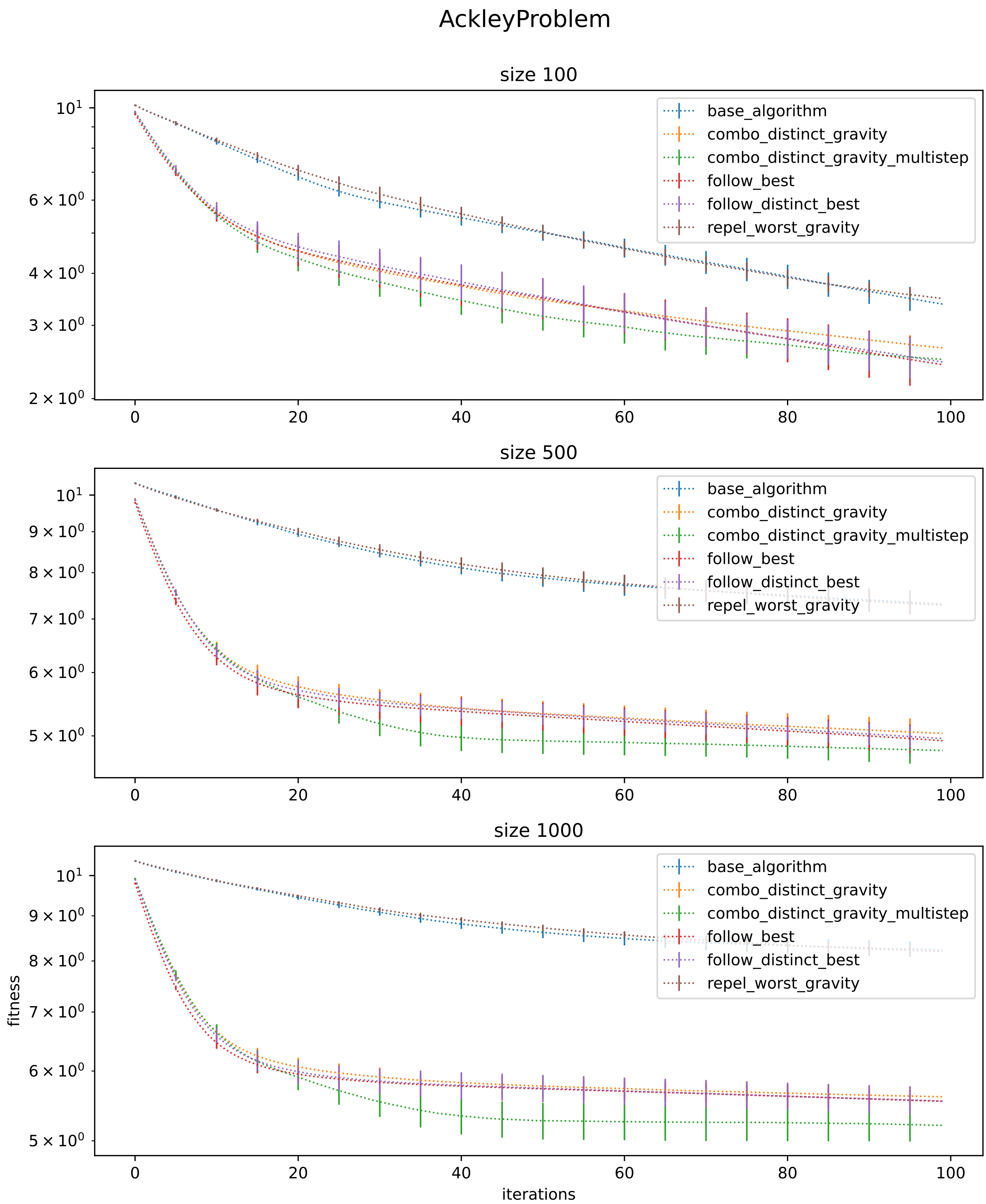

Figure 3. Trajectory of changes of mean fitness function value for Ackley problem and $(\mu + \lambda)$ Evolutionary Strategy, depending on the number of iterations

