Proceedings of the of the 17th Conference on Computer Science and Intelligence Systems pp. 291–300
DOI: 10.15439/2022F153
ISSN 2300-5963 ACSIS, Vol. 30

# GaMeDE2 — improved Gap–based Memetic Differential Evolution applied to Multimodal Optimization

Michał Antkiewicz, Paweł B. Myszkowski
Wrocław University of Science and Technology
Faculty of Information and Communication Technology
ul.Ignacego Łukasiewicza 5, 50-371 Wrocław, Poland
email: {michal.antkiewicz, pawel.myszkowski}@pwr.edu.pl

Maciej Laszczyk
email: maciej.laszczyk@gmail.com

*Abstract*—**This paper presents an improved Gap–based Memetic Differential Evolution (GaMeDE2), the modification of the GaMeDE method, which took second place in the GECCO 2020 Competition on Niching Methods for Multimodal Optimization. GaMeDE2 has reduced complexity, fewer parameters, redefined initialization, selection operator, and removed processing phases. The method is verified using standard benchmark function sets (classic ones and CEC2013) and a newly proposed benchmark set comprised of deceptive functions. A detailed comparison to state-of-the-art methods (like HVCMO and SDLCSDE) is presented, where the proposed GaMeDE2 outperforms or gives similar results to other methods. The document is concluded by discussing various insights on the problem instances and the methods created as a part of the research.**

## I. Introduction

MULTIMODAL Optimization (MMO) is a well-established problem in the literature (e.g.[1]). Due to its practical applications, it has been studied over the years. In real-world practical problems, the number of optimal solutions is not known *a priori*. It means, that valuable results might be ruled out by the assumption of sole optimal solution existence. As a result of real–life unpredictability and constraints, the optimal solution might not be feasible, and a suboptimal one may be preferred. What is more, the knowledge of other valuable solutions can support decision–making and allows for agile switches without the need for running the optimization process from the beginning with the risk of convergence to the same optima, which is the drawback of the standard unimodal optimization approaches.

The well-established competition in the multimodal optimization area is the annual GECCO Conference and Competition on Niching Methods for Multimodal Optimization based on the benchmark function suit [2]. GaMeDE2 proposed in this article is an extension of the GaMeDE [3] method, which took second place in the 2020 competition edition. The paper presents several modifications to boost the generality, which has been shown on the additional benchmark suits. Three other methods were selected to compare the final results. Self–adaptive Double-Layer-Clustering Speciation Differential Evolution (SDLCSDE) [4] is the recently published approach and gives very promising results. Unfortunately, it has not been evaluated on the GECCO Competition benchmark set, and the source code was not available to carry out the experiments - the results for another benchmark set of classical multimodal functions [5] have been used. The next reference method used is the Hill-Valley-Clustering-based VMO (HVcMO) [6], a novel method based on the HillVallEA [7] - a winning method in GECCO 2019 Competition on Niching Methods for Multimodal Optimization. The third compared method is the original GaMeDE approach.

The rest of the article is structured as follows. Section 2 covers the short literature review related to multimodal problems. The proposed GaMeDE2 method is introduced in section 3. The experimental setup, datasets descriptions, and results for the evaluated methods with the discussion of the results are given in Section 4. Lastly, the paper is concluded in section 5.

## II. Scientific background and related works

A series of articles show the importance of multimodal problems in multiple practical and various valuable areas, such as Drug Scheduling Problem, Protein Structure Prediction, Resource-Constrained Multiproject Scheduling Problem, cosmological applications, and an iconic machine-learning classification problem (e.g [8]) and many others. In literature, multimodality tends to be combined with multi-objectiveness, wherefore research in one area benefits both.

Besides real-world applications, sets of benchmark functions for the MMO have been proposed over the years. They are either manually fabricated to emphasise certain features or forged by combining multiple unimodal benchmark functions. Among the most recognizable in the literature [9] are multimodal benchmark functions based on: Rastrigin's function, Shubert's function, Vincent's function, Griewank's function, Schwefel's function, and Ackley's function. As a result of the number of combinations, a unified evaluation set has been introduced [2]. It consists of highly varied functions in terms of number of dimensions (1-20), domain and peak height scale, number of local and global optima (0-many/1-216), optima distribution, and landscape variability. Additionally, it contains a single instance of a deceptive function - the Five-Uneven-Peak Trap. This set has been used in the GECCO 2020 Niching Competition on Multimodal Optimization and the following editions. The alternative function composition has been proposed in [10] which shows the flexibility of combining the functions. Another benchmark set grounded in the literature [5] introduces new instances related to the competition set and modifies the budget for the subset of those already present. These functions are divided into six categories: Deceptive Functions, 1D Multimodal Functions, 2D Multimodal Functions, $>=$ 2D More Challenging Multimodal Functions, Inverted Rastrigin Function, and Generic Hump Functions. The added deceptive/trap functions are: 1D Two-Peak Trap and 1D Central Two-Peak Trap. The deceptiveness factor is an essential aspect of Multimodal Optimization [11]. There are limited works related to high dimensional multimodal trap functions. The one proposed approach is to apply function composition [12], yet it has not been openly adapted as the benchmark function, nor extended research in the area has been found at the moment of writing this article.

As stated in the introduction, a diverse set of valuable solutions is expected when solving a multimodal problem. For this reason, an efficient exploratory method is crucial. Evolutionary Computation is known to be effective in complex multimodal optimization, e.g. [1][13][12][14]. A significant amount of attention has been paid to Continues Multimodal Optimization, where Differential Evolution (DE) [15] is a reference method. It is still a widely used approach in the literature. Work [16] introduces a Dual-Strategy Mutation, adaptive individual selection, and converged individuals archive. Authors in [17] proposed a novel mutation strategy based on the Local Binary Pattern used for niche detection. Another modification was introduced in [18], where Distributed Individual for Multiple Peaks (DIMP) has been used to track optima. It has been extended by adding two novel mechanisms. First, Lifetime Mechanism is inspired by the natural phenomenon of organism aging and limited lifespan. Second, Elite Learning Mechanism (ELM) is introduced to refine the accuracy and efficiency of the archiving mechanism.

High-quality solutions in the multimodal landscape may often be found in different parts of the search space, complicating the single population convergence. Niching's [8] technique was introduced to prevent this effect by dividing the domain into multiple subsets called niches. The general idea in multimodal optimization is to detect niches, ideally located around the optima, and explore them separately. Niching is a widely applied concept in MMO and several modifications can be found in the literature. Nearest–neighbor niching introduced in [19] aids in achieving a well-balanced species. Another approach to increasing the solutions' (swarm) diversity is introducing the Equilibrium Factor to modify the individual's velocity [20]. Parameter–less niching mechanism (affinity propagation clustering) [21] is a valuable direction that reduces the method's parameter number and helps to locate the nearest peak, which boosts the convergence. Double-layer-clustering [4] has been proposed to increase the diversity and global exploration to locate more global optima. The DE method is applied on the niche level to support escaping local optima and detecting new promising areas in the search space. It also benefits from a self-adaptive strategy to reduce the number of parameters by self-adapting the crossover probability and scaling factor used by DE. On the other hand, it still requires the population size defined per problem instance. Hill-VallEA [22] is the GECCO 2019 Niching Competition on Multimodal Optimization competition winner. It introduced the Hill-Valley Clustering approach to adaptively cluster the search space in niches residing around a single optimum. It utilizes the Hill-Valley test [23] to determine whether two solutions belong to the same niche. Combined with a core search algorithm to optimize the niches and restart scheme, it outperformed its competitors. Hill-Valley-Clustering-based VMO (HVcMO) [6] merges the HillVallEA method with Variable Mesh Optimization [24], which significantly boosted the optima detection and improved the efficient use of budget.

