On Modifications Towards Improvement of the Exploitation Phase for SOMA Algorithm with Clustering-aided Migration and Adaptive Perturbation Vector Control

Tomas Kadavy
Faculty of Applied Informatics
Tomas Bata University in Zlin
Zlin, Czech Republic
kadavy@utb.cz

Adam Viktorin
Faculty of Applied Informatics
Tomas Bata University in Zlin
Zlin, Czech Republic
aviktorin@utb.cz

Michal Pluhacek
Faculty of Applied Informatics
Tomas Bata University in Zlin
Zlin, Czech Republic
pluhacek@utb.cz

Stanislav Kovar
Faculty of Applied Informatics
Tomas Bata University in Zlin
Zlin, Czech Republic
skovar@utb.cz

***Abstract*—This paper represents the next step in the development of the recently proposed single objective metaheuristic algorithm - Self-Organizing Migrating Algorithm with CLustering-aided migration and adaptive Perturbation vector control (SOMA-CLP). The CEC 2021 single objective bound-constrained optimization benchmark testbed was used for the performance evaluation of the modifications of the algorithm. The presented modifications were invoked by the results of CEC 2021 competition, where the SOMA-CLP ranked 7th out of 9 competing algorithms. This paper introduces three modifications of population organization process focusing on one particular phase of the SOMA-CLP algorithm aimed at exploitation. All results were compared and tested for statistical significance against the original variant using the Friedman rank test. The algorithm modification and analysis of the results presented here can be inspiring for other researchers working on the development and modifications of evolutionary computing techniques.**

***Index Terms*—SOMA, SOMA-CLP, k-means, clustering, benchmarking, CEC 2021, population dynamics**

I. Introduction

In recent decades, evolutionary algorithms (EA) gained popularity and reputation as robust and effective tools for solving various optimization tasks. This reputation is probably the main drive behind the number of newly developed metaheuristic algorithms. New algorithms or modifications of already established ones are presented to the community every year in enormous quantities, and it is nearly impossible to maintain informed overview. A powerful tool that helps to discover and compare interesting metaheuristic algorithms is benchmarking. For example, benchmark set for single-objective optimization in the form of a special session in IEEE Congress on Evolutionary Computation (CEC). Since 2005 [1], the series of CEC benchmarks represent a substantial pool of the most suitable test function candidates and rank the algorithms based on their overall performance. Therefore, anybody can use the CEC benchmark to select the state-of-the-art algorithm or compare new modifications or candidates against already established algorithms.

The paper is focused on a newly developed algorithm that has been tested on the CEC 2021 benchmarking testbed [2]. The algorithm represents a modern variant of the Self-Organizing Migrating Algorithm (SOMA) named Self-Organizing Migrating Algorithm with CLustering-aided migration and adaptive Perturbation vector control (SOMA-CLP).

Recently, SOMA-based algorithms have begun to appear at CEC competitions. As for the CEC 2019 competition [3], three SOMA representatives were attending which one of them achieved third place, therefore showing promising potential for this algorithms family.

The SOMA-CLP algorithm is a direct descendant of SOMA-CL [4]. SOMA-CLP [5] uses a linear adaptation of the *prt* control parameter, promoting the global transition from the tendency of exploration to exploitation. The workflow of the SOMA-CLP can be divided into three phases: search space mapping, clustering of the mapped space, and the exploitation by performing a more detailed screening of areas of interest discovered during the first phase. All three phases thus define one iteration of the algorithm.

The 2021 CEC benchmarking testbed encompasses several categories: non-shifted, shifted, non-rotated shifted, and rotated shifted cases. Based on the official ranking, the SOMA-CLP was placed 7th out of 9 contestants, which suggests a lot of room for improvement. Therefore, this paper is investigating several possible techniques for the improvement of the overall performance of the SOMA-CLP.

The paper is structured as follows. The next section covers SOMA and SOMA-CLP; section III describes new modifications proposed for SOMA-CLP; section IV provides the experiment setup with benchmark results, and the last section contains concluding remarks.

This work was supported by the Internal Grant Agency of Tomas Bata University under the Projects no. IGA/CebiaTech/2021/001, and further by the resources of A.I.Lab at the Faculty of Applied Informatics, Tomas Bata University in Zlin (ailab.fai.utb.cz).

II. Algorithm Descriptions

The following subsections cover the description of the SOMA and SOMA-CLP. Firstly, the basic, original SOMA is described together with its commonly used strategies. Secondly follows the description of its modern variant SOMA-CLP, introduced on the latest CEC 2021 benchmark.

A. SOMA

The Self-Organizing Migrating Algorithm (SOMA) was initially developed in 1999 by I. Zelinka [6], [7]. SOMA takes inspiration in self-organization and cooperative behavior while maintaining some of the fundamentals of nature-inspired methods. The discrete perturbation mimics the mutation process while the self-adaptation of movement over the search space allows easy scalability.