The stopping criteria was reaching maximum number of iterations of population updates (set as 100 for all the experiments). The number of individuals in the population was set to $\mu = 200$. The following settings have been used for the algorithms:

- $\mu = 20, \lambda = 140$.
- $\alpha_{\text{good}} = 0.1, \alpha_{\text{bad}} = 0.1, \beta = 0.01$.
- $\gamma = 1/d$, where d is the number of dimensions,
- number of the currently best or worst individuals: 5.

Each experiment has been repeated 12 times, and the mean value of the fitness function is taken as reference. The algorithms have been implemented using jMetalPy[1] computing framework. The source code is available on request. The computations have been conducted on a PC-class computer.

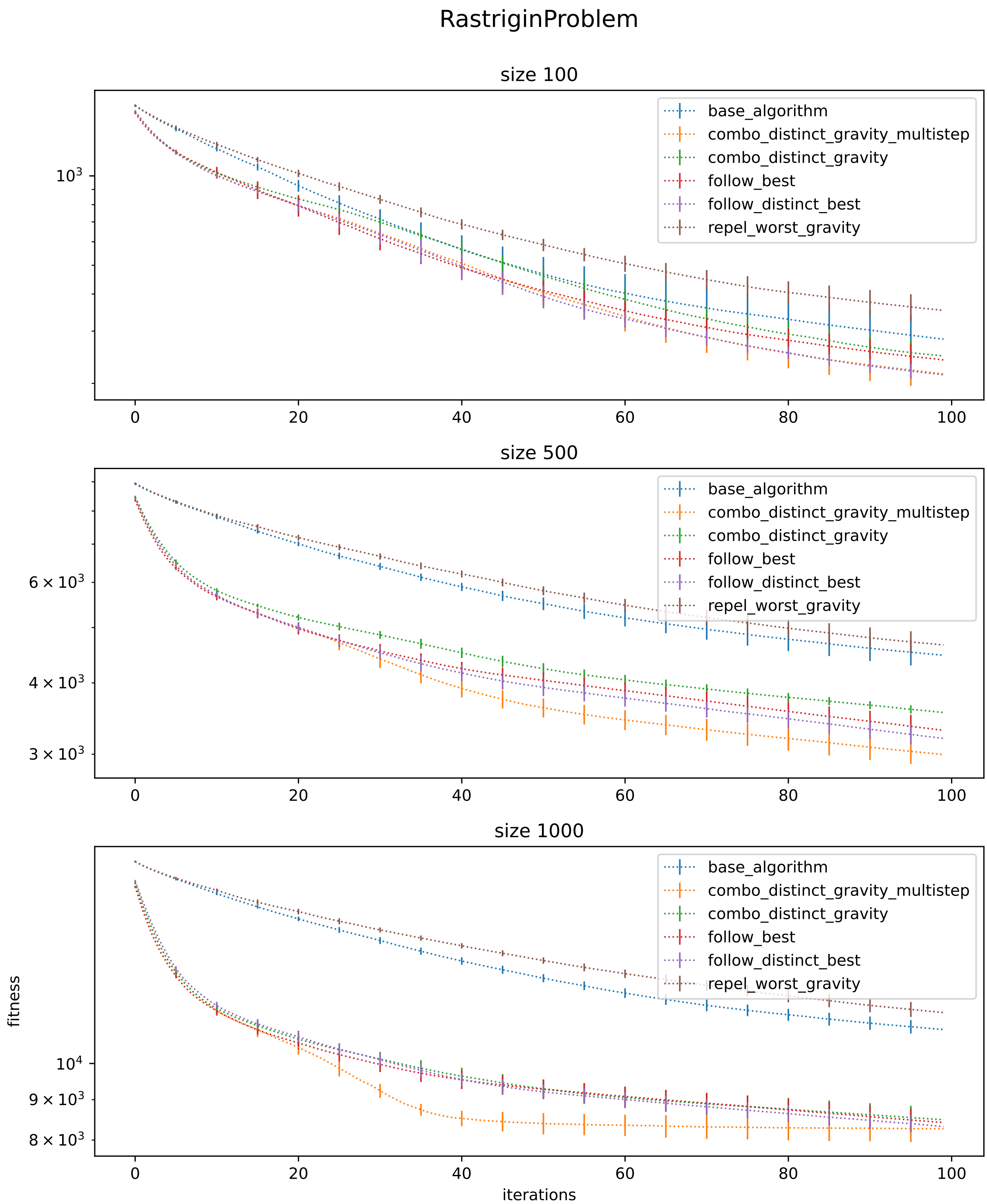

Figure 4. Trajectory of changes of mean fitness function value for Rastrigin problem and $(\mu + \lambda)$ Evolutionary Strategy, depending on the number of iterations

We start with observations of general behavior and on the repeatability (i.e., consistency of performance in repeated runs) of the algorithms when solving the problems for all the variants of the proposed algorithms. Therefore, we have prepared histogram-like visualizations of the computation runs. In Fig. 1, the actual trajectories of each algorithms can be seen. Moreover, each vertical slice shows the count of the values obtained at each iteration of the algorithm for all repeated experiments.

We can clearly see that all the variants of the modified $(\mu + \lambda)$ approaches are repeatable. Moreover, the results obtained for one of biggest problems tackled, namely Ackley in 1000 dimensions can also be observed in detail. Being convinced of the repeatability of the experiments we can proceed with subsequent phases of our studies.

Now we can focus on observations of the averages obtained for all the benchmark problems addressed with different configurations of the algorithms.