The original GaMeDE [3] is a novel method, drawing concepts from the MMO using DE as its base. It benefits from GAP selection (and archive management mechanism) to keep high diversity, clustering for identification of promising areas, and local search optimization. Its core functionality is a two-phase approach - the $WIDE$ phase uses standard random selection followed by the HillValley Clustering to split the population into niches. Each niche is further optimized by the AMaLGaM Univariate[25] local optimizer, and the best individual per cluster is stored in the archive. $FOCUS$ phase uses two selections alternately - standard tournament selection and GAP selection. The latter is a novel approach proposed by the authors. It follows the idea of tournament selection, but instead of fitness (or objective function value), it uses a 'gap' distance. The 'gap' distance is simply the Euclidean distance to the nearest existing individual in the decision space. Its goal is to guide the exploration of 'blank spaces' of the landscape. The last step of the $FOCUS$ phase is the HillClimber local optimization of new points in the archive. The method starts with the $FOCUS$ phase and switches to the $WIDE$ when no longer can find new optima.

## III. Proposed method

GaMeDE2 is a redesigned GaMeDE [3] method developed for the GECCO 2020 Competition on Niching Methods for MMO. It means a high probability of overtunning and overfitting for the competition benchmark suite. Experiments conducted using new instances presented a lack of generalization and exposed the need for improvements in this area. Mentioned results are shown in the experiments chapter. The goal of modifications introduced in this work is to simplify the original GaMeDE algorithm and improve its effectiveness across multiple benchmark sets. The main objective of the MMO aspects remains unchanged. In this article, it is defined as the search for global optima only (in contrast to the search for all, including local, optima), which is a common approach in literature [3], [4], [25]. Each multimodal problem instance

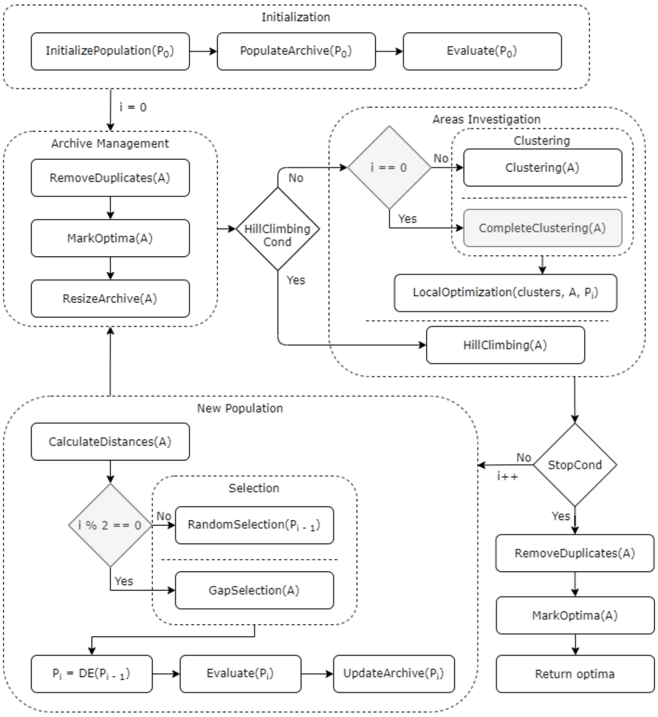


Fig. 1. A general schema of GaMeDE2

is being solved independently with an objective function described as:

$$\max_{x} f(x) = y, \quad x \in R^d \tag{1}$$

The aim is to find as many different $x$ vectors as possible where the minimum distance between two optima is given.

In this section, GaMeDE2 with a set of introduced modifications has been described. The Algorithm 1 presents the complete version of the GaMeDE2 pseudo-code, while Figure 1 illustrates the modifications described below.

GaMeDE2 is based mainly on a DE that explores/exploits new areas in the problem landscape. DE is steered by the archive (concept strictly incorporated from multi-objective problems), which stores candidates for global optima, and concentrates on 'gaps' between already found solutions. DE optimization is additionally boosted by a clustering mechanism to identify promising areas. Subsequently, local search speeds up the convergence in those areas. The $HillClimbing$ procedure further optimizes local optima.

GaMeDE2 initialization (see The Algorithm 1 or/and Figure 1) starts with a random initialization as presented in pseudocode line 3, the initial population is evaluated and used to populate an archive. The archive is used to store the global optima and promising individuals. The $PopulateArchive$ method tries to add each individual to the archive. It utilizes the simplified HillValley Test[7] - creates the middle-point $M$ between the tested individual $A$ and his nearest neighbor $B$) in the archive. If the middle-point is worse than both points ($A$ and $B$), they are kept as there is a chance they come from two different niches. Otherwise, only the best individual ($A$, $B$, or $C$) is stored. In lines 6 and 12, distances between points in the archive are calculated and stored for future usage. In lines 7 and 26, $ArchiveManagement$ takes place. Its pseudocode is presented in Algorithm 2 and remained unchanged from the original GaMeDE version. It removes duplicates from the archive, marks current optima, and truncates the archive by sorting it and removing worst solutions until it fits the $MaxArchiveSize$.

**Algorithm 1** GaMeDE2 pseudocode

1: $PrevGenOptima, i \leftarrow 0$
2: $Optima \leftarrow \emptyset$
3: $P_i \leftarrow InitRandomPop()$
4: $Evaluate(P_i)$
5: $A \leftarrow PopulateArchive(P_i)$
6: $CalculateDistances(A)$
7: $ArchiveManagement(A, Optima)$
8: $Clusters \leftarrow CompleteClustering(A)$
9: $LocalOptimization(Clusters, A, P_i)$
10: **while** !$StopCondition()$ **do**
11:     $i$++;
12:     $CalculateDistances(A)$
13:     $P_i \leftarrow \emptyset$
14:     **while** $|P_i|$ != $|P_{i-1}|$ **do**
15:         **if** $i$ % 2 == 0 **then**
16:             $Parents \leftarrow GapSelection(A)$
17:         **else**
18:             $Parents \leftarrow RandomSelection(P_{i-1})$
19:         **end if**
20:         $Mutants \leftarrow Mutate(Parents)$
21:         $Children \leftarrow Crossover(Mutants)$
22:         $P_i \leftarrow P_i + Children$
23:         $Evaluate(P_i)$
24:         $A \leftarrow UpdateArchive(P_i, A)$
25:     **end while**
26:     $ArchiveManagement(A, Optima)$
27:     **if** $|Optima|$ - $PrevGenOptima > MinNewOptima$ **then**
28:         $HillClimbing(A)$
29:     **else**
30:         $Clusters \leftarrow Clustering(A)$
31:         $LocalOptimization(Clusters, A, P_i)$
32:     **end if**
33:     $PrevGenOptima \leftarrow |Optima|$
34: **end while**
35: $A \leftarrow RemoveDuplicates(A)$
36: $Optima \leftarrow MarkOptima(A)$
37: **Return:** $Optima$

Before the first loop starts, $CompleteClustering$ takes place (see line 8). It applies for an additional clustering pass as described in the modifications section. In line 9, $LocalOptimization$ utilizes AMaLGaM Univariate (as in the original GaMeDE). Lines 10-34 present the main loop, which runs until $StopCondition$ is fulfilled - in this case, until the budget (number of evaluated individuals) is fully used. A new, empty population is initialized in line 13 and populated in lines 14-25. It uses $GapSelection$ and $RandomSelection$ alternately without the need for having any phases as described in the modifications section.