As mentioned, SOMA is based on the cooperation of individuals. Hence, the candidate solution is represented by an individual $\boldsymbol{x}$. The cooperation amongst individual is, by author, defined as a migration (1) of one particular individual from population towards another member of the population.

$$x_{i,j}^{k+1} = x_{i,j}^{k} + \left(x_{L,j}^{k} - x_{i,j}^{k}\right) \cdot t \cdot PRTVector_j \qquad (1)$$

The $x_{i,j}^{k+1}$ is a new position of an i-th individual in j-dimension for a next iteration step $k+1$. Accordingly, the $x_{i,j}^{k}$ is a position of the same individual in k iteration. The $x_{L,j}^{k}$ the position of a leader, which is selected based on the selected SOMA strategy (SOMA strategies are described below). Individual discrete steps between an i-th individual and selected leader $x_{L,j}^{k}$ are represented by t parameter. The best-found solution on this path is then transferred into a new iteration. The t parameter is a collection of values starting from 0 to $PathLength$ with increment (or step size) of $Step$.

The $PRTVector_j$ mimics the mutation process and is generated as (2) for all the individual t steps. This vector determines in which dimensions j the i-th individual will migrate towards a leader and which dimensions stay unchanged. From the equation (2) it is clear that the parameter prt has a direct impact on the resulting $PRTVector_j$ and on the strength of a mutation during the migration. This prt parameter can be considered as a threshold value and is chosen in the range from 0 to 1.

$$PRTVector_j = \begin{cases} 1 & ,\text{if}(rand_j < prt) \\ 0 & ,\text{otherwise} \end{cases} \qquad (2)$$

Original SOMA describes several different strategies for the leader selection. Three most common strategies are described bellow.

1) Strategy All-To-One: This easy to implement strategy will select for each migration cycle (one iteration of the algorithm) one leader. The leader is selected based on its objective function value. All the remaining individuals then migrate towards the leader.

2) Strategy All-To-Random: This strategy contains leader individual as in All-To-One strategy. However, the leader is selected randomly for each migrant at the beginning of the migration process.

3) Strategy All-To-All: The selection process of a leader is different for this strategy. One individual migrates towards all other individuals. After the end of the migration of a selected individual, this individual returns to its original position, and the process is repeated for the next individual. The migration cycle ends after all the individuals in population migrated towards each other, and all individuals then update their positions.

B. SOMA-CLP

The metaheuristic algorithm, Self-Organizing Migrating Algorithm with CLustering-aided migration and adaptive Perturbation vector control (SOMA-CLP) is a recent modification of SOMA which is the updated version of its predecessor SOMA-CL. SOMA-CLP uses a linear adaptation of the *prt* control parameter to generate a perturbation vector, promoting the global transition from the tendency of exploration to exploitation as the strength of perturbation of individuals' movement weakens. The workflow of the SOMA-CLP can be divided into three phases. The first exploration phase is focused on space mapping, the second phase is a clustering of the mapped space by k-means method [8], and the third phase is focused on exploitation by carrying out a more detailed screening of areas of interest discovered by the first phase. The end of the last phase also ends one iteration of the algorithm, and the whole process starts again with phase one. Detailed descriptions of the phases are in the following subsections in order of occurrence.

1) Exploration Phase: This phase uses the SOMA with All-To-Random strategy as described in subsection II-A2. The leader is selected randomly from the population set of NP individuals for each active individual ***x***. The migration strategy equation is the same as for the SOMA in (1). The main difference between SOMA-CL and the proposed SOMA-CLP is the usage of the linear adaptation of the prt parameter. Originally, the prt is one of the user-defined parameters of SOMA. The proposed SOMA variant employs the similar adaptivity of the prt parameter as in other modern variants of SOMA [9], [10]. This adaptation affects the covered area by the exploration phase over the algorithm execution. The prt represents the strength of a perturbation during the migration and starts with the low value (exploration), and it is steadily increasing to an upper limit (exploitation). Therefore, at the beginning of the algorithm, the exploration phase covers wider hyperspace of solutions between active individual and leader, and this mapping later becomes more focused on the "direct" path between them. The equation of the adaptive prt is defined as (3).

$$prt = 0.08 + 0.9 \cdot (FES/maxFES) \qquad (3)$$

Where the FES is the number of objective function evaluations in a given time, and the $maxFES$ is the maximal limit of such evaluations.

An essential part of this phase is that each evaluated individual is stored in a memory ***M***. This memory ***M*** represents all visited solutions and is used in the next phase of the algorithm.

2) Clustering of the Mapped Space: The evaluated solutions stored in the memory $\boldsymbol{M}$ from the previous exploration phase are investigated in this second phase. From the memory $\boldsymbol{M}$ are selected candidate leaders for the last exploitation phase. The basic idea is to select only a few promising solutions from the whole covered hyperspace. Therefore, a clustering method to divide all solutions by their parameter values into several groups (clusters) is used. Namely, the k-means clustering method [11]. The number of outcome clusters should be 10% of the NP, or it may be set by the user as NP_L. The k-means algorithm is an iterative algorithm that can be briefly described in three steps. In the first step, k-number of centroids are randomly selected from the pool of visited positions. For the second step, the objects (in this case, positions) are assigned to the nearest centroids (one object can have assigned only one centroid). The third step recalculates the positions of centroids to ensure that the new positions become the new mean. The second and third parts are then repeated until convergence is reached.