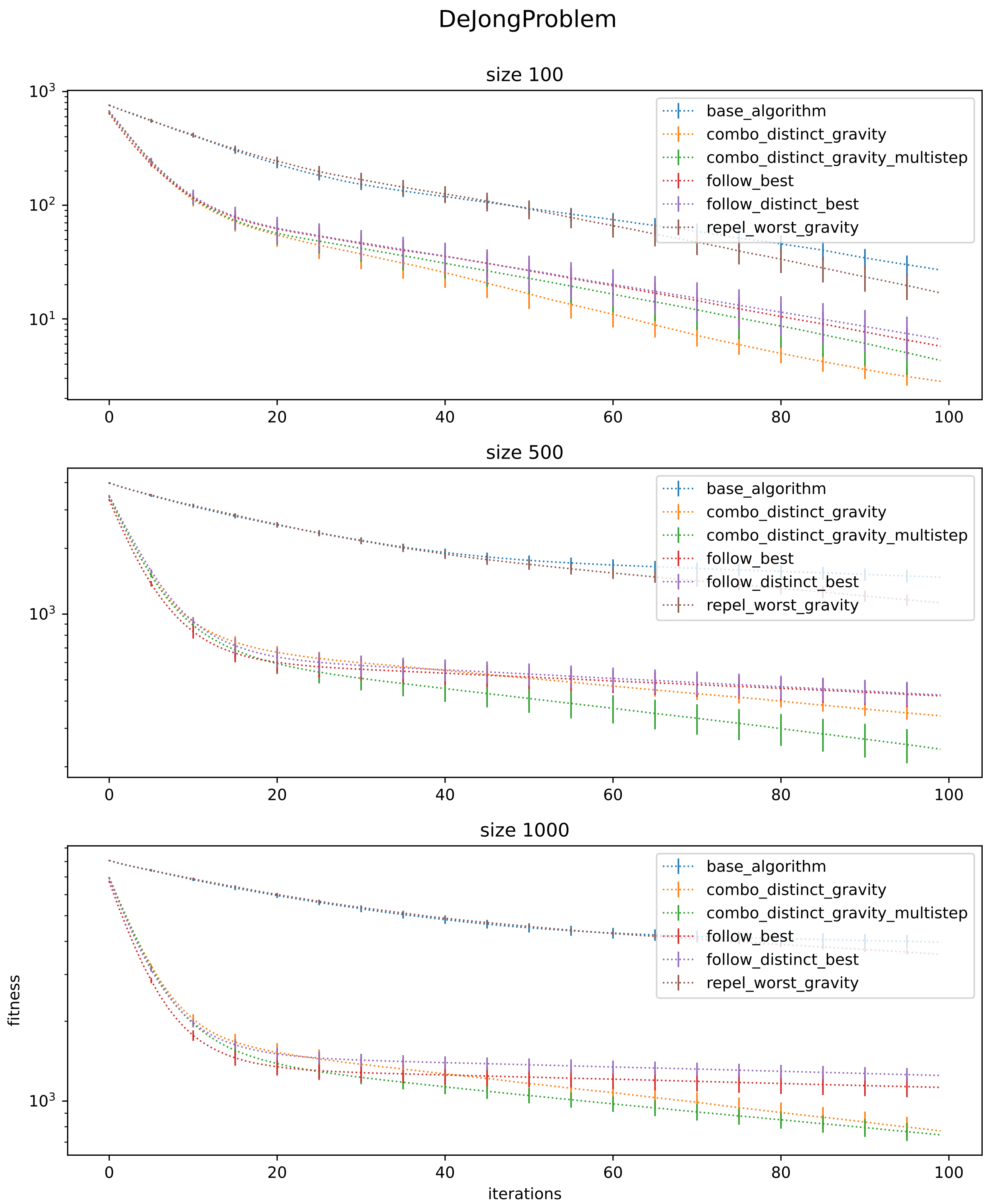

Figure 5. Trajectory of changes of mean fitness function value for DeJong problem and $(\mu + \lambda)$ Evolutionary Strategy, depending on the number of iterations

It is clear from observations of the results that our methods (including the base algorithm) are very effective in the case of Griewank and Ackley (see Figs. 2 and 3) problems. Not all our proposed methods are effective for De Jong and Rastrigin problem (see Figs. 5 and 4). **For example, the repel worst gravity approach does not always lead to improvements in the performance over the base algorithm**. This is not surprising following the main implication of the well-known *No Free Lunch Theorem* by Wolpert and MacReady [2], in which one of the important steps would be to optimize the parameters of the search for each individual problem.

Table 1. Mean and standard deviation of fitness value after 100 iterations of $(\mu+\lambda)$ ES and its hybrids for 100, 500 and 1000 dim. problems

Dimension	**100**		**500**		**1000**	
	Mean	**Std.**	**Mean**	**Std.**	**Mean**	**Std.**
Ackley						
Base Algorithm	3.37	0.23	7.31	0.26	8.23	0.17
Repel Worst Gravity	3.48	0.13	7.28	0.23	8.20	0.11
Follow Best	**2.41**	**0.33**	4.93	0.20	5.54	0.15
Follow Distinct Best	2.45	0.28	4.96	0.15	5.55	0.20