Lines 20-23 present a standard DE process. New individuals are processed in twos. Each of the two individuals returned from the selection undergoes the basic DE mutation process (based on two different, randomly selected, individuals and factor F). Genes are truncated to make sure all individuals are feasible. In the end, Uniform Crossover is applied. In line 24, the archive is updated with the same procedure used in $PopulateArchive$. Lines 27-32 present the local optimization step, which takes place at the end of each algorithm iteration. The default path is to run $LocalOptimization$ for the promising clusters (see line 9). Alternative $HillClimbing$ method is used if many new optima candidates appear in this iteration.

**Algorithm 2** ArchiveManagement pseudocode

**Params:** $A, Optima$
$A \leftarrow RemoveDuplicates(A)$
$Optima \leftarrow MarkOptima(A)$
$A \leftarrow ResizeArchive(MaxArchiveSize, Optima, A)$

GaMeDE2 method is based on the original GaMeDE, however it comprises several modifications that reduce the complexity of the method.

### A. GaMeDE2 – major proposed modifications

GaMeDE2 consists of several modifications as follows:

1) **Skip first selection -** In the GaMeDE, the first selection took place right after the population initialization. Both selection and initialization in GaMeDE are random, which does not introduce any improvement other than increasing the initial pool of random search. However, the evaluation cost is doubled, which might significantly reduce the number of further evaluations for the low-budget problem instances.
2) **Remove phase switching -** Based on the conducted experiments, it is not necessary to switch between $WIDE$ and $FOCUS$ phases between generations. Local optimization performed by a $HillClimber$ stays crucial for a group of problems, but it does not have to be paired with global algorithm phases. The condition to run a $HillClimber$ in Area investigation step remains unchanged, but it is not propagated further to the next steps.
3) **Double initial clustering -** An alternative clustering has been proposed to be used in the initial generation. Its purpose is to detect all the 'easy' optima faster. In the base algorithm, points found in the archive are spread around the search space, and many lay in the same niche. While it might increase the diversity in population, it significantly increases the number of clusters to search and the chance of crossing over the given budget. In GaMeDE2, an alternative approach to cluster generation uses candidates from the archive focusing on new areas. Solutions are clustered using additional Hill-Valley Clustering described in [7] to further reduce the number of candidates from the same global optimum (as an attractor). The Hill-Valley Test itself is simplified by reducing the middle points count to one. Adding second clustering was inspired by the approach in [4], where performing another DE iteration, based on top of the seeds found in the first pass, appeared to be successful. However, in this work, only the clustering was repeated with no additional DE run.
4) **Selection type is not related to phase -** Results of experiments confirm that the selection phases do not have to be bound to the $WIDE/FOCUS$ phases. However, the experiments showed that it is still crucial to keep both $Random$ and $GAP$ selections. Those two selections are used alternately through the subsequent generations, which allows to fully drop the need for defining two phases. The experiments showed that such selection composition gives the best results: one ($Random$) provides high diversity, while the other ($GAP$) focuses on search in poorly explored areas.

Both methods GaMeDE and GaMeDE2 are verified using 3 benchmark datasets, and the results are compared to two state-of-the-art methods. The research results are presented in the next section.

## IV. Experiments

Modifications proposed in GaMeDE2 have been experimentally verified across three different test sets. Each problem instance has been evaluated, and results are compared using GaMeDE, GaMeDE2 and Hill-Valley-Clustering-based VMO (HVcMO) [6]. For the second test set, results have also been compared with the recently presented Double-layer-clustering (SDLCSDE) [4]. Unfortunately, SDLCSDE cannot be used as the reference method for all test sets – the source code was not available to perform the experiments.

### A. Setup

MMO aims to find as many global optima as possible in a budget defined per each problem instance. The only metric used is the **Peak Ratio** ($PR$) which is a fraction of the global optima detected. Thus, the **Success Rate** ($SR$) has been calculated as the number of runs with all optima detected divided by the number of all runs. To verify if the global optimum has been reached, accuracy $\epsilon = 10^{-5}$ has been selected, the same as in [7]. For the SDLCSDE, accuracy levels were different across the test set. To compare performance precisely, GaMeDE2 has been tested on the problem instances where SDLCSDE accuracy levels were higher than the standard.

Both methods (GaMeDE and GaMeDE2) include non-deterministic elements, and experiments were repeated 30 times on all problem instances to average the results. The Wilcoxon signed-rank test has been applied to verify statistical significance using averaged results. The key advantage of the GaMeDE and GaMeDE2 is their generality, which means they can be successfully applied to a set of different problem instances without any configuration changes. Therefore, for both methods, only a single configuration has been used – in contrast to SDLCSDE, where the 'Population size' parameter was manually selected per each instance – which is not efficient while solving new, unknown problem instances.

To tune GaMeDE2 and find its optimal configuration, 5-Level Taguchi [26] Parameter Design procedure has been used. A set of experiment configurations was generated using an orthogonal matrix, and each configuration was repeated 10 times. This procedure was further repeated for a subset of test functions. The parameters with the highest *Signal-to-Noise* change were fine-tuned first based on the average results. All the parameters have been processed in that manner, subsequently in the descending order of Signal-to-Noise change. Table I presents selected values used by GaMeDE2. The GaMeDE uses the configuration proposed in [3].

TABLE I
**GAMEDE2 – BEST FOUND CONFIGURATION**

| Parameter | Value |
|---|---|
| PopulationSize | $1000 * dim$ |
| TournamentSize | 10 |
| MaxArchiveSize | $25 * dim^2$ |
| DiversityPhaseMinNewOptima | 5 |
| LocalOptInitialStep | 0.01 |
| MutationProbability | 0.6 |
| CrossoverProbability | 0.2 |
| F | 0.01 |

All presented experiments were carried out on three test sets, consisted of multimodal real–value problems.

### *B. Test sets*

The key attributes of each instance are presented in tables (see. Tab.II, Tab.III, Tab.IV and Tab.X) – it shows the number of global and local optima, number of dimensions and fitness evaluation budget. Three test sets are used in research: **CEC2013**, **Classical** Functions and **Deceptive** Functions, presented respectively in the rest of this section.