From each of the created clusters are selected only solutions with the best objective function value within their cluster – cluster leaders. The cluster leaders are then sorted by their objective function values in ascending order from the best-found solution to the worst.

3) Exploitation Phase: This phase uses the SOMA with the All-To-One strategy with two alterations. The leader $x_{L,j}$ in equation (1) is this time selected from the set of cluster leaders using the Rank Selection technique [12]. The leader is selected for each individual. The individual x_i is migrating by discrete steps, and the best-found solution on *t*-th position is propagated into a new iteration of the algorithm. The *t* parameter is generated in a range starting from 0 to $pathLength_L$ with step size $step_L$. The leader selection with parameters values of $pathLength_L$ and $step_L$ should ensure the exploitation of an interesting solutions discovered in the first phase. The $PRTVector_j$ is generated in the same way as in equation (2), and the prt is again computed by (3).

The described three phases of the SOMA-CLP are then repeated until the stopping condition is met, typically the $maxFES$ is reached.

Algorithm 1 SOMA-CLP

Set D, NP, NP_L, and $MAXFES$
Set *step*, and *pathLength*
Set $step_L$, and $pathLength_L$
while Stopping criterion not met **do**
  $\boldsymbol{M} = \emptyset$
  for $i = 1$ **to** NP **do**
    x_L = pick random solution $\boldsymbol{x}$
    $prt = 0.08 + 0.9 \cdot (FES/maxFES)$
    for $t = 0$ **to** $pathLength$ **with** $t+ = step$ **do**
      generate ***PRTVector*** by eq. (2)
      migrate x_i to x_L by eq. (1)
      save each evaluated solution into $\boldsymbol{M}$
    end for
  end for
  k-means clustering method for solutions stored in $\boldsymbol{M}$
  keep only best-solution from each cluster
  sort the remaining solutions in $\boldsymbol{M}$
  for $i = 1$ **to** NP_L **do**
    x_L = Rank Selection from cluster leaders
    $prt = 0.08 + 0.9 \cdot (FES/maxFES)$
    for $t = 0$ **to** $pathLength_L$ **with** $t+ = step_L$ **do**
      generate ***PRTVector*** by eq. (2)
      migrate x_i to x_L by eq. (1)
    end for
  end for
  record the best solution
end while

III. Proposed Modifications

The following subsections cover the proposed modifications to the original SOMA-CLP. To improve the average algorithm performance on the CEC benchmark, all three modifications were suggested and discussed at the latest *Genetic and Evolutionary Computation Conference, GECCO 2021* [5], where SOMA-CLP was firstly introduced as a CEC 2021 benchmark competition entry. The proposed modifications deal with the mechanisms of the cluster leader selection process for the third phase of the SOMA-CLP. For clarity and future reference, each modification is numbered in order of occurrence. Thus, modification 1 – *Narrow Cluster Leader Selection Process*, modification 2 – *Roulette Selection*, and modification 3 – *Single Cluster Leader*.

A. Narrow Cluster Leader Selection Process

SOMA-CLP defines the number of cluster as a 1/10 of the *NP*, resulting in also the very same amount of so-called cluster leaders. Simple idea was to limit the number of leaders for the third phase of the SOMA-CLP by eliminating some candidates. The elimination process is based on the objective function value of the cluster leaders and only 50% of them can be chosen as leaders in the third phase. We expect that the modification should increase the exploitation of the promising areas at the cost of a faster population diversity decrease.

B. Roulette Selection

Modification number two offers a different selection method for leaders in the third phase of the algorithm. It utilizes a roulette selection method [12] instead of the originally used rank selection. The expected behaviour of roulette selection is as follows. The solution with the best objective function value has the highest probability to be chosen as a leader, the second-best has the second-highest probability of being selected, and so on. The worst solution has the lowest chance of being chosen as a leader.

C. Single Cluster Leader

The last modification, number three, changes how often the leader is selected in the third phase of the SOMA-CLP. In the original algorithm, each individual selects a new leader from the pool of cluster leaders. This modification follows strict rule that the leader is chosen only once in each migration using roulette selection; therefore, all the individuals migrate towards one leader – similar to the All-To-One strategy of the original SOMA.