Combo Distinct Gravity	2.65	0.14	5.04	0.20	5.61	0.15
Combo Distinct Gravity Multistep	2.48	0.24	**4.79**	**0.20**	**5.21**	**0.22**
De Jong						
Base Algorithm	26.98	5.57	1474.59	98.64	3977.17	228.07
Repel Worst Gravity	16.93	4.37	1125.72	65.51	3578.57	77.48
Follow Best	5.76	1.63	422.32	53.79	1126.23	99.93
Follow Distinct Best	6.63	2.71	426.64	54.11	1249.48	71.77
Combo Distinct Gravity	**2.83**	**0.49**	342.40	24.51	772.03	71.87
Combo Distinct Gravity Multistep	4.31	1.60	**240.77**	**44.30**	**745.86**	**59.78**
Griewank						
Base Algorithm	0.059	0.012	0.63	0.05	1.38	0.07
Repel Worst Gravity	0.070	0.012	0.59	0.06	1.41	0.12
Follow Best	**0.016**	**0.005**	**0.21**	**0.02**	0.49	0.05
Follow Distinct Best	0.018	0.005	0.23	0.03	0.48	0.05
Combo Distinct Gravity	0.020	0.006	0.25	0.04	0.48	0.07
Combo Distinct Gravity Multistep	0.040	0.007	0.26	0.03	**0.42**	**0.07**
Rastrigin						
Base Algorithm	281.77	47.09	4471.16	235.14	11040.39	217.55
Repel Worst Gravity	352.57	37.46	4660.73	213.86	11599.90	264.36
Follow Best	239.88	26.33	3303.21	161.68	8418.42	277.27
Follow Distinct Best	**213.75**	**12.47**	3197.04	126.73	8320.57	210.38
Combo Distinct Gravity	247.19	27.88	3549.07	62.30	8488.31	296.62
Combo Distinct Gravity Multistep	215.04	24.96	**2994.79**	**149.64**	**8269.27**	**322.02**

