

A Novel Parametric Benchmark Generator for Dynamic Multimodal Optimization

Ali Ahrari ^{a,*},

Saber Elsayed ^a,Ruhul Sarker ^a,Daryl Essam ^a,Carlos A. Coello Coello ^b

^a *School of Engineering and Information Technology, University of New South Wales, ACT, Australia, (E-mail: {a.ahrari; s.elsayed; r.sarker; d.essam}@unsw.edu.au, aliahrari1983@gmail.com)*

^b *Departamento de Computación, Mexico, Mexico
Basque Center for Applied Mathematics (BCAM) & Ikerbasque, Spain (E-mail: ccoello@cs.cinvestav.mx)*

Abstract

In most existing studies on dynamic multimodal optimization (DMMO), numerical simulations have been performed using the Moving Peaks Benchmark (MPB), which is a two-decade-old test suite that cannot simulate some critical aspects of DMMO problems. This study proposes the Deterministic Distortion and Rotation Benchmark (DDRB), a method to generate deterministic DMMO test problems that can simulate more diverse types of challenges when compared to existing benchmark generators for DMMO. DDRB allows for controlling the intensity of each type of challenge independently, enabling the user to pinpoint the pros and cons of a DMMO method. DDRB first develops an existing approach for generation of static multimodal functions in which the difficulty of global optimization can be controlled. Then, it proposes a scaling function to dynamically change the relative distribution, shapes, and sizes of the basins. A deterministic technique to control the regularity of the pattern in the change is also proposed. Using these components, a parametric test suite consisting of ten test problems is developed for DMMO. Mean Robust Peak Ratio for measuring the performance of DMMO methods is formulated to overcome the sensitivity of the conventional peak ratio indicator to the predefined threshold and niche ra-

*Corresponding author

dius. Numerical results of a successful multimodal optimization method, when augmented with a simple strategy to utilize previous information, are provided on the proposed test problems in different scenarios with the aim of serving as a reference for future studies.

Keywords: Evolutionary algorithm, dynamic problem, performance indicator, robust peak ratio, niching, global optimization

1. Introduction

Dynamic optimization aims to find and track optimal solution(s) of problems that change over time. This change may originate from a variation in the decision parameters, problem objectives or constraints [1]. An optimization
5 method that can efficiently deal with the dynamic nature of these problems should be able to rapidly find the optimal solutions after the change. Since many real-world problems are dynamic, the field of dynamic optimization has gained much interest in the recent decade [1, 2] and many dynamic optimization methods have been applied to practical problems, such as optimal control
10 of time-varying systems [3, 4], mission planning [5], vehicle routing [6, 7], and scheduling [8].

Although most research in this field focuses on tracking only the best solution, the changes in the values of the optima requires tracking all the optima at the same time because a local optimum before the change can become the
15 global one after the change [9, 10]. This means that even if the changes are small, there can be a drastic change in the global optimum [11]. This challenge can be tackled by dynamic multimodal optimization (DMMO), which aims to detect and track multiple optima over time.

Some recent studies have defined the objective of DMMO as detecting and
20 tracking all the global optima in the problem [12, 13, 14], a definition that aligns more closely with the concept of multimodal optimization. From this perspective, DMMO is comparable to dynamic multiobjective optimization [15]: both aim to find and track a set of diverse optimal solutions. DMMO focuses

on diversity in the variable space whereas dynamic multiobjective optimization
25 focuses on diversity in the objective space.

Providing multiple best, or good, solutions is desired in many dynamic problems. An example is delivering supplies to multiple moving vehicles by a helicopter such that the total time for completing all the required deliveries or the fuel cost is minimized. The problem is dynamic since the vehicles may move.
30 There could be multiple best solutions for this problem (sequences for delivery), all of which may result in similar fuel consumption. In this case, the pilot may consider some preferences that were not considered in the simulation of the original problem, e.g., the current wind direction and velocity, the weight of the item(s) to be delivered to each vehicle, and the presence of potential risks in one
35 solution (e.g., passing by a mountain) to choose the best solution. Some other examples of practical DMMO problems, which have been explained in other publications, are finding the solutions to a time-dependent system of equations [16], dynamic tracking of multiple targets, and the dynamic multi-path routing problem [14].

40 The field of dynamic optimization heavily relies on numerical evaluation and comparison on test problems. For static optimization, test problems have undergone drastic changes from simple separable classical functions to rotated, skewed, ill-conditioned and composite functions in contemporary test suites for global [17] and multimodal [18, 19] optimization. A comprehensive test suite
45 for DMMO should be able to simulate all these challenges in addition to those particular to dynamic problems since DMMO is an extension to multimodal optimization.

Despite the importance of DMMO test problems for advancing this field, recent studies on this topic are scarce, at least when compared with recent publications on developing test suites for dynamic multiobjective optimization (e.g.
50 [20, 21]), and dynamic constrained optimization (e.g. [22, 23]). For two decades, the Moving Peaks Benchmark (MPB) [24] has been extensively employed by researchers to evaluate DMMO methods [10]. Although the formulation of MPB is simple, it relies on random numbers, which makes the generated test function

55 dependent on the available platform and software. Furthermore, there are some
challenging aspects of DMMO that cannot be simulated by MPB, which are
discussed in the next section.