IV. Experiment Setup

The CEC 2021 Special Session and Competition on Single Objective Bound Constrained Optimization [2] is accompanied by a technical report which describes the benchmark itself together with instructions on how to approach the test problems. It also provides test function definitions and describes the evaluation criteria. The benchmark suite consists of 10 test functions (one unimodal function, three basic multimodal functions, three hybrid functions, and three composition functions). Each test function can be further parametrized. The setting of parametrization vector may enable bias, shift, rotation, or any combination to each test function. The parametrization vector introduces 8 possible configurations to a test function. Therefore, the total number of test functions for one dimension is 80. Each test function has a defined search range in span from -100 to 100 and a different minimum value. The tested dimension sizes are 10 and 20 for all test functions. Each test function, for a particular scenario, should be optimized in 30 independent runs, which then represent the final results. Finally, both of the tested dimension sizes has a fixed budget of maximal function evaluations - $maxFES$.

The values of the control parameters for the SOMA-CLP variants are given in Table. I. All three proposed modifications share the same parameter values for a fair comparison and also they are the same as for the original SOMA-CLP version.

The source code of the SOMA-CLP is available at the A.I.Lab Github[1].

TABLE I: SOMA-CLP parameters

Parameter	**Value**
NP	100
$\mathrm{NP_L}$	10
step	0.33
$\mathrm{step_L}$	0.11
pathLength	3.0
$\mathrm{pathLength_L}$	3.0

V. Results

This section contains the results and performance analyses of all proposed modifications compared to the original SOMA-CLP. The overall performance is evaluated and compared using the Friedman rank tests [13] with the significance level $\alpha = 0.05$, accompanied by Nemenyi critical distance post-hoc test for multiple comparisons [14]. The computed p-values of all presented Friedman rank tests are lower than 0.05; thus, all tests are relevant. The dashed line represents the critical distance from the best-performed algorithm (the lowest mean rank). The lower the rank is, the better is the overall performance of that algorithm on a particular dimension size.

Additionally, the Holm-Bonferroni method was used to test significant differences among SOMA versions on each test function and for all combinations of parametrization vector *C*. The results are shown in Tables II and III for dimension sizes 10 and 20 respectively. The symbol $\surd$ represents a significant difference between two SOMA versions, and the symbol $\times$ stands for the insignificant difference. For example, in Table II can be seen that for test function f_4, the SOMA-CLP is significantly different to Modification 2 only. Due to the strict limitation of the number of pages,complete results are available at the A.I.Lab website[2].

The CEC 2021 benchmark itself is divided into four categories based on the configuration of the test functions. The categories are:

- **Non-shifted Cases** – This category contains four configurations of test functions without the shift of the global minima.
- **Shifted Cases** – This group encompasses four possible configurations of test functions with the shift of the global minima.
- **Non-rotated Shifted Cases** – This category consists of two configurations of test functions.
- **Rotated Shifted Cases** – The last category again contains only two configurations of test functions.

A more detailed explanation of the CEC 2021 benchmark and the reasons behind the division into four categories can be found in the technical report [2]. For more accurate comparisons and detailed analyses, the results presented here are also divided into four cases. However, to compare the overall performance of the modifications, all categories were joined together, and the visual outputs of comparisons on both tested dimension settings (10*D* and 20*D*) with rankings are given in figures Figure 1 and Figure 2. Based on the Friedman rank test for both dimension settings, it is clear that the average performance of the proposed modifications 1 and 2 are not statistically significantly better than the original SOMA-CLP. However, the last modification 3 is significantly different.

A. Non-shifted Cases

The Friedman rank tests for Non-shifted Cases are depicted in figures Figure 3 and Figure 4. The results are almost indistinguishable to the overall combined results (see Figure 1). However, modification 2 using roulette selection of the leaders in the third phase seems to be more suitable for the higher dimension sizes problems.

B. Shifted Cases

The ranking for Shifted Cases are shown in Figure 5 and Figure 6. Again, for the dimension size of 10, the results are very similar to the overall performance in Figure 1. Nevertheless, the results for *D* = 20 have no significant difference in performance among the suggested modifications and the

[1]https://github.com/TBU-AILab/SOMA_CLP

[2]https://ailab.fai.utb.cz/resources/

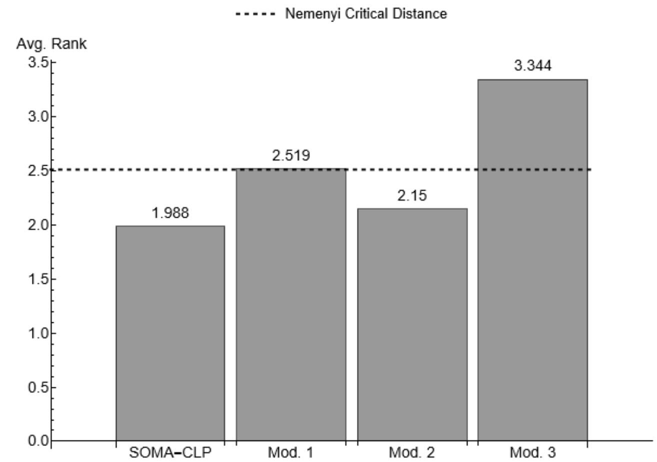

Fig. 1: Friedman rank tests for 10D.