Our motivation for this study is to test the efficiency and efficacy of our proposed mechanisms in their baseline configurations. As such, we have sought to determine their general capabilities to improve the reference ES algorithm over the whole set of selected benchmark problems.

When a particular mechanism did not lead to improvement but lead to lower average performance for a particular benchmark problem, results indicate that the difference is not statistically significant (e.g., Table 2 for Repel Worst Gravity compared with the base or reference ES algorithm) on the Griewank Problem at $d = 1000$. This suggests scope to optimize the parameter configurations of our proposed mechanisms that warrant further, future studies. In addition to a systematic parameter sweep to ascertain optimal parameter configurations for the mechanisms, other approaches would be to apply some dedicated algorithm tuning method such as iRace [32]. One additional conclusion of this phase is that the best of our modification was Combo Dist Gravity along with Repel Best.

In addition to providing qualitative descriptions of the behaviour of the algorithms is solving the benchmark problems using graphs, we corroborate those findings with quantitative results (e.g., average with standard deviation) that are presented in a tabular form.

Table 2. Dunn test p-values of algorithm pairs that exceeded the 0.01 threshold and are considered not significantly different

Problem	**Algorithms**	**p-value**
AckleyProblem 100	Follow Distinct Best Follow Best	0.76
AckleyProblem 1000	Repel Worst Gravity Base Algorithm	0.26
AckleyProblem 1000	Follow Distinct Best Follow Best	0.26
DejongProblem 500	Follow Distinct Best Follow Best	0.087
GriewankProblem 1000	Follow Distinct Best Follow Best	0.022
GriewankProblem 1000	Repel Worst Gravity Base Algorithm	0.022

These results are provided in Table 1. The observations confirm the findings perceived when analyzing the graphs, and the information obtained from the spread of results when the individual algorithms are repeated via standard deviation further convinces us about the repeatability of those algorithms and significance of the findings.

We have systematically performed various statistical testing on the quantitative results we have obtained. First, we have applied the Shapiro-Wilk test with significance threshold of 0.05 to check whether the observed sample had a normal distribution. **The null hypothesis that the sample obtained for each proposed algorithm is rejected**. As such, we proceed with the Kruskal-Wallis test in order to check whether their cumulative distribution functions differed, and finally pairwise comparisons via Dunn's test in order to check which ones were significantly different. Except for the results listed in Table 2, all other algorithms achieved statistically significant values with p-values below 0.01 (assuming this value as significance level α) using Dunn's test.

5. Conclusion

In this paper, we proposed and studied novel methods for hybridizing socio-cognitive inspirations in ES. The proposed algorithms are based on the principle of introducing certain mechanisms of attracting the currently modified genotypes to the best ones and repelling them from the worst ones in the population.