This study proposes a benchmark generator, called Deterministic Distortion
and Rotation Benchmark (DDRB), for generating deterministic parametric test
60 functions for DMMO, in which each parameter controls the difficulty of one
class of challenges associated with DMMO. Such a control allows the user to
intensify or moderate each type of challenge to pinpoint the pros and cons of
an algorithm. The contributions of this paper are the following:

- It improves the procedure used in [19] such that a user-defined parameter
65 can control the difficulty of the resulting static multimodal problem.
- It proposes a time-dependent nonlinear scaling function to dynamically
modify the shapes, sizes and relative distribution of basins of a static
multimodal function.
- It introduces a simple technique to control the regularity of the change
70 pattern, a feature that may or may not exist in practical dynamic prob-
lems.
- It proposes the Robust Peak Ratio (RPR) indicator for multimodal opti-
mization as a robust alternative to the well-known Peak Ratio indicator.
RPR defines a non-binary success rule for basin detection and releases the
75 user from the burden of finding a proper value for the niche radius.
- It provides numerical results of a sample DMMO method on the proposed
test suite in different scenarios as a reference for future research.

The rest of this article is organized as follows: Section 2 reviews previous
related studies on benchmark generators for DMMO. Section 3 discusses some
80 desirable properties of a comprehensive test suite for DMMO. Section 4 develops
a set of parametric static multimodal functions with controllable degree of diffi-
culty. Section 5 proposes a procedure to simulate the dynamic behavior in these

functions and generates ten parametric DMMO problems. Section 6 proposes Mean Robust Peak Ratio to measure the performance of DMMO methods. Section 7 provides numerical results of a sample method on these ten test problems in different scenarios. Finally, conclusions are drawn in Section 8.

2. Previous Related Work

As discussed earlier, a DMMO method should be able to detect and track multiple optima over time. These methods employ a diversity preservation strategy, generally referred to as *niching*, to make parallel detection and tracking of optima possible. A popular strategy is the concept of *multi-population*, according to which the population is divided into multiple subpopulations, and each subpopulation is assigned a different task or should converge to a distinct optimum. Developed more than two decades ago (e.g. see [25, 11]), this strategy was later employed in many other methods by different groups of researchers [26, 27, 28, 29, 30, 31]

The performance of these methods has been generally evaluated according to the off-line error [32], which calculates the average fitness error over time. The fitness error is the difference between the best value and the global optimum value since the last change. This indicator thus favors methods with a fast initial progress rate. Alternatively, *accuracy* [33], which is also known as *best error before change* [1], considers the fitness error at the end of each time step (the immediate time before occurrence of a change). This performance indicator aligns more closely with practical situations, in which a period is allocated for the optimization process, and the best solution found at the end of this period is implemented within the implementing window [3, 34].

MPB, developed by Branke [24], is the most well-known benchmark generator for DMMO [35]. It defines a solution value as the maximum of several moving peaks with spherical basins whose locations, heights and widths change over time. The peak's height and weight changes slightly by adding a Gaussian random term to the previous values. The movement of a peak is determined

by a weighted combination of a random location and the previous movement. since its introduction in 1999 [24], this benchmark generator has been employed for numerical evaluation and performance comparison by several dynamic optimization methods based on metaheuristics such as particle swarm optimization [36, 37, 38] and differential evolution [12, 39, 40]. A similar idea has been followed in the DF1 generator [41] except for the valleys, which are cones whose slope gradually increases over time. However, DF1 has not enjoyed the popularity of MPB among researchers (see [42] for a comparison of the popularity of these methods).

Some variants of MPB were later developed for benchmarking specific aspects of DMMO methods. For instance, Li and Yang [27] developed a variant of MPB in which only a fraction of the peaks are allowed to change to benchmark efficacy of the change detection strategies. Modular variants of MPB [43, 10] were later developed for high-dimensional partially separable problems. The concept of MPB was also employed in [22] to create a problem generator for dynamic constrained optimization. Generalized Moving Peaks Benchmark (GMPB) [42] formulates a more general case of MPB in which the basins may be rotated ellipses, with non-symmetric basins and additional local minima. However, their formulation is much more complex than that of MPB and introduces several control parameters and complex equations to pose these irregularities.

Luo et al. [13] proposed a modification to MPB to tailor it for the situation in which the algorithms should find and track multiple equally good global maxima. Global maxima have fixed values, whereas local maxima change their values over time. A new rule was also defined so that the global maxima always lie sufficiently far from each other. For performance evaluation, they calculated the number of peaks identified at the end of each time step, and averaged it for all time steps. Their method could provide a reasonable outcome for relatively simple problems in low to moderate dimensionality.

The simplicity of the equations, scalability, and flexibility in the locations, widths and movements of maxima may explain the popularity of MPB among researchers; nevertheless, MPB has the following limitations:

- The basins are all spherical and remain spherical during the optimization process.
- 145 • The generated problems are random. This property results in further uncertainty when comparing two methods over different platforms/software and makes the reproducibility of results difficult.
- The global structure of the problem is always weak since the landscape consists of isolated maxima.
- 150 • Each solution evaluation requires N_{maxima} independent evaluations; therefore, the solution evaluation cost is high and proportional to the number of maxima (local and global). This may explain the low number of maxima (ten) which is commonly adopted when using MPB.

The Real Rotation Dynamic Benchmark Generator (RRDBG) [44] changes
155 the peaks' heights and widths dynamically, but it performs a time-dependent rotation of the search space to relocate the peaks. Like MPB, the peaks are formed using superposition of a convex function, although this function is exponential in RRDBG. In the Real Composition Dynamic Benchmark Generator (RCDBG) [44], more complex basic functions taken from the global optimization literature are used instead of only simple convex functions. Peaks move in
160 the search space using a time-dependent shift term. The computational time is proportional to the number of global maxima, which makes it more efficient than MPB and RRDBG, in which the computational cost is proportional to the overall number of local and global maxima. However, the size and the shapes
165 of the minima remain unchanged. This benchmark generator was also used in the CEC'2009 competition on dynamic optimization [45], which has been used in several subsequent studies for performance evaluation of dynamic optimization methods [39]. The rotation flexibility of RRDBG and the arbitrary basic function of RCDBG were combined in the benchmark generator developed in
170 [46].