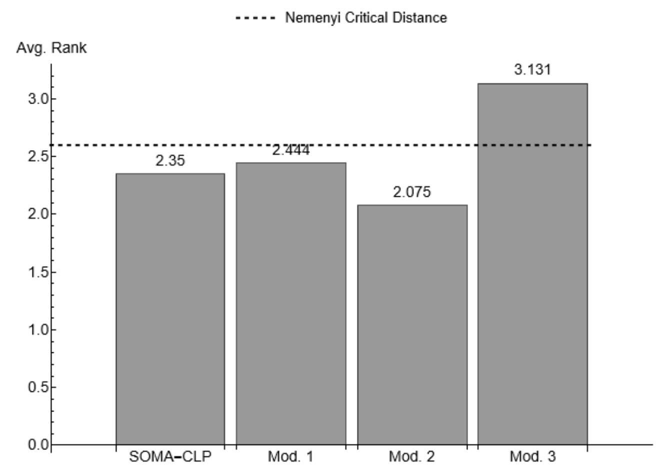

Fig. 2: Friedman rank tests for 20D.

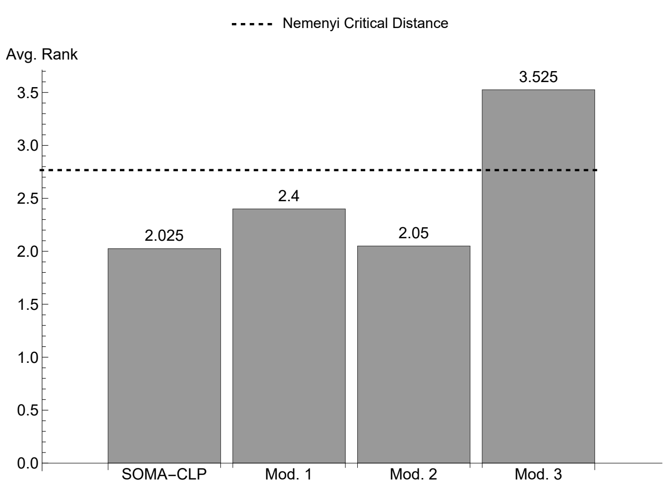

Fig. 3: Friedman rank tests for 10D for Non-shifted Cases.

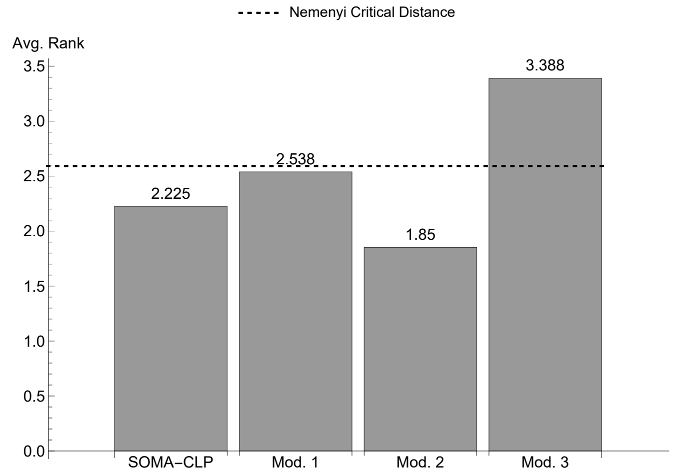

Fig. 4: Friedman rank tests for 20D for Non-shifted Cases.

original SOMA-CLP. The results also suggest a shift of the global minima may be beneficial to modification 3 in higher dimensions, where all individuals migrate towards one selected leader in phase three of the algorithm.

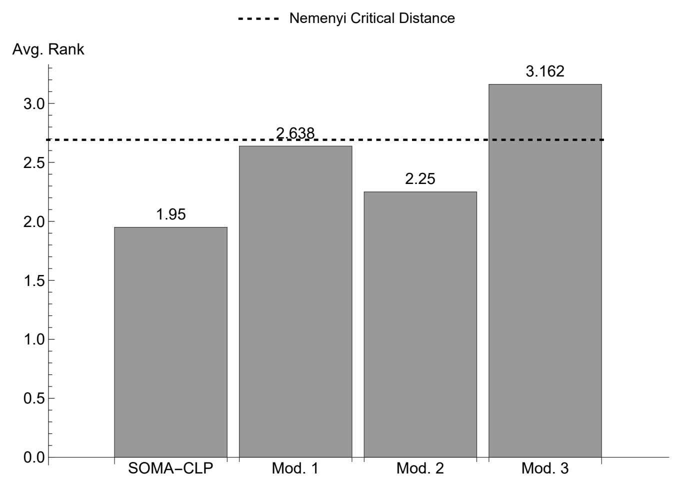

Fig. 5: Friedman rank tests for 10D for Shifted Cases.