Our experiments yielded interesting results. It turns out that the proposed mechanisms were apparently successful for two of four tackled Benchmark problems (Ackley and Griewank) in all the dimensions tested. We verified this claim through both qualitative analysis via plots of the search performances of the algorithms and quantitative analysis via the use of systematic statistical analysis on the samples of search performances from repeated runs of the algorithms. However, the socio-cognitive mutation was successful for the two other problems, namely De Jong and Rastrigin, only in the case of 100 dimensions. It should be noted that we did not perform individual tuning of the parameters so as to obtain improvements. Our current motivation is to establish the generality of the proposed mechanisms as they are in baseline configuration.

Nevertheless, we showed that different variants of our methods succeeded – therefore following the well-known *No Free Lunch* theorem by Wolpert and MacReady, in our future research we would like to tune our methods to meet particular needs of all the tackled problem. Moreover, we will study if our modification of the base algorithm (in this case, ES) will work as well when applied in other metaheuristics, as the modification itself can be perceived as general one, not particularly connected with ES that is studied in this paper.

Notes

[1]https://github.com/jMetal/jMetalPy

AUTHORS

Aleksandra Urbanczyk – AGH University, Al. Mickiewicza 30, 30-059 Krakow, e-mail: aurbanczyk@agh.edu.pl.

Krzysztof Kucaba – AGH University, Al. Mickiewicza 30, 30-059 Krakow, e-mail: kkcba98@gmail.com.

Mateusz Wojtulewicz – AGH University, Al. Mickiewicza 30, 30-059 Krakow, e-mail: mateusz.wojtulewicz@gmail.com.

Marek Kisiel-Dorohinicki – AGH University, Al. Mickiewicza 30, 30-059 Krakow, e-mail: doroh@agh.edu.pl.

Leszek Rutkowski – Institute of Systems Science Research, Warsaw, Poland; AGH University, Al. Mickiewicza 30, 30-059 Krakow, e-mail: rutkowski@agh.edu.pl.

Piotr Duda – Czestochowa University of Technology, Poland, e-mail: piotr.duda@pcz.pl.

Janusz Kacprzyk – Institute of Systems Science Research, Warsaw, Poland; AGH University, Al. Mickiewicza 30, 30-059 Krakow, e-mail: janusz.kacprzyk@ibspan.waw.pl.

Xin Yao – Southern University of Science and Technology, Shenzhen, China, e-mail: xiny@sustech.edu.cn.

Siang Yew Chong – Southern University of Science and Technology, Shenzhen, China, e-mail: chongsy@sustech.edu.cn.

Aleksander Byrski* – AGH University, Al. Mickiewicza 30, 30-059 Krakow, Poland, e-mail: olekb@agh.edu.pl, www: https://orcid.org/0000-0001-6317-7012.

*Corresponding author

ACKNOWLEDGEMENTS

The research presented in this paper received support from the Polish National Science Centre project no. 2019/35/O/ST6/00570 (AU), the funds assigned by Polish Ministry of Education and Science to AGH University (MKD, JK), by the program "Excellence initiative research university" for the AGH University in Krakow, the ARTIQ project: UMO-2021/01/2/ST6/00004 and 84 ARTIQ/0004/2021 (LR) and NCN Project no. 2020/39/I/ST7/02285 (AB).

References

[1] Z. Michalewicz and D. B. Fogel, *How to Solve It: Modern Heuristics*. Springer Science & Business Media, 2013.

[2] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, Apr. 1997, pp. 67–82, doi: 10.1109/4235.585893.

[3] E.-G. Talbi, *Metaheuristics: From Design to Implementation*. John Wiley & Sons, 2009.

[4] K. Sörensen, "Metaheuristics—the metaphor exposed," *International Transactions in Operational Research*, vol. 22, no. 1, 2015, pp. 3–18, doi: 10.1111/itor.12001.