First test set - **CEC2013 /** GECCO2020 presented in Table II is commonly used in literature benchmark (e.g. [6][3]) and the same set which has been used in GECCO 2020 Competition on Niching Methods for Multimodal Optimization. It contains a variety of functions with different properties such as: deceptiveness (F1), wide spread of global optima count (F3 vs F9), wide spread of local optima count (F4 vs F6), composite functions (CF11 - CF20), flat and steep niches (CF20).

The second test set, called **Classical** Functions presented in Table III is a benchmark described in [5] and the one that has been selected by the authors of SDLCSDE [4]. The test set contains a number of functions already introduced in the CEC2013 set, yet it significantly decreases the budget given for each instance. The only fully repeated entries are B13 and B14. It also introduces a few new variants: Two-Peak Trap, Central Two-Peak Trap, Decreasing Maxima, Uneven Maxima, and Shekel's Foxholes. Due to the small budget, this set is used to verify the efficiency of the algorithms.

The last test set (**Deceptive** Functions, see Table IV) has been proposed to explore further the aspect of resistance to deceptive traps, which is an essential aspect in the optimization area. It is based on the Classical Functions set, and it consists

TABLE II
**CEC2013 / GECCO2020 MULTIMODAL FUNCTION SET**

| # | Function Name | D | #GOPT | #LOPT | Budget |
|---|---|---|---|---|---|
| F1 | Five-Uneven-Peak Trap | 1 | 2 | 3 | 50K |
| F2 | Equal Maxima | 1 | 5 | 0 | 50K |
| F3 | Uneven Decreasing Maxima | 1 | 1 | 4 | 50K |
| F4 | Himmelblau | 2 | 4 | 0 | 50K |
| F5 | Six-Hump Camel Back | 2 | 2 | 5 | 50K |
| F6 | Shubert | 2 | 18 | many | 200K |
| F7 | Vincent | 2 | 36 | 0 | 200K |
| F8 | Shubert | 3 | 81 | many | 400K |
| F9 | Vincent | 3 | 216 | 0 | 400K |
| F10 | Modifier Rastrigin | 2 | 12 | 0 | 200K |
| CF11 | Composite Function 1 | 2 | 6 | many | 200K |
| CF12 | Composite Function 2 | 2 | 8 | many | 200K |
| CF13 | Composite Function 3 | 2 | 6 | many | 200K |
| CF14 | Composite Function 3 | 3 | 6 | many | 400K |
| CF15 | Composite Function 4 | 3 | 8 | many | 400K |
| CF16 | Composite Function 3 | 5 | 6 | many | 400K |
| CF17 | Composite Function 4 | 5 | 8 | many | 400K |
| CF18 | Composite Function 3 | 10 | 6 | many | 400K |
| CF19 | Composite Function 4 | 10 | 8 | many | 400K |
| CF20 | Composite Function 4 | 20 | 8 | many | 400K |

TABLE III
**CLASSICAL MULTIMODAL BENCHMARK FUNCTION SET**

| # | Function Name | D | #GOPT | #LOPT | Budget |
|---|---|---|---|---|---|
| B1 | Two-Peak Trap | 1 | 1 | 1 | 10K |
| B2 | Central Two-Peak Trap | 1 | 1 | 1 | 10K |
| B3 | Five-Uneven-Peak Trap | 1 | 2 | 3 | 10K |
| B4 | Equal Maxima | 1 | 5 | 0 | 10K |
| B5 | Decreasing Maxima | 1 | 1 | 4 | 10K |
| B6 | Uneven Maxima | 1 | 5 | 0 | 10K |
| B7 | Uneven Decreasing Maxima | 1 | 1 | 4 | 10K |
| B8 | Himmelblau | 2 | 4 | 0 | 10K |
| B9 | Six-Hump Camel Back | 2 | 2 | 2 | 10K |
| B10 | Shekel's Foxholes | 2 | 1 | 24 | 10K |
| B11 | Shubert | 2 | 18 | many | 100K |
| B12 | Vincent | 1 | 6 | 0 | 20K |
| B13 | Vincent | 2 | 36 | 0 | 200K |
| B14 | Vincent | 3 | 216 | 0 | 400K |

of three deceptive functions: Two-Peak Trap, Central Two-Peak Trap and Five-Uneven-Peak Trap. All have been expanded into the higher dimension number by using the simple formula:

$$y = \frac{\sum_{i=1}^{D} f(x_i)}{D} \tag{2}$$

The main difficulty introduced by those Deceptive Functions is a small niche area for the global optima and their location in the far 'corners' of the domain, where niches for the local optima are wide and located in the 'center' of the search space. Figure 2 illustrates selected function landscapes in 2D versions.

The Deceptive Functions test set has been evaluated using two budgets (see Table IV) for solving methods. The standard budget has been defined as more restrictive to create a challenge for the methods. However, the experiments have shown that so small number of births does not allow for convergence for any of the researched methods. Hence, in additional experiments extended budget has been used, where

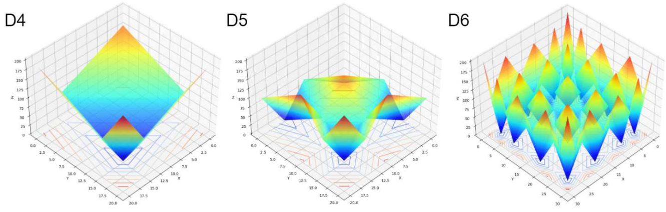


Fig. 2. **Deceptive functions visualization.** All three functions: Two-Peak Trap, Central Two-Peak Trap, Five-Uneven-Peak Trap in 2D variants.

TABLE IV
DECEPTIVE MULTIMODAL FUNCTION SET

| # | Function Name | D | #GOPT | #LOPT | Budget | Budget+ |
|---|---|---|---|---|---|---|
| D1 | Two-Peak | 1 | 1 | 1 | 10K | 10K |
| D2 | Central Two-Peak | 1 | 1 | 1 | 10K | 10K |
| D3 | Five-Uneven-Peak | 1 | 2 | 3 | 10K | 10K |
| D4 | Two-Peak | 2 | 1 | 3 | 20K | 40K |
| D5 | Central Two-Peak | 2 | 1 | 3 | 20K | 40K |
| D6 | Five-Uneven-Peak | 2 | 4 | 21 | 20K | 40K |
| D7 | Two-Peak | 3 | 1 | 7 | 40K | 200K |
| D8 | Central Two-Peak | 3 | 1 | 7 | 40K | 200K |
| D9 | Five-Uneven-Peak | 3 | 8 | 117 | 40K | 200K |

examined methods were given enough 'time' to converge and stabilize the results.

## C. Results

To measure the efficiency of the examined method, two standard measures are used – $PR$ and $SR$ (defined in the previous section). In Table V the values of $PR$ and $SR$ of the GaMeDE and modified algorithms for the CEC2013 set are given. It is worth mentioning that scores for GaMeDE do not exactly match those achieved in the GECCO competition. The results are slightly diverse because of the non-deterministic character of GaMeDE but are within one standard deviation.

However, the average $PR$ from all 20 problems maintains second place in the GECCO competition leader board. The GaMeDE2 managed to achieve comparable and even slightly better performance than GaMeDE. SDLCSDE algorithm results have not been found for this test set.