C. Non-rotated Shifted Cases

The ranking for Non-rotated Shifted Cases are given in figures Figure 7 and Figure 8. The results for this category of tests vary for different dimension sizes. For $D = 10$, there is no significant difference between the original SOMA-CLP and the first modification. Also, modifications 2 and 3 are significantly worse than the original compared algorithm. The results for dimensional size of 20 show no significant difference between all three modifications and the original SOMA-CLP. However, modification 1 maintains the best average rank. Therefore, a limited number of cluster leaders for the third phase of the algorithm seems to be more beneficial for this case.

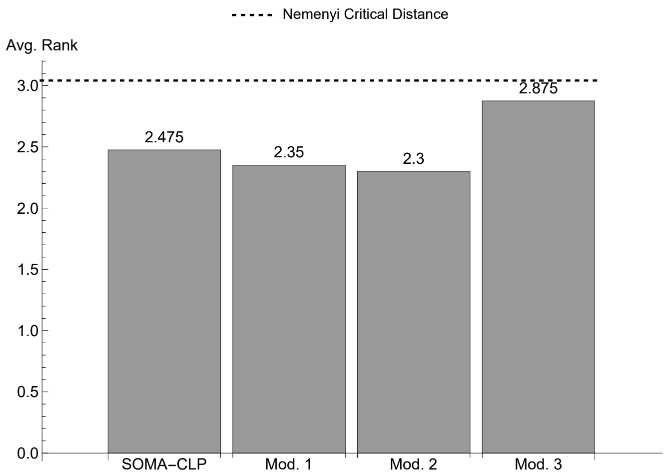

Fig. 6: Friedman rank tests for 20D for Shifted Cases.

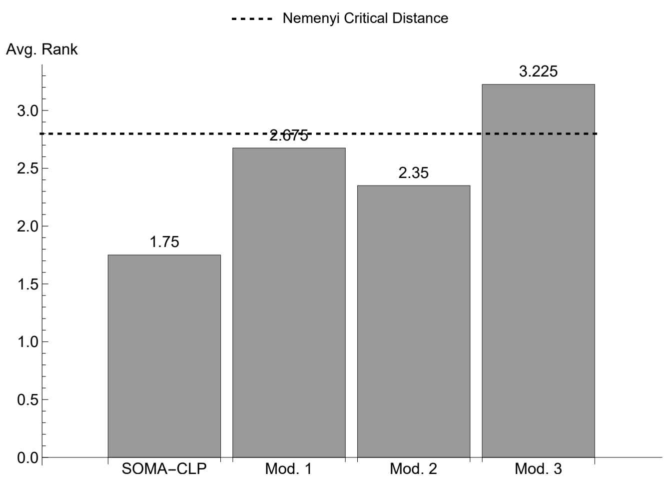

Fig. 7: Friedman rank tests for 10D for Non-rotated Shifted Cases.

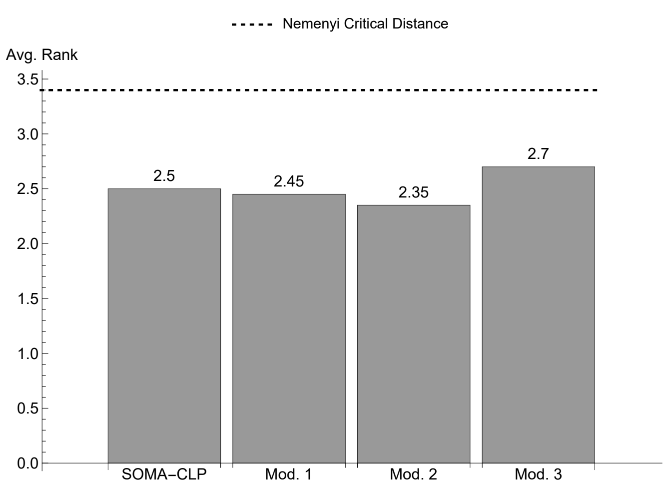

Fig. 8: Friedman rank tests for 20D for Non-rotated Shifted Cases.

D. *Rotated Shifted Cases*

The Friedman rank tests for Rotated Shifted Cases are shown in figures Figure 9 and Figure 10. The results for dimension $D = 10$ are similar to overall combined results, but modification 2 has the best average rank for this case study. For higher dimension settings, again there is no significant difference between original SOMA-CLP and the proposed tested modifications.

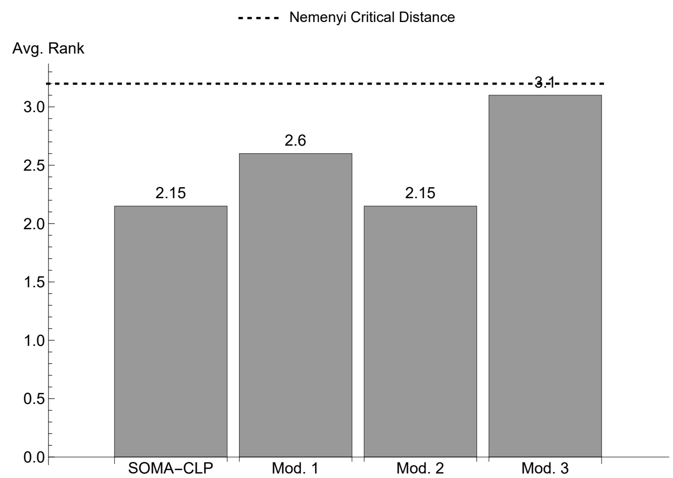

Fig. 9: Friedman rank tests for 10D for Rotated Shifted Cases.