3. Desired Properties of a DMMO Test suite

There are already well-developed artificial test suites for simple classes of optimization problems. These artificial test problems do not exactly represent practical problems but aim to simulate the challenges commonly found in them. It is expected that if a method can cope with these challenges, it will be able to optimize relevant practical problems efficiently. For example, for unconstrained single-objective global optimization, the BBOB'2009 test suite [17] can simulate the presence or absence of deceptive local minima, ill-conditioned landscapes, absence of global structure, and so on. We argue that since DMMO can be interpreted as the union of global optimization, multimodal optimization, and dynamic optimization, a robust DMMO benchmark generator should be able to simulate all, or at least most, of the well-known challenges associated with each of them. Therefore, this study distinguishes three types of challenges that a comprehensive DMMO test suite should be able to simulate:

- avoiding local optima and efficiently converging to the global optimum (global optimization)
- detecting distinct global optima (niching/multimodal optimization)
- tracking global optima over time (dynamic optimization)

Each challenge contributes to the overall hardness of a DMMO test problem. Besides, there should be a reasonable intensity of each type of challenge in the test problem. A test problem with many deceptive local optima might be a reasonable benchmark for global optimization since finding the global optimum is the only goal. This is not the case for a dynamic problem because a successful method should be able not only to detect the global optimum but also to track it over time. Similarly, we argue that the underlying static multimodal function should not be too hard; otherwise, the difficulty of the DMMO test problem predominantly originates from the static part. Such a problem overlooks the importance of the dynamic aspects of the problem, a shortcoming in some existing dynamic test problems [20]. Conversely, a test problem

200 with overly simplified static function (e.g., a function having identically-sized
regularly-distributed global optima with low condition number and no local optima)
underestimates the importance of the properties of the underlying static
function.

Consequently, there should be a comparable intensity of each type of chal-
205 lenge in a meaningful benchmark for DMMO. Ideally, the significance of each
type should be controllable so that the researchers can quickly identify the pros
and cons of a method in coping with each type of challenge. For example,
by intensifying the niching difficulty (e.g., by increasing the dissimilarity among
global optima or their relative distances), one can reliably analyze the capability
210 of a method in dealing with this specific type of challenge.

The proposed benchmark generator in this study aims to achieve these goals.
It consists of two separate modules:

- a static multimodal problem with multiple global minima. The difficulty
of global optimization is controllable by one tunable parameter.
- 215 • a dynamic mechanism which simulates dynamic behavior in the static
problem. This mechanism can control the maximal change in the sizes,
shapes, and relative distribution of global minima by a single tunable
parameter. It also controls the dynamic aspects of the problem such as
change severity, change frequency, and unpredictability in the change pat-
220 tern.

Without loss of generality, the objective in this method is minimization. The
following notation is used: bold upright for matrices and tensors, bold italics
for vectors and italics for scalars.

4. Composite Multimodal Test Problem

225 The procedure adopted for the generation of static MMO test problems
builds upon the one proposed in [19], in which a parametric composite function
 G is generated by a combination of three simple basic functions.

4.1. General Formulation

The static multimodal composite function has the following form:

$$G : \mathbb{R}^D \rightarrow \mathbb{R}; z = G(\mathbf{R}_0 \mathbf{X}, D_I, h_{GO}), \quad (1)$$

in which D is the problem's dimensionality, $\mathbf{X} \in \Omega$ is a solution, Ω is the search range, z is the function value, \mathbf{R}_0 is a rigid rotation matrix, and D_I is a parameter that controls the number of global minima in the problem (see [19], for details). h_{GO} is introduced in this study to control the difficulty of finding each global minimum.

To calculate the solution value, first, the solution \mathbf{X} is divided into two vectors, \mathbf{X}_I and \mathbf{X}_{II} :

$$\mathbf{X}_I = [x_1, x_1, \dots, x_{D_I}]^T, \quad \mathbf{X}_{II} = [x_{1+D_I}, x_{2+D_I}, \dots, x_D]^T. \quad (2)$$

These two vectors are further divided into sub-vectors of size d_i and d_{ii} , respectively. Then, the basic functions $g_I: \mathbb{R}^{d_i} \rightarrow \mathbb{R}_{\geq 0}$ and $g_{II}: \mathbb{R}^{d_{ii}} \rightarrow \mathbb{R}_{\geq 0}$ are applied to these sub-vectors which return vectors \mathbf{y}_I of size n_i and \mathbf{y}_{II} of size n_{ii} :

$$\mathbf{y}_I = \begin{bmatrix} g_I(\mathbf{x}_{i,1}) \\ g_I(\mathbf{x}_{i,2}) \\ \vdots \\ g_I(\mathbf{x}_{i,n_i}) \end{bmatrix}, \quad \mathbf{y}_{II} = \begin{bmatrix} g_{II}(\mathbf{x}_{ii,1}, h_{GO}) \\ g_{II}(\mathbf{x}_{ii,2}, h_{GO}) \\ \vdots \\ g_{II}(\mathbf{x}_{ii,n_{ii}}, h_{GO}) \end{bmatrix}. \quad (3)$$

Function g_I has multiple global minima whereas g_{II} has the single global minimum $\mathbf{x}_{ii}^* = \mathbf{0}$ with $g_{II}(\mathbf{x}_{ii}^*) = 0$. In g_{II} , the parameter $0 \leq h_{GO} \leq 1$ controls the difficulty associated with the global optimization of this function (e.g., depths of the local minima or the condition number of g_{II} , two of the main factors that affect the difficulty of global optimization). Finally, g_{III} functions on $\mathbf{y} = [\mathbf{y}_I^T, \mathbf{y}_{II}^T]^T$:

$$z = G(\mathbf{X}; D_I, h_{GO}) = g_{III}(\mathbf{y}) + c_0, \quad (4)$$

in which c_0 is a constant value. For $\mathbf{R}_0 = \mathbf{I}_D$ (\mathbf{I}_D is the identity matrix of size D), the number of global minima in the composite function is:

$$N_{\text{gmin}} = (n_{\text{gmin}})^{n_i}, \quad (5)$$

in which n_{gmin} is the number of global minima in g_{I} (see [19] for the proof).