The values of $PR$ and $SR$ of all three methods for the Classical Functions set are given in Table VI. Results of GaMeDE and GaMeDE2 were calculated for the $\epsilon = 10^{-5}$ accuracy level, where those for SDLCSDE algorithm have been acquired from the original publication [4](see Table VIII) and with matching accuracy levels. Results show that for the functions B4, B5, B6, B7, B9, accuracy is higher (see Table VIII), but it is either the same or lower for the rest. To get more fair results, GaMeDE2 has been re-evaluated on functions B4–B9 with a matching accuracy which was presented in Table VII. Experiments on function 8 have been repeated due to lower performance than SDLCSDE while using higher accuracy. Thus, results show that GaMeDE2 achieves better results for B13 and B14 than SDLCSDE algorithm while maintaining a higher or the same accuracy level for all functions. Additionally, results show that GaMeDE struggles with a number of functions.

TABLE V
RESULTS FOR THE CEC2013 / GECCO2020 MULTIMODAL FUNCTION SET

| | GaMeDE | | GaMeDE2 | | HVCMO | |
|---|---|---|---|---|---|---|
| # | PR | SR | PR | SR | PR | SR |
| F1 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| F2 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| F3 | 0.967 | 0.967 | **1.000** | **1.000** | 1.000 | 1.000 |
| F4 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| F5 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| F6 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| F7 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| F8 | **1.000** | **1.000** | **1.000** | **1.000** | 0.967 | 0.000 |
| F9 | **1.000** | **1.000** | **1.000** | **1.000** | 0.937 | 0.000 |
| F10 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| CF11 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| CF12 | 0.983 | 0.867 | **1.000** | **1.000** | 1.000 | 1.000 |
| CF13 | 0.994 | 0.967 | **1.000** | **1.000** | 1.000 | 1.000 |
| CF14 | 0.806 | 0.033 | 0.761 | 0.033 | **0.861** | **0.267** |
| CF15 | 0.750 | 0.000 | 0.750 | 0.000 | 0.750 | 0.000 |
| CF16 | 0.667 | 0.000 | 0.667 | 0.000 | **0.689** | 0.000 |
| CF17 | 0.750 | 0.000 | 0.750 | 0.000 | 0.750 | 0.000 |
| CF18 | 0.667 | 0.000 | 0.667 | 0.000 | 0.667 | 0.000 |
| CF19 | 0.554 | 0.000 | **0.575** | 0.000 | 0.575 | 0.000 |
| CF20 | 0.496 | 0.000 | **0.500** | 0.000 | 0.488 | 0.000 |
| Avg | **0.882** | 0.642 | **0.883** | **0.652** | **0.884** | 0.563 |
| stat | = | | = | ++ | = | |

It is worth mentioning that GaMeDE2 results for the B8 instance have been achieved for the 0.00001 precision, while SDLCSDE has been evaluated for the 0.0005 precision. After re-evaluating (see Table VII) this instance on the same accuracy level, GaMeDE2 also achieved $PR = 1.0$ and $SR = 1.0$.

TABLE VI
RESULTS FOR THE CLASSICAL MULTIMODAL BENCHMARK FUNCTION SET.

| | GaMeDE [3] | | GaMeDE2 | | HVCMO [6] | | SDLCSDE [4] | |
|---|---|---|---|---|---|---|---|---|
| # | PR | SR | PR | SR | PR | SR | PR | SR |
| B1 | 0.933 | 0.933 | **1.000** | **1.000** | 1.000 | 1.000 | 1.000 | 1.000 |
| B2 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| B3 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| B4 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| B5 | 0.900 | 0.900 | **1.000** | **1.000** | 1.000 | 1.000 | 1.000 | 1.000 |
| B6 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| B7 | 0.833 | 0.833 | **1.000** | **1.000** | 1.000 | 1.000 | 1.000 | 1.000 |
| B8 | 0.433 | 0.000 | 0.992* | 0.967* | 1.000 | 1.000 | 1.000 | 1.000 |
| B9 | 0.550 | 0.100 | **1.000** | **1.000** | 1.000 | 1.000 | 1.000 | 1.000 |
| B10 | 0.067 | 0.067 | **1.000** | **1.000** | 0.967 | 0.967 | 1.000 | 1.000 |
| B11 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| B12 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| B13 | 1.000 | 1.000 | **1.000** | **1.000** | 1.000 | 1.000 | 0.999 | 0.940 |
| B14 | 1.000 | 1.000 | **1.000** | **1.000** | 0.938 | 0.000 | 0.889 | 0.000 |
| Avg | 0.837 | 0.774 | **1.000** | **1.000** | **0.993** | 0.926 | **0.992** | 0.924 |
| stat | | | = | ++ | = | | | |

Results of experiments in Table IX present GaMeDE, GaMeDE2 and HVCMO applications to the Deceptive Functions set. GaMeDE2 proved to achieve better results for every function except for D2 and D3 where both algorithms found all the solutions. Due to the novelty (and lack of SDLCSDE code) of this set there are no results for SDLCSDE.

In the methods evaluation process and experiments, there are some suggestions that for the Deceptive Multimodal Function Set it is worth extending the budget for several functions

TABLE VII
RESULTS FOR CLASSICAL MULTIMODAL BENCHMARK FUNCTION SET – REEVALUATION

| # | Accuracy | PR | SR |
|---|---|---|---|
| B4 | 0.000001 | 1.000 | 1.000 |
| B5 | 0.000001 | 1.000 | 1.000 |
| B6 | 0.000001 | 1.000 | 1.000 |
| B7 | 0.000001 | 1.000 | 1.000 |
| B8 | 0.0005 | 1.000 | 1.000 |
| B9 | 0.000001 | 1.000 | 1.000 |

TABLE VIII
ACCURACY LEVELS FOR SDLCSDE [4]

| # | Accuracy |
|---|---|
| B1 | 0.05 |
| B2 | 0.05 |
| B3 | 0.05 |
| B4 | 0.000001 |
| B5 | 0.000001 |
| B6 | 0.000001 |
| B7 | 0.000001 |
| B8 | 0.0005 |
| B9 | 0.000001 |
| B10 | 0.00001 |
| B11 | 0.05 |
| B12 | 0.0001 |
| B13 | 0.001 |
| B14 | 0.001 |

TABLE IX
RESULTS FOR THE DECEPTIVE MULTIMODAL FUNCTION SET – STD. BUDGETS

| | GaMeDE | | GaMeDE2 | | HVCMO | |
|---|---|---|---|---|---|---|
| # | PR | SR | PR | SR | PR | SR |
| D1 | 0.900 | 0.900 | **1.000** | **1.000** | 1.000 | 1.000 |
| D2 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| D3 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| D4 | 0.067 | 0.067 | **1.000** | **1.000** | 1.000 | 1.000 |
| D5 | 0.167 | 0.167 | **1.000** | **1.000** | 1.000 | 1.000 |
| D6 | 0.642 | 0.133 | 0.867 | 0.600 | **0.992** | **0.967** |
| D7 | 0.333 | 0.333 | **1.000** | **1.000** | 1.000 | 1.000 |
| D8 | 0.133 | 0.133 | **1.000** | **1.000** | 1.000 | 1.000 |
| D9 | 0.238 | 0.000 | 0.267 | 0.000 | **0.479** | **0.033** |
| Avg | 0.498 | 0.415 | 0.904 | 0.844 | **0.941** | **0.889** |
| stat | | | | | ++ | ++ |

