European Proceedings of
Computers and Technology
EpCT

www.europeanproceedings.com

e-ISSN: 2672-8834

DOI: 10.15405/epct.23021.13

HMMOCS 2022
International Workshop "Hybrid methods of modeling and optimization in complex systems"

ADAPTIVE DIFFERENTIAL EVOLUTION WITH TWO POPULATIONS OF NEW AND BEST INDIVIDUALS

Vladimir Stanovov (a)*, Shakhnaz Akhmedova (b), Eugene Semenkin (c)
*Corresponding author

(a) Siberian Federal University, Krasnoyarsk, 660074, Russian Federation, vladimirstanovov@yandex.ru
(b) Independent Researcher, Berlin, 12489, Germany, shahnaz@inbox.ru
(c) Siberian Federal University, Krasnoyarsk, 660074, Russian Federation, eugenesemenkin@yandex.ru

Abstract

In this paper a novel algorithmic scheme for differential evolution is proposed with two populations and new mutation strategies. The populations contain latest and best individuals, and the sizes of these populations are controlled independently. The LL-NADE algorithm uses the parameter adaptation scheme proposed in L-SHADE, and the sizes of both populations are reduced at the same time. The experiments described in the paper are performed on the CEC 2022 benchmark suite. The total standard score is used for comparison of different algorithms on a set of benchmark problems. The experiments are performed with six different mutation strategies, which utilize individuals from both populations. The most efficient mutation strategy utilizes directed search from newest to top individuals. It is shown that LL-NADE is capable of demonstrating results comparable with state-of-the-art algorithms. Comparison on various test functions and dimensions have shown that LL-NADE performs in a way that is not similar to other differential evolution algorithms, achieving better results on some problems, but worse on other.

Keywords: Differential evolution, population size, parameter adaptation

https://doi.org/10.15405/epct.23021.13
Corresponding Author: Vladimir Stanovov
Selection and peer-review under responsibility of the Organizing Committee of the conference
eISSN: 2672-8834

1. Introduction

The development of modern optimization methods is one of the important directions of computational intelligence research nowadays, as such methods are often utilized in other learning or search algorithms. The heuristic optimization, which includes evolutionary algorithms (EAs), focuses on black box optimization, including single-objective, multi-objective, constrained and other types of problems (Sloss & Gustafson, 2020). The differential evolution (DE) (Price et al., 2006) today represents a popular family of approaches, which are relatively simple and easy to implement, and at the same time very efficient in many scenarios. This can be seen by the success of the DE-based algorithms in optimization competitions in the last several years, such as Congress on Evolutionary Computation competition on single-objective numerical optimization, where DE is taking leading places (Škvorc et al., 2019).

One of the problems of DE is the sensitivity to parameter values, which lead to numerous studies on parameter adaptation schemes, with SHADE being one of the most popular approaches (Tanabe & Fukunaga, 2013). However, few attempts have been made in changing the basics of the DE algorithmic scheme. A recent study on Unbounded DE (UDE) (Kitamura & Fukunaga, 2022) has shown that it is possible to apply specific selection mechanisms with infinitely growing population to achieve competitive results.

In this paper the dual population algorithm is proposed, which further develops the ideas of UDE. LL-NADE (Linear and Linear populations sizes reduction Newest and top Adaptive Differential Evolution) maintains the newest population, which contains recently discovered solutions, and top population, that keeps the best solutions throughout the whole search process. The numerical experiments are performed on the CEC 2022 benchmark problems (Kumar et al., 2021). The main features of the proposed algorithm are the new mutation strategies, combining top and newest solutions, and the novel selection (replacement) scheme for the population of newest individuals.

2. Problem Statement

This paper is focused on the problem of developing new algorithmic schemes for differential evolution for solving numerical optimization tasks. Most known algorithmic schemes rely on a single population and an archive, while other possibilities should be discovered.

3. Research Questions

The problem statement raises a set off questions, including the following. First of all, is it possible to develop an algorithmic scheme with more than one population and find an efficient control strategy. Secondly, if the first population contains new individuals, and the second keeps the best, what should be the mutation strategies in such design? In third place, what would be the performance of such approach compared to the state-of-the-art algorithms?

https://doi.org/10.15405/epct.23021.13
Corresponding Author: Vladimir Stanovov
Selection and peer-review under responsibility of the Organizing Committee of the conference
eISSN: 2672-8834

4. Purpose of the Study

The purpose of this study is to propose a new efficient differential evolution algorithm and estimate its efficiency on a set of benchmark problems compared to the best known alternative approaches.