235 The sufficient condition that employing an arbitrary \mathbf{R}_0 does not change the number of global minima is that all global minima are inside the hypersphere inscribed in the search space whose center lies on the center of the search space AND there is no other global minimum outside the search range. The global minima ($\mathbf{X}_i^*, i = 1, \dots, N_{\text{gmin}}$) can be calculated as follows:

$$\mathbf{X}^* = \mathbf{R}_0^T \times \begin{bmatrix} \mathbf{X}_{\text{I}}^* \\ \mathbf{X}_{\text{II}}^* \end{bmatrix}, \mathbf{X}_{\text{I}}^* = \begin{bmatrix} \mathbf{x}_{\text{I}}^* \\ \mathbf{x}_{\text{I}}^* \\ \vdots \\ \mathbf{x}_{\text{I}}^* \end{bmatrix}_{n_{\text{I}} \times 1}, \mathbf{X}_{\text{II}}^* = \begin{bmatrix} \mathbf{x}_{\text{II}}^* \\ \mathbf{x}_{\text{II}}^* \\ \vdots \\ \mathbf{x}_{\text{II}}^* \end{bmatrix}_{n_{\text{II}} \times 1} \quad (6)$$

240 in which \mathbf{x}_{I}^* and \mathbf{x}_{II}^* are the global minima of g_{I} and g_{II} , respectively.

A proper design of $g_{\text{II}}(\mathbf{x}, h_{\text{GO}})$, such that a greater h_{GO} makes the optimization of g_{II} harder, lets the user easily control the difficulty of global optimization of the composite function. This tunability substantially improves the utility of the test problem because it allows to benchmark a method on test problems
245 with similar niching but different global optimization hardness.

4.2. Suggested Static multimodal test functions

Table 1 summarizes five parametric composite functions developed using the improved procedure. The definitions of the basic functions used for generation of these composite functions are provided in supplementary material S1. These
250 basic functions are formulated by modification of simple existing test problems which can be found in [47, 48, 49].

The following procedure is employed in this study to generate a rotation matrix. For two orthogonal unit vectors \mathbf{u} and \mathbf{v} , the rigid rotation matrix \mathbf{R} that rotates the span of \mathbf{u} and \mathbf{v} by an angle of α can be calculated as follows [50]:

$$\mathbf{R}(\alpha) = \mathbf{I}_D + \sin(\alpha) (\mathbf{v}\mathbf{u}^T - \mathbf{u}\mathbf{v}^T) + (\cos(\alpha) - 1) (\mathbf{u}\mathbf{u}^T + \mathbf{v}\mathbf{v}^T) \quad (7)$$

Any \mathbf{R}_0 that is generated with $0 < \alpha < \pi/2$ can theoretically eliminate the separability in the objective function. A value far from the limits is preferable;

otherwise, the behavior of the rotated problem may still be close to a separable
 255 problem. In this study, the rotation matrix \mathbf{R}_0 is generated with $\alpha = \pi/6$.

For the sake of reproducibility, an array of 2000 random numbers sampled
 from a standard normal distribution are provided as a CSV-file in the supple-
 mentary material S2. These random numbers were generated using the com-
 mand `randn(2000, 1)` of MATLAB 2017b with the random seed #0 and a 10-digit
 260 decimal precision. The first D numbers form \mathbf{u}_0 , and $\mathbf{u} = \mathbf{u}_0/\|\mathbf{u}_0\|$. Similarly,
 the subsequent D numbers form \mathbf{v}_0 , $\mathbf{v}_0 \leftarrow \mathbf{v}_0 - (\mathbf{u}^\top \mathbf{v}_0)\mathbf{u}$, and $\mathbf{v} = \mathbf{v}_0/\|\mathbf{v}_0\|$.

The parameters of the suggested multimodal optimization test functions
 should be selected considering the additional complexity imposed by their dy-
 namic behavior. In particular, the parameter D_I controls the number of global
 265 minima. The recommended static test functions are presented in Table 2.

4.3. Effect of h_{GO}

To illustrate the effect of h_{GO} , we perform multimodal optimization of
 the suggested static test problems (Table 2) using the covariance matrix self-
 adaption evolution strategy with repelling subpopulation (RS-CMSA-ES) [51],
 270 which is one the most successful multimodal optimization methods currently
 available [19]. The number of subpopulations is set to 5, and the population
 size increase factor is set to 1.1. Other parameters are set to default values
 [51]. For this experiment, $D = 10$, the maximum evaluations budget ($maxFE$) is
 $20000DN_{\min}$, and the acceptable tolerance for the objective value is $\epsilon_f = 0.001$.