TABLE X
RESULTS FOR THE DECEPTIVE MULTIMODAL FUNCTION SET – EXT. BUDGETS

| | GaMeDE | | GaMeDE2 | | HVCMO | |
|---|---|---|---|---|---|---|
| # | PR | SR | PR | SR | PR | SR |
| D1 | 0.900 | 0.900 | **1.000** | **1.000** | 1.000 | 1.000 |
| D2 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| D3 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| D4 | 0.100 | 0.100 | **1.000** | **1.000** | 1.000 | 1.000 |
| D5 | 0.367 | 0.367 | **1.000** | **1.000** | 1.000 | 1.000 |
| D6 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| D7 | 0.667 | 0.667 | **1.000** | **1.000** | 1.000 | 1.000 |
| D8 | 0.567 | 0.567 | **1.000** | **1.000** | 1.000 | 1.000 |
| D9 | 0.554 | 0.233 | **1.000** | **1.000** | 0.996 | 0.967 |
| Avg | 0.684 | 0.648 | **1.000** | **1.000** | **1.000** | **0.996** |
| stat | | | = | = | = | = |

(especially for D7-D9) to explore the three-dimensional landscape more extensively. The standard and extended budgets are presented in Table IV. In Table X the summary of results for examined methods is presented.

Table X includes results with extended budgets for all examined methods. It can be concluded that each method uses the extended budget effectively and achieves better results. However, GaMeDE2 is the most effective: $PR = 1.0$ and $SR = 1.0$, as it solves each function and outperforms other methods.

To summarise, the results for three test sets: CEC2013, Classic and Deceptive are presented in Table XI. Presented data show the average values for $PR$ and $SR$ for four examined methods: GaMeDE, GaMeDE2, HVCMO and SDLCSDE. Results show that GaMeDE2 outperforms referenced methods.

The results presented in this section contain four methods and three test sets. It gives a rather bird's eye view of examined methods. More detailed results, analysis and discussion, are included in the next section.

### D. Discussion

The GaMeDE2 maintained the average $PR = 0.883$ ($W = 6 > W_{\rho<0.05}$) on the CEC2013 (see Table V) test set but improved the $SR$ for F3, F12, F13. CF14 is the only instance where $PR$ value decreased, but the improvement on the remaining ones has balanced it. It was not expected to observe any significant change in this benchmark set used for the original GaMeDE method development. On the other hand, it is a crucial result suggesting that proposed modifications do not harm any original components. Difference between GaMeDE2 and HVCMO $PR = 0.884$ is not a significant difference either ($W = 7 > W_{\rho<0.05}$). In comparison to the HVCMO method, the key advantage of the GaMeDE2 method is a $SR$ difference, especially for functions F8 and F9, where it found all optima in every run, while HVCMO failed to do so even once. These two functions' main feature is the uneven distribution of high optima number (same as B13 - see Figure 5). Later in this chapter, it is explained which element is responsible for this improvement.

The second test set, containing Classical Multimodal Benchmark Functions (see Table VI), introduces novel instances for the GaMeDE. While some of the functions are repeated, they have narrowed the evaluation budget. Plots in Figure 3 visualize that selection in the first iteration doubles the initial evaluation cost, which is the 80% for both B8 and B10. With only 20% budget left, it is unlikely for the algorithm to locate all global optima. Skipping the first selection proposed as the first modification reduced the initial cost to 40% of the total budget.

Moreover, to fully take advantage of the extended evaluation budget, the clustering procedure in GaMeDE2 has been replaced with the initial iteration. In GaMeDE, the current population is clustered around the set of archived points. Those archive points are selected to reward unexplored areas. It is a valuable mechanism. However, there is no application in the first iteration, where the whole area is unexplored. The

TABLE XI
SUMMARY RESULTS FOR CEC2013, CLASSIC AND DECEPTIVE SETS

| | GaMeDE | | GaMeDE2 | | HVCMO | | SDLCSDE | |
|---|---|---|---|---|---|---|---|---|
| Set | PR | SR | PR | SR | PR | SR | PR | SR |
| CEC2013 | **0.882** | 0.642 | **0.883** | **0.652** | **0.884** | 0.563 | - | - |
| Classic | 0.837 | 0.774 | **1.000** | **1.000** | **0.993** | 0.926 | **0.992** | 0.924 |
| Deceptive (std) | 0.498 | 0.415 | 0.904 | 0.844 | **0.941** | **0.889** | - | - |
| Deceptive (ext) | 0.684 | 0.648 | **1.000** | **1.000** | **1.000** | **0.996** | - | - |

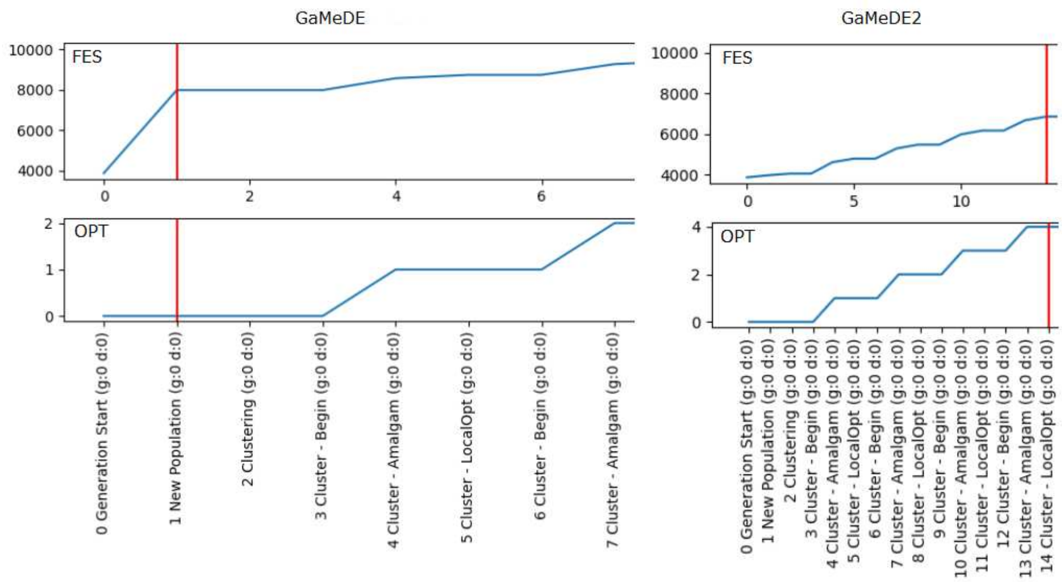


Fig. 3. **Modification #1 improvement.** Budget used (FES) and Optima found (OPT) for B8 problem instance before and after skipping selection in initial iteration. Similar effect can be observed for the instance B10.

alternative version introduces additional clustering of archive points just before the current population is clustered around them. It allows for filtering out archive points if there is a probability that they lay in the same niche. The modification results are presented on the Figure 4.