5. Research Methods

5.1. Differential evolution

The differential evolution algorithm was proposed by Storn and Price 1995. The differential evoltuion is a population-based heuristic for numerical optimization. The algorithm begins with initialization of a set of N individuals $x_i = (x_{i,1}, x_{i,2}, ..., x_{i,D})$, where D is the dimensionality of the search space:

$$x_{i,j} = x_{lb,j} + rand \times (x_{ub,j} - x_{lb,j}).$$

Here rand is a uniformly distributed random number in $[0,1]$, $x_{lb,j}$ and $x_{ub,j}$ are the lower and upper boundaries of variable j, $j = 1,...,D$.

The main search operator of DE algorithm is the difference-based mutation. There are several popular mutation strategies, with current-to-pbest being one of the best:

$$v_{i,j} = x_{i,j} + F \times (x_{pbest,j} - x_{i,j}) + F \times (x_{r1,j} - x_{r2,j}),$$

where v_i is called mutant vector, $x_{i,j}$, is the value from current population, j-th coordinate of the i-th solution, the indexes i, $r1$ and $r2$ are different from each other, and F is the scaling factor parameter, chosen from $[0,1]$, $pbest$ is the index of one of the $pb*100\%$ individuals, sorted by fitness, and different from i, $r1$ and $r2$.

The most widely used crossover operator in DE is the binomial crossover, which is applied after mutation:

$$u_{i,j} = \begin{cases} v_{i,j} \ if \ rand < Cr \ or \ j = jrand \\ x_{i,j} \ otherwise \end{cases},$$

where $Cr \in [0,1]$ is the crossover rate, and $jrand$ is a randomly chosen index from $[1,D]$, required to make sure that at least one index is taken from the donor vector, u_i is called trial vector.

After applying bound constraint handling method, such as midpoint-target, the fitness value of the trial vector is calculated, and the selection step is applied:

$$x_i = \begin{cases} u_i \ if \ f(u_i) \le f(x_i) \\ x_i \ otherwise \end{cases}$$

The greedy selection in original DE allows it to always increase the average fitness of the population or at least to keep it at the same level.

5.2. Proposed approach

As mentioned above, the main idea of LL-NADE, namely the usage of newest and top individuals, is inspired by UDE, which has shown that proper selection strategies in combination with the whole search history could be helpful in achieving better performance. The LL-NADE algorithm first initializes a population of N_{init} individuals x_i^{init}, $i = 1,...,N_{init}$. After that, the individuals in this population are evaluated and sorted by their fitness values, N_{new} individuals are copied to the newest population x^{new}, and N_{top} individuals are copied to the top population x^{top}. The N_{init} parameter should be set larger than both N_{new} and N_{top}. The LL-NADE uses variants of the mutation strategy used in JADE, SHADE and L-SHADE, i.e. current-to-pbest mutation strategy. LL-NADE also applies adaptation of parameters proposed in the SHADE algorithm, however the archive is not used. The modified mutation strategies use individuals from both populations and work as follows:

- r-new-to-ptop/t/t: $v_{i,j} = x_{r1,j}^{new} + F \times (x_{pbest,j}^{top} - x_{i,j}^{new}) + F \times (x_{r2,j}^{top} - x_{r3,j}^{top})$
- r-new-to-ptop/t/n: $v_{i,j} = x_{r1,j}^{new} + F \times (x_{pbest,j}^{top} - x_{i,j}^{new}) + F \times (x_{r2,j}^{top} - x_{r3,j}^{new})$,
- r-new-to-ptop/n/t: $v_{i,j} = x_{r1,j}^{new} + F \times (x_{pbest,j}^{top} - x_{i,j}^{new}) + F \times (x_{r2,j}^{new} - x_{r3,j}^{top})$,
- r-new-to-ptop/n/n: $v_{i,j} = x_{r1,j}^{new} + F \times (x_{pbest,j}^{top} - x_{i,j}^{new}) + F \times (x_{r2,j}^{new} - x_{r3,j}^{new})$,
- r-new-to-pnew/n/n: $v_{i,j} = x_{r1,j}^{new} + F \times (x_{pbest,j}^{new} - x_{i,j}^{new}) + F \times (x_{r2,j}^{new} - x_{r3,j}^{new})$,
- r-top-to-ptop/t/t: $v_{i,j} = x_{r1,j}^{top} + F \times (x_{pbest,j}^{top} - x_{i,j}^{top}) + F \times (x_{r2,j}^{top} - x_{r3,j}^{top})$

Here r-new and r-top mean that for a target solution a random individual will be chosen from the population of newest solutions or the population of top solutions. Next, pnew and ptop mean that one of the $pb\%$ individuals, sorted by fitness, will be chosen from either newest or top population. Finally, t/n stands for using top or newest individuals in the last difference. Unlike most DE algorithms, here the target vector is chosen randomly, instead of being the i-th vector from the population.