Table 1: Parametric static multimodal optimization test problems ($k, k' \in \mathbb{Z}_{\geq 0}$)

Function	g_I	g_{II}	g_{III}	c_0	D_I	d_I	d_{II}	D	N_{\min}
$G_{11}(\mathbf{X}, D_I, h_{GO})$	f_7	f_6	f_{15}	-49.2	$2k$	2	2	$2k+2k'$	4^k
$G_{12}(\mathbf{X}, D_I, h_{GO})$	f_{11}	f_8	f_{12}	-29.7	$2k$	2	2	$2k+2k'$	2^k
$G_{13}(\mathbf{X}, D_I, h_{GO})$	f_3	f_{13}	f_9	95.1	$2k$	2	2	$2k+2k'$	3^k
$G_{14}(\mathbf{X}, D_I, h_{GO})$	f_{10}	f_{14}	f_2	0	k	k	k'	$k+k'$	$k \times 3^k$
$G_{15}(\mathbf{X}, D_I, h_{GO})$	f_4	f_1	f_5	65.5	$2k$	2	2	$2k+2k'$	4^k

Table 2: Proposed static multimodal optimization test function to be used for the generation of dynamic problems. For all functions, the search range is $[-5, 5]^D$

PID	Function	D_I	N_{gmin}
1	$G_{11}(\mathbf{X}, D_I, h_{\text{GO}})$	2	4
2	$G_{12}(\mathbf{X}, D_I, h_{\text{GO}})$	2	2
3	$G_{13}(\mathbf{X}, D_I, h_{\text{GO}})$	2	3
4	$G_{14}(\mathbf{X}, D_I, h_{\text{GO}})$	1	3
5	$G_{15}(\mathbf{X}, D_I, h_{\text{GO}})$	2	4
6	$G_{11}(\mathbf{X}, D_I, h_{\text{GO}})$	4	16
7	$G_{12}(\mathbf{X}, D_I, h_{\text{GO}})$	6	8
8	$G_{13}(\mathbf{X}, D_I, h_{\text{GO}})$	4	9
9	$G_{14}(\mathbf{X}, D_I, h_{\text{GO}})$	2	18
10	$G_{15}(\mathbf{X}, D_I, h_{\text{GO}})$	4	16

275 Figure 1 shows the Mean Peak Ratio (MPR) [18], which is the fraction of the identified global minima, versus the number of evaluations divided by maxFE for different values of h_{GO} . The results are averaged over 20 independent runs for each setting. As can be observed:

- Reaching $\text{MPR} \approx 1$ takes a longer time for a greater h_{GO} . This means
280 that h_{GO} can successfully control the difficulty of the global optimization.
- All global minima can be found after a reasonably large number of function evaluations, which shows that for any value of $0 \leq h_{\text{GO}} \leq 1$, the static multimodal optimization is not so hard that some of the global minima are impossible to detect even when the evaluations budget is large.

285 5. Simulating Dynamic Behavior

This section presents the template for simulating the dynamic behavior in static functions. This template exploits three distinct techniques, which are

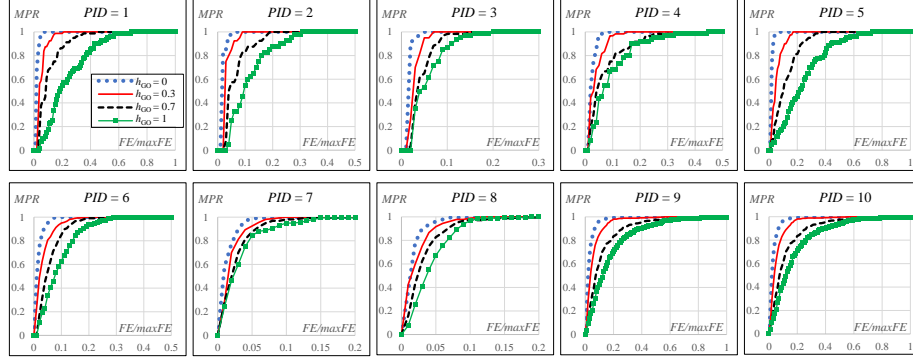


Figure 1: MPR as a function of the number of evaluations for 10 suggested multimodal optimization problems ($D = 10$) for different values of h_{GO} .

elaborated and formulated first. Subsequently, they are used in conjunction with the suggested static multimodal test problems to form DMMO problems.

290 5.1. General template

The general template of the proposed test problems is as follows:

$$g_d(\mathbf{X}, t) = g_0(\mathbf{R}(t) \mathbf{s}(\mathbf{X}, t)) + c(t), \quad (8)$$

In this equation:

- $g_d(\mathbf{X}, t)$ is the dynamic multimodal function
- $g_0(\mathbf{X})$ is a static multimodal function
- $\mathbf{R}(t)$ is a time-dependent rigid rotation matrix
- $\mathbf{s}(\mathbf{X}, t)$ is a time-dependent scaling function in decision variable space.
- $c(t)$ is a time-dependent constant

295

As observed, the dynamic behavior is simulated using three components: $\mathbf{R}(t)$, $\mathbf{s}(x, t)$ are applied to the solution space whereas $c(t)$ works on the objective space. If the global minima of $g_0(\mathbf{X})$ are known, the global minima of $g_d(\mathbf{X}, t)$ can be easily calculated using the following theorem:

300

Theorem: For any $\mathbf{X} \in \Omega$, there is the corresponding solution $\check{\mathbf{X}} = \mathbf{s}^{-1}(\mathbf{R}(t)^T \mathbf{X}, t)$ such that $g_d(\check{\mathbf{X}}, t) = g_0(\mathbf{X}) + c(t)$

Proof:

$$\begin{aligned} g_d(\check{\mathbf{X}}, t) &= g_d\left(\mathbf{s}^{-1}(\mathbf{R}(t)^T \mathbf{X}, t), t\right) + c(t) \\ &= g_0\left(\mathbf{R}(t) \mathbf{s}\left(\mathbf{s}^{-1}(\mathbf{R}(t)^T \mathbf{X}, t), t\right)\right) + c(t) \\ &= g_0(\mathbf{R}(t) \mathbf{R}(t)^T \mathbf{X}) + c(t) = g_0(\mathbf{X}) + c(t) \end{aligned}$$

Based on this theorem, if \mathbf{X}^* is a global minimum of $g_0(\mathbf{X})$, then $\check{\mathbf{X}}^* = \mathbf{s}^{-1}(\mathbf{R}(t)^T \mathbf{X}^*, t)$ is a global minimum of $g_d(\mathbf{X}, t)$.

if $\Omega = [-\omega, \omega]^D$, then for any $\mathbf{R}(t)$, all global minima will remain inside the search space after rotation if after scaling, they remain inside the sphere which is inscribed in the search space cube. If not, it is possible that some global minima leave the search space at specific time-steps. However, since after detection, tracking the global minima is the primary objective of DMMO, this requirement is enforced for the developed test problems in this study, and thus, the number of global minima remains unchanged.