While having the explicit $WIDE$ and $FOCUS$ phases is not crucial for GaMeDE2, it is still important to maintain both types of selection: $GAP$ and $Random$. The important difference is that the former is archive-based, while the latter is population-based. Based on the experiments, both selections proved to be crucial. Using solely $Random$ Selection gives comparable results for most of the instances, except for those with a high number of optima (F8, F9, B13, B14), where method struggled with finding the narrow optima in far corners of the search space. On the contrary, while using $GAP$ selection only, all the optima in mentioned instances have been found. Figure 5 presents the difference in population distribution after using $Random$ or $GAP$ selection. A drawback of this approach has been however observed in decreased $PR$ for high-dimensional instances (CF19, see Table V). It confirms that using both alternately gives the best results. Further research could support further simplification of the method by limiting to $GAP$ selection only. Statistical tests for the Classical Functions Set (see Table VI) confirm all these changes introduce a significant improvement of $PR = 1.0$ ($W = 0 <= W_{\rho<0.05}$) and $SR = 1.0$ ($W = 0 <= W_{\rho<0.05}$) over the original GaMeDE method. It allowed achieving similar effectiveness to the HVCMO method.

The proposed Deceptive Multimodal Function Set introduces Trap functions in two- and three- dimensions. The essential difficulty is the moderate budget scaling which creates a challenge in three-dimension variants. Another difficulty is the very steep global optima in the far corners of the search space opposite the large area of deceptive local optima. The GaMeDE struggles with those functions because a significant fraction of the archive points lies on the local optima, and there is a bigger chance of selecting them for optimization. GaMeDE2 has been statistically verified to improve the $PR = 1.0$ ($W = 0 <= W_{\rho<0.05}$) and $SR = 1.0$ ($W = 0 <= W_{\rho<0.05}$) in relation to the original GaMeDE method for both budget sets (see Table IX and Table X).

## V. CONCLUSIONS AND FUTURE WORK

Developing a method for a specific benchmark suite allows to focus on the improvements and quickly verify their effectiveness. However, it may lead to overfitting of the proposed solution. The original GaMeDE could be such a case. While very competitive on the CEC2013 benchmarks,

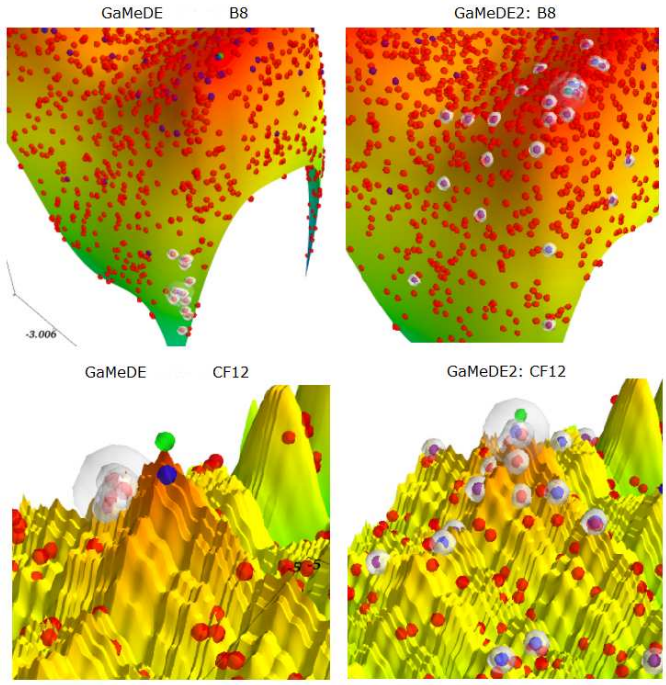


Fig. 4. **Modification #3 improvement.** After additional archive points clustering, the actual clusters (marked by grey spheres) and located around the global optima (green spheres) instead of on their side.

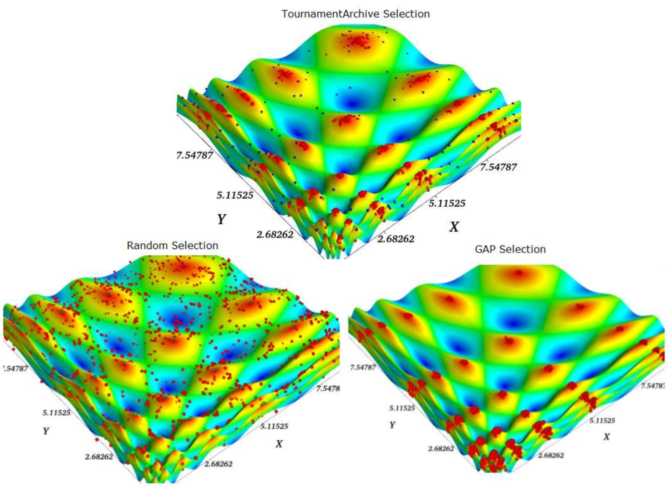


Fig. 5. **Random and GAP selection effects (B13 func.)** population distribution (red spheres) after Random or GAP selection

it does not maintain the effectiveness on novel instances. Two additional test sets have been proposed to verify the method's generalization ability. First, two deceptive functions (Two-Peak Trap, Central Two-Peak Trap), Decreasing Maxima, Uneven Maxima, and Shekel's Foxholes have been added. Additionally, the computational budget for the number of already present functions has been limited. The second proposed test set further explores the range of the deceptive function – it expands three trap functions into 2 and 3 dimensions. The Hill-Valley-Clustering-based VMO (HVcMO), a novel solution based on the HillVallEA, has been selected to provide a fair comparison.

In tuning procedure GaMeDE2, the Taguchi parameter design procedure has been used to evaluate the GaMeDE parameters and fine-tune them across all three sets. While it provided a promising outcome for a single instance, it could not point a one configuration valid for all instances of the test suites. Such an outcome is that specific instances introduce different challenges, sometimes overlapping each other. For example, Vincent function has strongly irregularly distributed yet smooth optima. Composition Functions such as CF17/CF20 consist of two steep optima. Shekel's Foxholes is a nearly binary landscape with a minimal variety among the optima. Furthermore, there is a wide span of provided budget, even for the same functions: 50K evaluations for Himmelblau in F4 and only 10K in B8. Decreasing the budget might expose the weak spots of methods that otherwise work successfully. Fine-tuning parameters per instance ensures the best results yet requires far more studies of the problem and time. However, the number of optima, their distribution, and local landscape disturbance have not been known *a priori*. For all those reasons, having a single configuration could be a superior approach, and optimization methods that do not require tuning many parameters are far more practical.

Based on the results of the experiments, a set of changes has been proposed to improve the original method's generality, leading to better results for the two novel test sets. Moreover, it maintains the CEC2013 benchmark functions' competitive level while using just a single parameter configuration. In comparison to another state-of-the-art method – HVCMO, GaMeDE2 manages to completely solve instances with a high number of unevenly distributed optima (F8, F9, B14), while maintaining the average $PR$ at the same level. The only case where introduces method has lower effectiveness is a novel Deceptive Functions Set if a very restrictive budget is provided.