[5] G. Lakoff and M. Johnson, *Metaphors We Live By*. Chicago, IL: University of Chicago Press, 2003. Accessed: Feb. 16, 2024. [Online]. Available: https://press.uchicago.edu/ucp/books/book/chicago/M/bo3637992.html.

[6] A. Bandura, "Self-efficacy: Toward a unifying theory of behavioral change," *Psychological Review*, vol. 84, no. 2, 1977, pp. 191–215, doi: 10.1037/0033-295X.84.2.191.

[7] A. Bandura, *Social foundations of thought and action: A social cognitive theory*. in Social foundations of thought and action: A social cognitive theory. Englewood Cliffs, NJ, US: Prentice-Hall, Inc, 1986, pp. xiii, 617.

[8] A. Bandura, D. Ross, and S. A. Ross, "Transmission of aggression through imitation of aggressive models," *The Journal of Abnormal and Social Psychology*, vol. 63, no. 3, 1961, pp. 575–582, doi: 10.1037/h0045925.

[9] A. Byrski, *et al.*, "Socio-cognitively inspired ant colony optimization," *Journal of Computational Science*, vol. 21, Jul. 2017, pp. 397–406, doi: 10.1016/j.jocs.2016.10.010.

[10] I. Bugajski, *et al.*, "Enhancing Particle Swarm Optimization with Socio-cognitive Inspirations," *Procedia Computer Science*, vol. 80, Jan. 2016, pp. 804–813, doi: 10.1016/j.procs.2016.05.370.

[11] A. Urbanczyk, B. Nowak, P. Orzechowski, J. H. Moore, M. Kisiel-Dorohinicki, and A. Byrski, "Socio-cognitive Evolution Strategies," in *Computational Science – ICCS 2021*, M. Paszynski, D. Kranzlmüller, V. V. Krzhizhanovskaya, J. J. Dongarra, and P. M. A. Sloot, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2021, pp. 329–342. doi: 10.1007/978-3-030-77964-1_26.

[12] E.-G. Talbi, "A Taxonomy of Hybrid Metaheuristics," *Journal of Heuristics*, vol. 8, no. 5, Sep. 2002, pp. 541–564, doi: 10.1023/A:1016540724870.

[13] Y.-S. Ong, M.-H. Lim, N. Zhu, and K.-W. Wong, "Classification of adaptive memetic algorithms: a comparative study," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 36, no. 1, Feb. 2006, pp. 141–152, doi: 10.1109/TSMCB.2005.856143.

[14] Robert Schaefer, Aleksander Byrski, Joanna Kolodziej, and Maciej Smolka. An agent-based model of hierarchic genetic search. *Comput. Math. Appl.*, 64(12):3763–3776, 2012.

[15] Kamil Pietak, Adam Wos, Aleksander Byrski, and Marek Kisiel-Dorohinicki. Functional integrity of multi-agent computational system supported by component-based implementation. In Vladimír Marík, Thomas I. Strasser, and Alois Zoitl, editors, *Holonic and Multi-Agent Systems for Manufacturing, 4th International Conference on Industrial Applications of Holonic and Multi-Agent Systems, HoloMAS 2009, Linz, Austria, August 31 - September 2, 2009. Proceedings*, volume 5696 of *Lecture Notes in Computer Science*, pages 82–91. Springer, 2009.

[16] Robert Schaefer, Aleksander Byrski, and Maciej Smolka. Stochastic model of evolutionary and immunological multi-agent systems: Parallel execution of local actions. *Fundam. Informaticae*, 95(2-3):325–348, 2009.

[17] I. Rechenberg, *Cybernetic Solution Path of an Experimental Problem by Ingo Rechenberg*. Royal Aircraft Establishment, 1965.

[18] H.-P. Schwefel, *Numerische Optimierung von Computer-Modellen mittels der Evolutions strategie: Mit einer vergleichenden Einführung in die Hill-Climbing- und Zufallsstrategie*. Basel: Birkhäuser, 1977. doi: 10. 1007/978-3-0348-5927-1.