5.2. Rigid Rotation

The time-dependent rotation matrix $\mathbf{R}(t)$ is generated using Eq. 7 with a time-dependent rotation angle of $\alpha(t)$. Through a gradual increase in $\alpha(t)$, the fitness landscape gradually rotates in the plane defined by \mathbf{u} and \mathbf{v} . This rotation only changes the location of global minima without changing their sizes, shapes, and relative distribution.

5.3. Time-dependent Nonlinear Scaling in Variable Space

Non-linear scaling of the search space distorts the shapes, sizes, locations, and relative distances of the global minima. In our case, this scaling operates on each dimension independently:

$$\mathbf{s}(\mathbf{X}, t) = [s(x_1, t), s(x_2, t), \dots, s(x_D, t)]^T, \quad (9)$$

320 in which $s(x, t): \mathbb{R} \times \mathbb{N} \rightarrow \mathbb{R}$ is a one-dimensional time-dependent scaling function. For simplicity, we assume that $\mathbf{X} \in [-1, 1]^D$ (if not, it can always be linearly scaled). The function $s(x, t)$ must satisfy the following conditions:

- $s(x, t)$ must be strictly monotonic with respect to x ; otherwise, it may introduce new global minima into the problem.
- 325 • $s(x, t)$ must be continuous to ensure that the number of global minima does not decrease after scaling the search space.
- $s(1, t) = 1$ and $s(-1, t) = -1$, so that if \mathbf{X} is inside/outside the search range, $s(\mathbf{X}, t)$ is also inside/outside the search range.
- For any x and t , we should have $0 < \frac{d}{dx}s(x, t) < \infty$; otherwise, the basin of
- 330 a global minimum may become too small or too large.

The scaling function suggested in this study is formulated following these requirements. First, two auxiliary functions $s_1(x)$ and $s_2(x)$ are defined to formulate the nonlinear component of the scaling function. These two functions are segments of two circles of radius $R = e_c^2 + (1 + e_c)^2$ whose centers lie on $[1 + e_c, -e_c]^T$ and $[-e_c, 1 + e_c]^T$, respectively (see Figure 2).

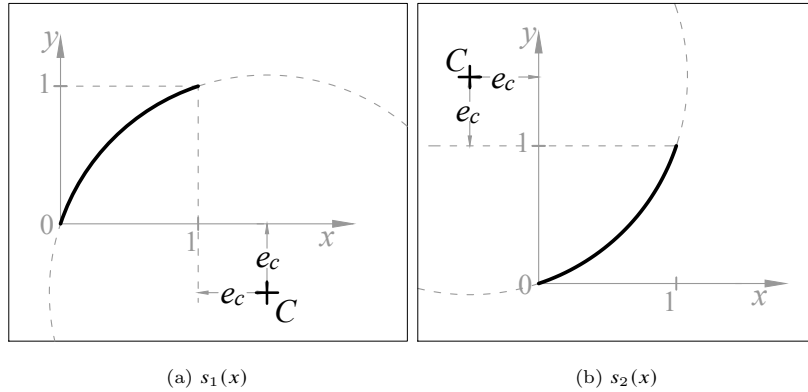


Figure 2: Auxiliary functions used in this study to generate nonlinearity in the scaling functions. The arcs in black solid lines represent the plots of these functions. The center of the arc is denoted by C .

The scaling function is now defined by performing time-dependent weighted average of $s_1(x)$ or $s_2(x)$ and x :

$$s(x, t) = \begin{cases} (1 - |w_s(t)|)x + |w_s(t)|s_1(|x|)\operatorname{sgn}(x) & \text{if } w_s(t) > 0 \\ (1 - |w_s(t)|)x + |w_s(t)|s_2(|x|)\operatorname{sgn}(x) & \text{otherwise} \end{cases} \quad (10)$$

Figure 3 shows the plots of $s(x, t)$ for selected values of $w_s(t)$. It is remarkable that the effect of the scaling function is the contraction of the fitness landscape wherever $\frac{d}{dx}s(x, t) > 1$ as well as the expansion of it wherever $\frac{d}{dx}s(x, t) < 1$.

The suggested $s(x, t)$ in this study has some interesting properties. Note that the maximum expansion/contraction occurs at $x = -1, 0, 1$ when $|w_s(t)| = 1$. Furthermore, $\frac{d}{dx}s(-1, t) = \frac{d}{dx}s(1, t)$, and besides:

$$\begin{aligned} \frac{d}{dx}s_1(0) &= \frac{1}{\frac{d}{dx}s_1(1)} = \frac{1+e_c}{e_c} \\ \frac{d}{dx}s_2(0) &= \frac{1}{\frac{d}{dx}s_2(1)} = \frac{e_c}{1+e_c} \end{aligned} \quad (11)$$

This means that the maximum expansion is equal to the maximum contraction

and:

- when $w_s(t) = 1$, the maximum expansion occurs at $x = \pm 1$ while the

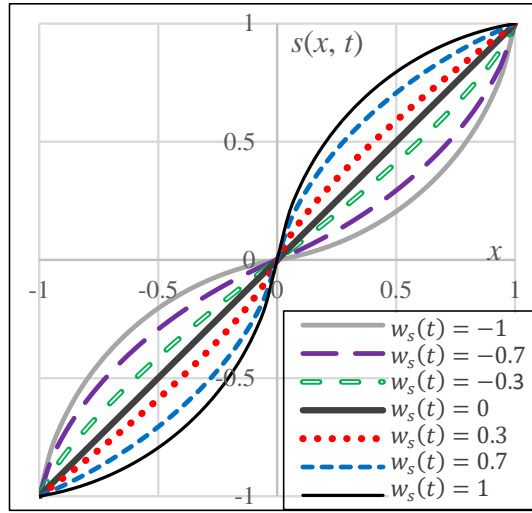


Figure 3: The suggested scaling function in this study ($s(x, t)$) for some selected values of $w_s(t)$

maximum contraction occurs at $x = 0$;

- when $w_s(t) = -1$, the maximum expansion occurs at $x = 0$ while the maximum contraction occurs at $x = \pm 1$;
- The non-linearity of the scaling function can easily be controlled by e_c . For $e_c \rightarrow \infty$, the non-linearity disappears, whereas for $e_c \rightarrow 0$ its effect becomes maximal.

Figure 4 illustrates the effect of the proposed scaling function, when $e_c = 0.5$, on the shapes, sizes, locations, and relative distribution of the global minima of the following static function:

$$g_0(\mathbf{x}) = - \sum_{i=1}^2 \cos(\pi x_i) \quad (12)$$

For $w_s(t) = 0$ (Figure 4(a)), the scaling function has no effect on the landscape since $s(x, t) = x$. As observed, the global minima are regularly distributed and have identically-sized spherical basins. For $w_s(t) = 0.5$, the scaling function shrinks the center and expands the regions close to the boundaries (Figure 4(b)). The impact of the scaling function intensifies for a greater $w_s(t)$ and it reaches its extreme when $w_s(t) = 1$ (Figure 4(c)). Now, the global minima have different shapes and sizes, and their relative distances have also reduced. An opposite effect can be observed for a negative $w_s(t)$. For example, for $w_s(t) = -1$ (the opposite extreme), the distance between the global minima has increased. Furthermore, the minimum in the center, which had the smallest basin for $w_s(t) = 1$, has now the largest basin. The shapes and sizes of other minima have changed as well. This figure clearly illustrates that the proposed scaling function dynamically changes shapes, sizes, locations, and distribution of the global minima, and thus, it can simulate some of the main features of DMMO.

5.4. Controllable Regularity of the Change Pattern

Change severity and change frequency are two features of dynamic test problems [52]. A comprehensive test suite for dynamic optimization should be able

365 to simulate these two features. Another important feature is the regularity of
the pattern in the change, which can be interpreted as the presence of some pat-
terns according to which global minima change over time. The changes may or
may not show such a pattern [53], [54], [20]. Therefore, a dynamic test problem
should allow for controlling the irregularity/randomness in the pattern of the
370 change.

In this study, we propose a deterministic method to control the regularity
of the change pattern. This method defines two distinct control parameters for

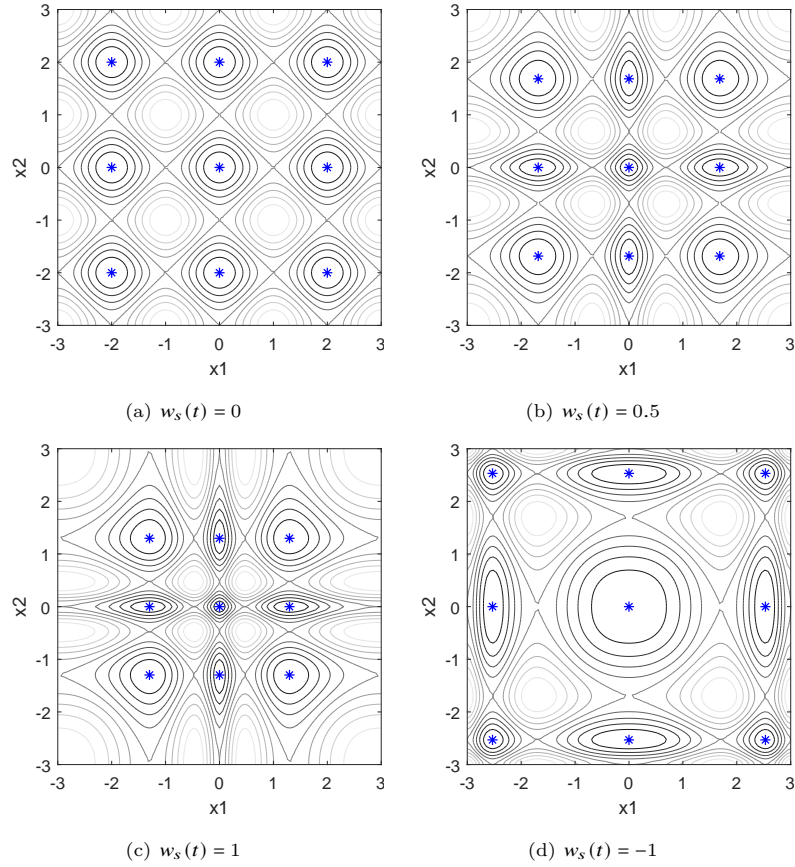


Figure 4: Landscape of a simple problem subject to the proposed dynamic scaling function at selected times when $e_c = 0.5$. Darker lines represent a lower function value. The blue asterisks represent the global minima.