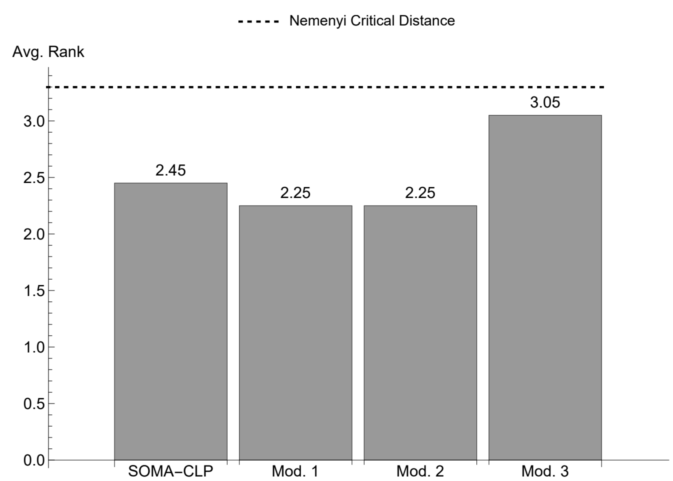

Fig. 10: Friedman rank tests for 20D for Rotated Shifted Cases.

VI. Conclusion

This paper presents and analyses three different modifications of the recently proposed metaheuristic algorithm SOMA-CLP. These new modifications have been designed to improve the overall performance of the metaheuristic algorithm. The results of the modifications were compared with the SOMA-CLP using the CEC 2021 bound-constrained single objective numerical optimization benchmark. All three modifications were focused on one particular phase of the SOMA-CLP algorithm aimed at exploitation.

f_1	O	$M1$	$M2$	$M3$
O	–	√	√	√
$M1$	–	–	√	√
$M2$	–	–	–	√
$M3$	–	–	–	–

f_2	O	$M1$	$M2$	$M3$
O	–	×	×	×
$M1$	–	–	×	×
$M2$	–	–	–	×
$M3$	–	–	–	–

f_3	O	$M1$	$M2$	$M3$
O	–	×	×	×
$M1$	–	–	×	×
$M2$	–	–	–	×
$M3$	–	–	–	–

f_4	O	$M1$	$M2$	$M3$
O	–	×	√	×
$M1$	–	–	×	×
$M2$	–	–	–	×
$M3$	–	–	–	–

f_5	O	$M1$	$M2$	$M3$
O	–	×	×	√
$M1$	–	–	×	√
$M2$	–	–	–	√
$M3$	–	–	–	–

f_6	O	$M1$	$M2$	$M3$
O	–	×	×	×
$M1$	–	–	×	×
$M2$	–	–	–	×
$M3$	–	–	–	–

f_7	O	$M1$	$M2$	$M3$
O	–	×	×	×
$M1$	–	–	×	×
$M2$	–	–	–	×
$M3$	–	–	–	–

f_8	O	$M1$	$M2$	$M3$
O	–	×	×	×
$M1$	–	–	×	×
$M2$	–	–	–	×
$M3$	–	–	–	–

f_9	O	$M1$	$M2$	$M3$
O	–	√	√	×
$M1$	–	–	√	√
$M2$	–	–	–	√
$M3$	–	–	–	–

f_{10}	O	$M1$	$M2$	$M3$
O	–	×	×	×
$M1$	–	–	×	×
$M2$	–	–	–	×
$M3$	–	–	–	–

TABLE II: Holm-Bonferroni procedure for parametrization vector *C*=8 on 10D.