The proposed GaMeDE2 method mainly addresses the initialization and clustering process. Skipping the initial mutation frees a significant amount of budget, which prevents premature algorithm stopping for the low-budget instances. The alternative (double) clustering in the first iteration allows for faster exploring all the promising niches. While it has low chances of finding narrow optima, it marks the 'easy' ones. The idea is similar to the mechanism of two natural metabolism phases (*Anaerobic* and *Aerobic*) in the human body. The first is used in short, burst activity (a big number of 'easy' optima). The second can be for long-duration activities and far goals (narrow, 'hard' steep optima). Additionally, GaMeDE2 is further simplified by removing the selection dependency from the phase.

The proposed Deceptive Functions set illustrates that Trap functions are not a bigger challenge for the multimodal solving methods. Multimodal optimization, by definition, does not seek a sole solution, which makes it more resistant to deceptiveness. Further research on the more irregular high-dimensional composition of different Trap Functions could be valuable for further research direction.

Though GaMeDE2 uses a single configuration, its further version could also benefit from adaptive parameters steering, such as population size, archive size, mutation or crossover probability. Moreover, simplifying the original method, e.g. the parameter used in enabling $HillClimbing$ step was not fully eliminated – it requires further work to make it fully adaptive. Indeed, improvements in clustering could be a promising research area. At the current state, the first iteration allows searching all the promising niches, leading to extensive use of the evaluation budget. An efficient mechanism in balancing and prioritizing the clusters to explore could introduce a significant value.

## REFERENCES


[1] S. Das, S. Maity, B.-Y. Qu, P. N. Suganthan, Real-parameter evolutionary multimodal optimization—a survey of the state-of-the-art, Swarm and Evolutionary Computation 1 (2) (2011) 71–88.

[2] X. Li, A. Engelbrecht, M. G. Epitropakis, Benchmark functions for cec'2013 special session and competition on niching methods for multimodal function optimization, RMIT University, Evolutionary Computation and Machine Learning Group, Australia, Tech. Rep.

[3] M. Laszczyk, P. B. Myszkowski, A gap–based memetic differential evolution (gamede) applied to multi–modal optimisation–using multi–objective optimization concepts, in: Intelligent Information and Database Systems: 13th Asian Conference, ACIIDS 2021, Phuket, Thailand, April 7–10, 2021, Proceedings 13, Springer International Publishing, 2021, pp. 211–223.

[4] Q. Liu, S. Du, B. J. van Wyk, Y. Sun, Double-layer-clustering differential evolution multimodal optimization by speciation and self-adaptive strategies, Information Sciences 545 (2021) 465–486.

[5] X. Li, Niching without niching parameters: particle swarm optimization using a ring topology, IEEE Transactions on Evolutionary Computation 14 (1) (2009) 150–169.

[6] R. Navarro, C. H. Kim, Niching multimodal landscapes faster yet effectively: Vmo and hillvallea benefit together, Mathematics 8 (5) (2020) 665.

[7] S. Maree, T. Alderliesten, D. Thierens, P. A. Bosman, Real-valued evolutionary multi-modal optimization driven by hill-valley clustering, in: Proceedings of the genetic and evolutionary computation conference, 2018, pp. 857–864.

[8] S. W. Mahfoud, Niching methods for genetic algorithms, Ph.D. thesis, University of Illinois at Urbana-Champaign (1995).

[9] J. Barrera, C. A. C. Coello, A Review of Particle Swarm Optimization Methods Used for Multimodal Optimization, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 9–37.

[10] B.-Y. Qu, P. N. Suganthan, Novel multimodal problems and differential evolution with ensemble of restricted tournament selection, in: IEEE Congress on Evolutionary Computation, IEEE, 2010, pp. 1–7.

[11] D. E. Goldberg, K. Deb, J. Horn, Massive multimodality, deception, and genetic algorithms., in: PPSN, Vol. 2, 1992.

[12] B.-Y. Qu, J. J. Liang, Z. Wang, Q. Chen, P. N. Suganthan, Novel benchmark functions for continuous multimodal optimization with comparative results, Swarm and Evolutionary Computation 26 (2016) 23–34.

[13] B.-Y. Qu, J. J. Liang, P. N. Suganthan, Niching particle swarm optimization with local search for multi-modal optimization, Information Sciences 197 (2012) 131–143.

[14] S. Yazdani, H. Nezamabadi-pour, S. Kamyab, A gravitational search algorithm for multimodal optimization, Swarm and Evolutionary Computation 14 (2014) 1–14.

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

[16] Z.-J. Wang, Z.-H. Zhan, Y. Lin, W.-J. Yu, H.-Q. Yuan, T.-L. Gu, S. Kwong, J. Zhang, Dual-strategy differential evolution with affinity propagation clustering for multimodal optimization problems, IEEE Transactions on Evolutionary Computation 22 (6) (2017) 894–908.

[17] H. Zhao, Z.-H. Zhan, Y. Lin, X. Chen, X.-N. Luo, J. Zhang, S. Kwong, J. Zhang, Local binary pattern-based adaptive differential evolution for multimodal optimization problems, IEEE transactions on cybernetics 50 (7) (2019) 3343–3357.

[18] Z.-G. Chen, Z.-H. Zhan, H. Wang, J. Zhang, Distributed individuals for multiple peaks: A novel differential evolution for multimodal optimization problems, IEEE Transactions on Evolutionary Computation 24 (4) (2019) 708–719.

[19] P. Haghbayan, H. Nezamabadi-Pour, S. Kamyab, A niche gsa method with nearest neighbor scheme for multimodal optimization, Swarm and evolutionary computation 35 (2017) 78–92.

[20] Y. Li, Y. Chen, J. Zhong, Z. Huang, Niching particle swarm optimization with equilibrium factor for multi-modal optimization, Information Sciences 494 (2019) 233–246.

[21] Z.-J. Wang, Z.-H. Zhan, Y. Lin, W.-J. Yu, H. Wang, S. Kwong, J. Zhang, Automatic niching differential evolution with contour prediction approach for multimodal optimization problems, IEEE Transactions on Evolutionary Computation 24 (1) (2019) 114–128.

[22] S. Maree, T. Alderliesten, P. A. Bosman, Benchmarking hillvallea for the gecco 2019 competition on multimodal optimization, arXiv preprint arXiv:1907.10988.

[23] R. K. Ursem, Multinational evolutionary algorithms, in: Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406), Vol. 3, IEEE, 1999, pp. 1633–1640.

[24] A. Puris, R. Bello, D. Molina, F. Herrera, Variable mesh optimization for continuous optimization problems, Soft Computing 16 (3) (2012) 511–525.

[25] P. A. Bosman, J. Grahl, D. Thierens, Benchmarking parameter-free amalgam on functions with and without noise, Evolutionary computation 21 (3) (2013) 445–469.

[26] V. N. Nair, B. Abraham, J. MacKay, G. Box, R. N. Kacker, T. J. Lorenzen, J. M. Lucas, R. H. Myers, G. G. Vining, J. A. Nelder, et al., Taguchi's parameter design: a panel discussion, Technometrics 34 (2) (1992) 127–161.