For the pb parameter the following control scheme is used: at the start the pb value is set to pb_{max}, and it is decreased to pb_{min} by the end of the search linearly as follows:

$$pb_{g+1} = round(\frac{pb_{min} - pb_{max}}{NFE_{max}} NFE) + pb_{max},$$

where g is the generation number. Note that if the $pb * N_{new}$ or $pb * N_{top}$, value, depending on the mutation strategy drops to 1, then it is set to 2, so there is always a choice between two best solutions.

One other important feature of the LL-NADE algorithm is the new selection (replacement) mechanism. The selection used in the algorithm depends on the chosen mutation strategy, i.e. if the target vector is from newest or top population. If the newly generated vector is better than the target vector, it is saved to the newest population. The place for the new solution is saved is determined by index nc, which is iterated from 1 to N_{new}, and is incremented only if there is a successful solution. The replacement mechanism in LL-NADE works as shown below:

$$x_{nc} = \begin{cases} u_i \ if \ f(u_i) \le f(x_{r1}^{t|n}) \\ x_{nc} \ otherwise \end{cases},$$

where $t \mid n$ for the target vector means that it can be chosen from newest population or top population. The new solutions, which replace old ones in the population, can be used for generating other vectors in the same generations. Besides, successful vectors are stored in x^{temp}, a temporary population. At the end of each generation x^{temp} and x^{top} are joined, sorted, and only N_{top} solutions are copied back to x^{top}. This ensures that the top population always contains the best solutions from the whole search process. LL-NADE also uses linear population size reduction, same as in L-SHADE algorithm (Tanabe and Fukunaga, 2014), however here it is used for two populations, newest and top, at the same time.

6. Findings

The computational experiments with LL-NADE were performed on the CEC 2022 benchmark suite, developed for the Single Objective Bound Constrained Numerical Optimization Competition (Kumar et al., 2021). The benchmark contained 12 functions, defined for 10D and 20D, 30 independent runs were required for each of them. The algorithm was developed in C++ and ran on OpenMPI-powered cluster. For the first set of experiments the population sizes N^{new} and N^{top} changed from 5D to 60D with a step of 5D, and 6 mutation strategies were considered. This resulted in a total of 864 experiments. To compare the efficiency, the Mann-Whitney rank sum test with tie-breaking and normal approximation was applied with significance level of 0.01. In addition, the total standard score, summed over 12 functions, was calculated to compare a pair of approaches. The parameters of LL-NADE used in the experiments are the following: $N^{init} = 100D$, $H = 7$, $M_{F,r} = 0.3$, $M_{Cr,r} = 0.8$, $pb_{max} = 0.3$, $pb_{min} = 0.01$. For the first set of experiments the NL-SHADE-LBC algorithm (Stanovov et al., 2022) was chosen as the baseline approach, as it is announced to be second best algorithm in CEC 2022 competition. Figures 1 and 2 show the total standard score of comparing LL-NADE with NL-SHADE-LBC with different population sizes and strategies. Lighter areas mean better results, i.e. larger total score.