f_1	O	$M1$	$M2$	$M3$
O	–	×	×	×
$M1$	–	–	×	×
$M2$	–	–	–	×
$M3$	–	–	–	–

f_2	O	$M1$	$M2$	$M3$
O	–	×	×	×
$M1$	–	–	×	×
$M2$	–	–	–	×
$M3$	–	–	–	–

f_3	O	$M1$	$M2$	$M3$
O	–	×	×	×
$M1$	–	–	×	×
$M2$	–	–	–	×
$M3$	–	–	–	–

f_4	O	$M1$	$M2$	$M3$
O	–	×	×	×
$M1$	–	–	×	×
$M2$	–	–	–	×
$M3$	–	–	–	–

f_5	O	$M1$	$M2$	$M3$
O	–	√	×	√
$M1$	–	–	×	×
$M2$	–	–	–	×
$M3$	–	–	–	–

f_6	O	$M1$	$M2$	$M3$
O	–	√	×	×
$M1$	–	–	√	×
$M2$	–	–	–	×
$M3$	–	–	–	–

f_7	O	$M1$	$M2$	$M3$
O	–	×	×	√
$M1$	–	–	×	×
$M2$	–	–	–	√
$M3$	–	–	–	–

f_8	O	$M1$	$M2$	$M3$
O	–	×	×	×
$M1$	–	–	×	×
$M2$	–	–	–	×
$M3$	–	–	–	–

f_9	O	$M1$	$M2$	$M3$
O	–	×	×	√
$M1$	–	–	×	×
$M2$	–	–	–	√
$M3$	–	–	–	–

f_{10}	O	$M1$	$M2$	$M3$
O	–	√	×	×
$M1$	–	–	√	√
$M2$	–	–	–	×
$M3$	–	–	–	–

TABLE III: Holm-Bonferroni procedure for parametrization vector *C*=8 on 20D.

Despite the exciting discussion at the international genetic and evolutionary computation conference, GECCO 2021, leading to proposed modifications, the results are inconclusive. In many tested scenarios, the changes in SOMA-CLP algorithm do not offer a significant improvement to the overall performance. Furthermore, it can be concluded that the third modification, where all individuals migrate to one selected leader at the beginning of the migration, only compromises the performance of the original algorithm. The version referred to as modification 1, where a limited number of cluster leaders are chosen for the third phase of the algorithm, is more beneficial for the non-rotated shifted cases of the benchmark set. And modification 2, which uses a roulette selection process instead of the rank selection, showed improved performance for rotated shifted benchmark cases.

Therefore, it is safe to conclude that more research is needed, especially towards other possible changes in population dynamics and organization processes mechanisms. Thanks to the benchmark classification into four categories of test cases, it is possible for the researchers to track details in performance changes of algorithm variants for different test scenarios and if any modification introduces some bias, for example, towards non-shifted global optima. The future research will continue on a detailed investigation of the already implemented modifications, for example, their impact on the population diversity. Also, alterations with promising proposals to other phases of the algorithm will be investigated for the SOMA-CLP, emphasizing better robustness and performance improvement.

References

[1] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger, and S. Tiwari, "Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization," *KanGAL report*, vol. 2005005, no. 2005, p. 2005, 2005.

[2] A. Wagdy, A. A. Hadi, A. K. Mohamed, P. Agrawal, A. Kumar, and P. N. Suganthan, "Problem definitions and evaluation criteria for the cec 2021 special session and competition on single objective bound constrained numerical optimization." Technical Report, Nanyang Technological University, Singapore, 2020.

[3] K. Price, N. Awad, M. Ali, and P. Suganthan, "Problem definitions and evaluation criteria for the 100-digit challenge special session and competition on single objective numerical optimization," in *Technical Report*. Nanyang Technological University, 2018.

[4] T. Kadavy, M. Pluhacek, A. Viktorin, and R. Senkerik, "Self-organizing migrating algorithm with clustering-aided migration," in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, ser. GECCO '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1441–1447. [Online]. Available: https://doi.org/10.1145/3377929.3398129

[5] ——, "Self-organizing migrating algorithm with clustering-aided migration and adaptive perturbation vector control," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, ser. GECCO '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 1916–1922. [Online]. Available: https://doi.org/10.1145/3449726.3463212

[6] I. Zelinka, "Soma—self-organizing migrating algorithm," in *New optimization techniques in engineering*. Springer, 2004, pp. 167–217.

[7] ——, "Soma—self-organizing migrating algorithm," in *Self-Organizing Migrating Algorithm*. Springer, 2016, pp. 3–49.

[8] J. A. Hartigan, *Clustering algorithms*. John Wiley & Sons, Inc., 1975.

[9] Q. Diep, I. Zelinka, and S. Das, "Self-organizing migrating algorithm pareto," *MENDEL*, vol. 25, no. 1, pp. 111–120, Jun. 2019. [Online]. Available: https://mendel-journal.org/index.php/mendel/article/view/87

[10] Q. B. Diep, I. Zelinka, S. Das, and R. Senkerik, "Soma t3a for solving the 100-digit challenge," in *Swarm, Evolutionary, and Memetic Computing and Fuzzy and Neural Computing*, A. Zamuda, S. Das, P. N. Suganthan, and B. K. Panigrahi, Eds. Cham: Springer International Publishing, 2020, pp. 155–165.

[11] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[12] G. Zames, N. Ajlouni, N. Ajlouni, N. Ajlouni, J. Holland, W. Hills, and D. Goldberg, "Genetic algorithms in search, optimization and machine learning." *Information Technology Journal*, vol. 3, no. 1, pp. 301–302, 1981.

[13] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the american statistical association*, vol. 32, no. 200, pp. 675–701, 1937.

[14] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine learning research*, vol. 7, no. Jan, pp. 1–30, 2006.