[19] D. V. Arnold, "Weighted multirecombination evolution strategies," *Theoretical Computer Science*, vol. 361, no. 1, Aug. 2006, pp. 18–37, doi: 10.1016/j.tcs.2006.04.003.

[20] D. Brockhoff, A. Auger, N. Hansen, D. V. Arnold, and T. Hohm, "Mirrored Sampling and Sequential Selection for Evolution Strategies," in *Parallel Problem Solving from Nature, PPSN XI*, R. Schaefer, C. Cotta, J. Kołodziej, and G. Rudolph, Eds., in Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2010, pp. 11–21. doi: 10.1007/978-3-642-15844-5_2.

[21] T.-Y. Huang and Y.-Y. Chen, "Modified evolution strategies with a diversity-based parent-inclusion scheme," in *Proceedings of the 2000. IEEE International Conference on Control Applications. Conference Proceedings (Cat. No.00CH37162)*, Sep. 2000, pp. 379–384. doi: 10.1109/CCA.2000.897454.

[22] N. Hansen and A. Ostermeier, "Completely Derandomized Self-Adaptation in Evolution Strategies," *Evolutionary Computation*, vol. 9, no. 2, Jun. 2001, pp. 159–195, doi: 10.1162/106365601750190398.

[23] P. P. Repoussis, C. D. Tarantilis, O. Bräysy, and G. Ioannou, "A hybrid evolution strategy for the open vehicle routing problem," *Computers & Operations Research*, vol. 37, no. 3, Mar. 2010, pp. 443–455, doi: 10.1016/j.cor.2008.11.003.

[24] D. Koulocheris, H. Vrazopoulos, and V. Dertimanis, "Hybrid evolution strategy for the design of welded beams," in *Proc. of Int. Congress on Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems EUROGEN 2003*, 2003.

[25] L. Dos Santos Coelho and P. Alotto, "Electromagnetic device optimization by hybrid evolution strategy approaches," *COMPEL – The international journal for computation and mathematics in electrical and electronic engineering*, vol. 26, no. 2, Apr. 2007, pp. 269–279, doi: 10.1108/03321640710727638.

[26] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," *Journal of Global Optimization*, vol. 11, no. 4, Dec. 1997, pp. 341–359, doi: 10.1023/A:1008202821328.

[27] Kenneth Price, Rainer M. Storn, and Jouni A. Lampinen. *Differential Evolution*. in Natural Computing Series. Berlin/Heidelberg: Springer-Verlag, 2005. doi: 10.1007/3-540-31306-0.

[28] C.-L. Hwang, Y.-J. Lai, and T.-Y. Liu, "A new approach for multiple objective decision making," *Computers & Operations Research*, vol. 20, no. 8, Oct. 1993, pp. 889–899, doi: 10.1016/0305-0548(93)90109-V.

[29] M. Nabywaniec, et al., "Socio-cognitive Optimization of Time-delay Control Problems using Agent-based Metaheuristics," in *2022 IEEE 11th International Conference on Intelligent Systems (IS)*, Oct. 2022, pp. 1–7. doi: 10.1109/IS57118.2022.10019693.

[30] P. Kipinski, et al., "Socio-cognitive Optimization of Time-delay Control Problems using Evolutionary Metaheuristics." arXiv, Oct. 23, 2022. doi: 10.48550/arXiv.2210.12872.

[31] J. Dieterich and B. Hartke, "Empirical Review of Standard Benchmark Functions Using Evolutionary Global Optimization," *Applied Mathematics*, vol. 03, Jul. 2012, doi: 10.4236/am.2012.330215.

[32] M. López-Ibáñez, J. Dubois-Lacoste, L. Pérez Cáceres, M. Birattari, and T. Stützle, "The irace package: Iterated racing for automatic algorithm configuration," *Operations Research Perspectives*, vol. 3, Jan. 2016, pp. 43–58, doi: 10.1016/j.orp.2016.09.002.