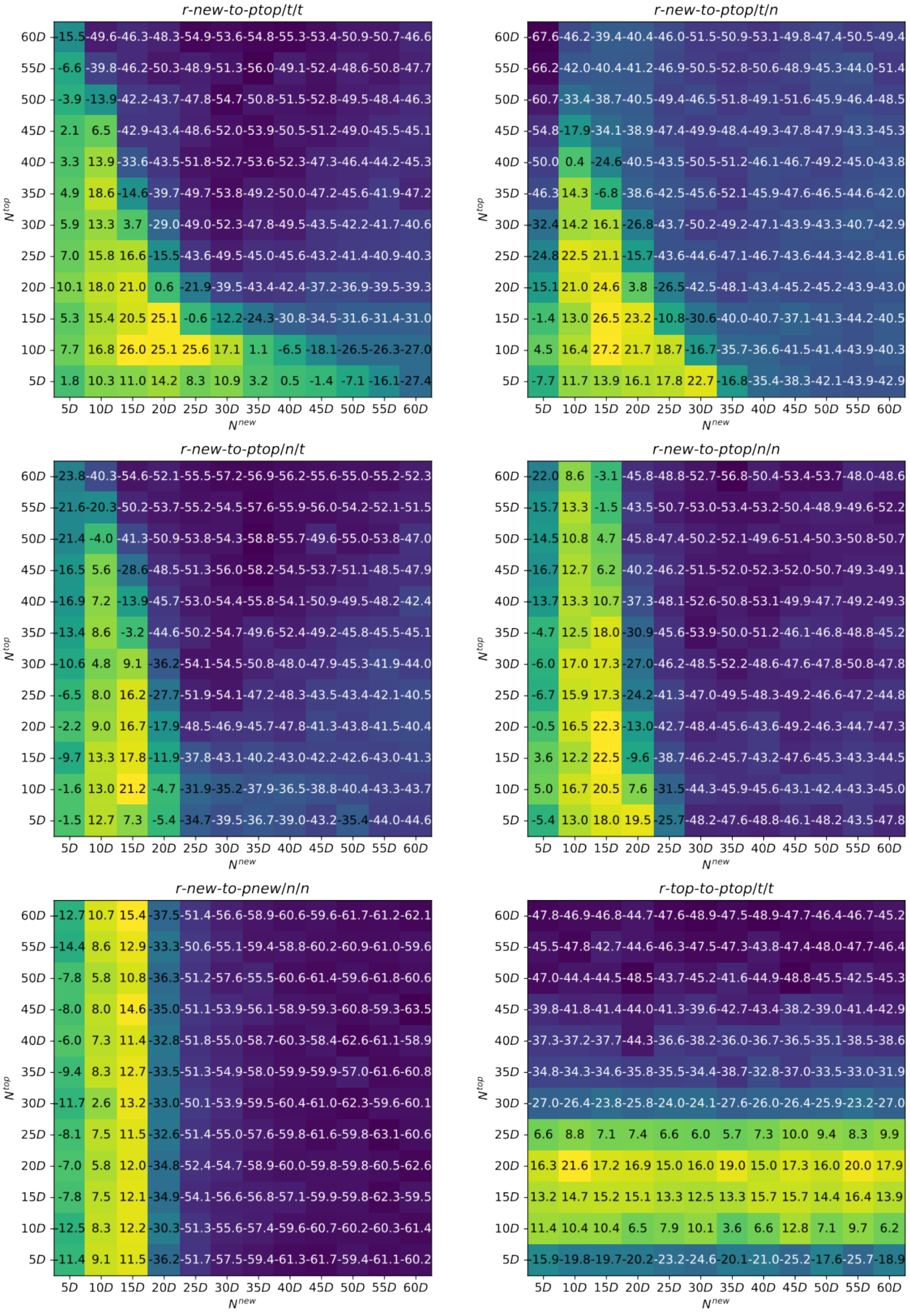

Figure 1. Total Mann-Whitney scores of LL-NADE compared to NL-SHADE-LBC, *10D*

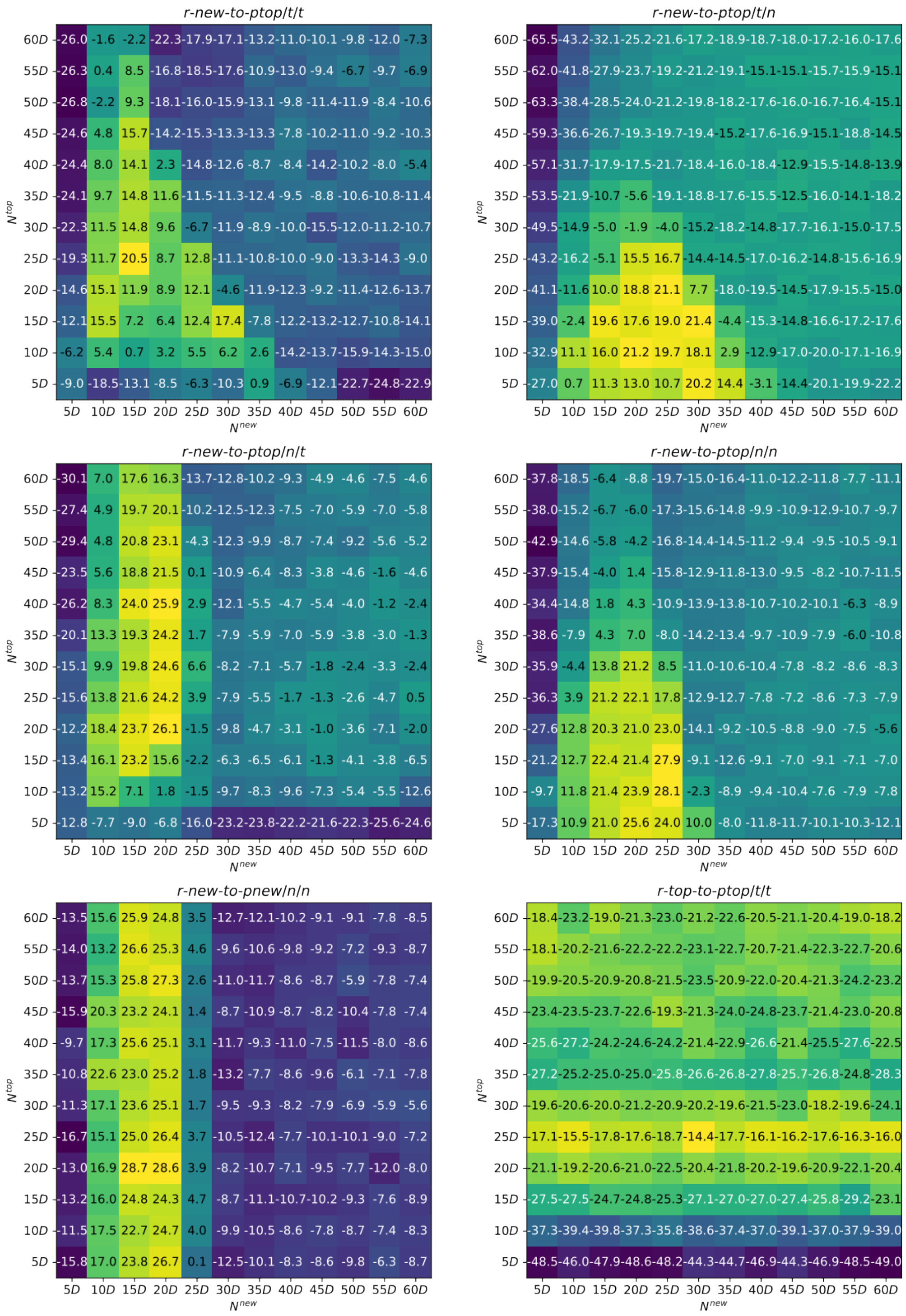

Figure 2. Total Mann-Whitney scores of LL-NADE compared to NL-SHADE-LBC, *20D*

Considering the results in Figures 1, it can be concluded that for all mutation strategies for 10D functions the area of highest efficiency is around 15D or 20D for both N^{new} and N^{top}. For r-new-to-ptop/t/n and r-new-to-ptop/t/t there is a clearer dependence of two populations sizes, while r-new-to-ptop/n/t and r-new-to-ptop/n/n are characterized by smaller influence of N^{top}. Speaking of the efficiency of the proposed algorithms, LL-NADE is able to outperform NL-SHADE-LBC using all mutation strategies except the last one, r-top-to-ptop/t/t. The efficiency of r-new-to-pnew/n/n is also lower compared to the strategies which use both populations.

Figure 2 shows similar results, i.e. total standard score of Mann-Whitney tests is larger than 0 for population sizes of around 20D. An important difference compared to 10D case is that the strategy using only newest vectors r-new-to-pnew/n/n may have the same level of efficiency compared to strategies using both populations. Also, increasing the population size significantly decreases the efficiency of the algorithm, probably because the convergence speed is also considered.

To conclude the experiments section, we provide the comparison of LL-NADE using r-new-to-ptop/n/t strategy and $N^{new} = N^{top} = 15D$ with best participants of CEC 2022 competition, the results are shown in Table 1. The top 3 algorithms are EA4eig (Bujok & Kolenovsky, 2022) combining several DE variants and CMA-ES, mentioned earlier NL-SHADE-LBC (Stanovov et al., 2022) and NL-SHADE-RSP-MID (Biedrzycki et al., 2022), an advanced version of the NL-SHADE-RSP with k-means algorithm. The numbers in Table 1 are the numbers of wins/ties/losses over 12 tested functions.

Table 1. Mann-Whitney tests of LL-NADE using *r-new-to-ptop/n/t* against the top 3 algorithms of CEC 2022 competition, number of wins/ties/losses

Dimension	EA4eig	NL-SHADE-LBC	NL-SHADE-RSP-MID
10D	5/3/4	5/4/3	5/3/4
20D	7/2/3	6/4/2	8/2/2
Total	12/5/7	11/8/5	13/5/6

The results in Table 3 show that LL-NADE has comparable or even better performance against the top 3 methods of the recent CEC competition, which shows that the proposed approach can be promising.

7. Conclusion

In this paper a new differential evolution algorithm was proposed, which used a new algorithmic scheme with two populations and novel mutations. The experiments performed in this study have shown that the proposed algorithm LL-NADE is highly competitive compared to the best methods in the CEC 2022 benchmark, especially on complex functions. One of the disadvantages of LL-NADE is that it requires more parameters to be set compared to known alternatives. The LL-NADE is a non-hybrid method, and it does not increase the computational complexity compared to modern DE variants, and it can be improved by implementing modifications, developed for other DE algorithms.

Acknowledgments

This research was funded by the Ministry of Science and Higher Education of the Russian Federation, Grant No. 075-15-2022-1121.

https://doi.org/10.15405/epct.23021.13
Corresponding Author: Vladimir Stanovov
Selection and peer-review under responsibility of the Organizing Committee of the conference
eISSN: 2672-8834

References

Biedrzycki, R., Arabas, J., & Warchulski, E. (2022, July). A version of nl-shade-rsp algorithm with midpoint for cec 2022 single objective bound constrained problems. In *2022 IEEE Congress on Evolutionary Computation (CEC)* (pp. 1-8). IEEE. https://doi.org/10.1109/cec55065.2022.9870220

Bujok, P., & Kolenovsky, P. (2022, July). Eigen crossover in cooperative model of evolutionary algorithms applied to cec 2022 single objective numerical optimisation. In *2022 IEEE Congress on Evolutionary Computation (CEC)* (pp. 1-8). IEEE. https://doi.org/10.1109/CEC55065.2022.9870433

Kitamura, T., & Fukunaga, A. (2022). Differential Evolution with an Unbounded Population. In *2022 IEEE Congress on Evolutionary Computation (CEC)* (pp. 1-8). IEEE. https://doi.org/10.1109/CEC55065.2022.9870363

Kumar, A., Price, K., Mohamed, A. K. A. A., & Suganthan, P. N. (2021). Problem Definitions and Evaluation Criteria for the CEC 2022 Special Session and Competition on Single Objective Bound Constrained Numerical Optimization. Technical Report Nanyang Technological University, Singapoure.

Price, K., Storn, R. M., & Lampinen, J. A. (2006). *Differential evolution: a practical approach to global optimization.* Springer Science & Business Media.

Škvorc, U., Eftimov, T., & Korošec, P. (2019). CEC real-parameter optimization competitions: Progress from 2013 to 2018. In 2019 IEEE congress on evolutionary computation (CEC) (pp. 3126-3133). IEEE. https://doi.org/10.1109/CEC.2019.8790158

Sloss, A. N., & Gustafson, S. (2020). 2019 Evolutionary Algorithms Review. In W. Banzhaf, E. Goodman, L. Sheneman, L. Trujillo, & B. Worzel (Eds.), *Genetic Programming Theory and Practice XVII. Genetic and Evolutionary Computation* (pp. 307-344). Springer, Cham. https://doi.org/10.1007/978-3-030-39958-0_16

Stanovov, V., Akhmedova, S., & Semenkin, E. (2022). NL-SHADE-LBC algorithm with linear parameter adaptation bias change for CEC 2022 Numerical Optimization. In *2022 IEEE Congress on Evolutionary Computation (CEC)* (pp. 01-08). IEEE. https://doi.org/10.1109/CEC55065.2022.9870295

Tanabe, R., & Fukunaga, A. (2013). Success-history based parameter adaptation for differential evolution. In 2013 IEEE congress on evolutionary computation (pp. 71-78). IEEE. https://doi.org/10.1109/cec.2013.6557555

Tanabe, R., & Fukunaga, A. (2014). Improving the search performance of SHADE using linear population size reduction*. In Proceedings of the IEEE Congress on Evolutionary Computation, CEC* (pp. 1658-1665). https://doi.org/10.1109/CEC.2014.6900380