controlling the change severity: $0 < n_{tr}$ determines the regular component of the change pattern whereas the irregular part is controlled by $0 < n_{ti}$. A greater value for each of these parameters means a less severe change. Besides, the intensity of each type of change is independent of the other type. For example, the problem may show a severe irregular change but a moderate regular change (a low n_{ti} but a high n_{tr}), or it may show significant regular and irregular changes at the same time (low values for both n_{ti} and n_{tr}). These two parameters are involved in the calculation of $\alpha(t)$, $w_s(t)$ and $c(t)$ as follows:

$$\begin{aligned}
\alpha(t) &= 2\pi \left(\frac{t_0}{n_{tr}} + \frac{\sin(t_0^2)}{n_{ti}} \right) \\
w_s(t) &= \sin \left(\frac{2\pi}{n_{tr}} t_0 + \frac{2\pi}{n_{ti}} \sin(t_0^2) \right) \\
c(t) &= 100 \sin \left(\frac{2\pi}{n_{tr}} t_0 + \frac{2\pi}{n_{ti}} \sin(t_0^2) \right) \\
t_0 &= t \bmod n_{tr}
\end{aligned} \tag{13}$$

in which ‘mod’ denotes the modulo operation. Ideally, a uniform random number in $[-1,1]$ should have been used in Eq. 13 instead of $\sin(t_0^2)$. However, a simple deterministic function is preferred to avoid randomness in the definition of the function.

Figure 5 illustrates the effect of the severity of the irregular change on $\alpha(t)$ and $w_s(t)$ when $n_{tr} = 30$. It can be observed that for a high n_{ti} , the change in $\alpha(t)$ and $w_s(t)$ is gradual and smooth, whereas for a low n_{ti} , this smoothness disappears and the change in their values is more chaotic. Since the change in $\alpha(t)$ and $w_s(t)$ is proportional to the change in the fitness landscape, the value of n_{ti} controls the irregularity in the pattern of the change.

5.5. Pseudocode

Given n_{tr} , n_{ti} , and the change Frequency (τ_t), the pseudocode to evaluate a solution \mathbf{X} when FE function evaluations have been used is as follows:

- Calculate the time step as follows:

$$t = \max \left\{ 0, \left\lceil \frac{FE - FE_0}{\tau_t} \right\rceil \right\},$$

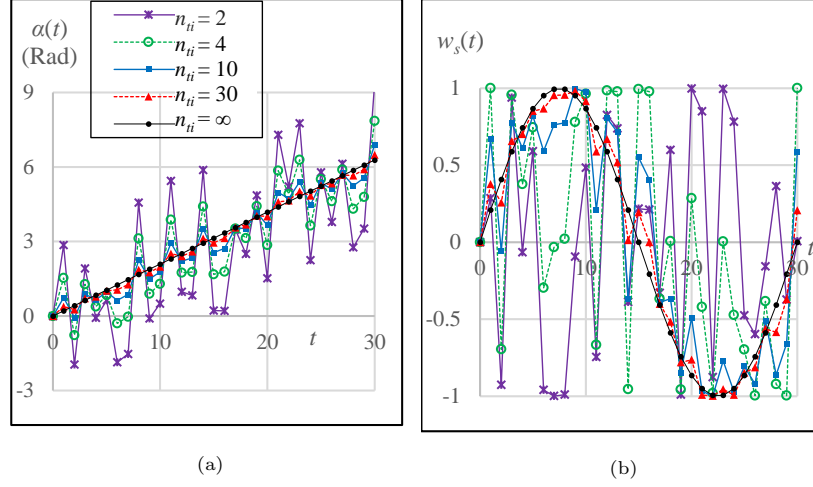


Figure 5: Effect of the severity of irregular change n_{ti} on the a) rotation angle and b) scaling function weight when $n_{tr} = 30$

in which FE_0 is the initial evaluation budget before the first change occurs.

- 395 • Calculate $\alpha(t)$, $w_s(t)$, and $c(t)$ using Eq. 13.
- Calculate $\mathbf{R}(t)$ using Eq. 7 with $\alpha = \alpha(t)$.
- Calculate $s(x, t)$ using Eq. 10.
- Calculate $g_d(X, t)$ using Eq. 8.

Supplementary material S3 illustrates the effect of the proposed method
 400 for simulating dynamic behavior on the landscape of a simple problem defined
 in Eq. 12 with different settings for e_c , n_{ti} , and absence/presence of dynamic
 rotation.

5.6. Comparison with Other Benchmark Generators

This subsection compares the properties of the DDRB with some well-known
 405 or recently proposed benchmark generators. These benchmark generators were
 selected for comparison since they are the most popular ones (see [42] for a com-
 parison of the popularity of dynamic benchmark generators), and/or they have

been recently published in high-quality venues. These benchmark generators are as follows:

410

- M-MPB, a slightly modified variant of the well-known MPB which was employed in [13].
- Real Composite Dynamic Benchmark (RCDB) and Real Rotation Dynamic Benchmark (RRDB) generators developed in [44]
- 415 • The benchmark generator developed in [46] by a combination of RCDB and RRDB. Therefore, it is denoted by Real Rotation Composite Dynamic Benchmark (RRCDB) generator.
- GMPB, the generalized moving peaks benchmark formulated in [42]

Table 3 presents the comparison criteria. Table 4 makes a comparison
 420 among the benchmark generators, which shows that DDRB enjoys several advantages. The generated test problems are deterministic and computationally cheap. DDRB allows the user to control the non-uniformity in the distribution of global minima, a major challenge associated with multimodal optimization. Besides, its dynamic simulation mechanism can be easily combined with an arbitrary static multimodal test problem. This important feature allows for the
 425 generalization of existing widely-accepted static multimodal optimization test problems (e.g., the CEC’2013 test problems for multimodal optimization [18]) to DMMO.

6. Mean Robust Peak Ratio

430

When tracking all global minima is desired, performance indicators based on Peak Ratio (PR), one of the most commonly used performance indicators for static multimodal optimization [18], are a more reasonable choice than the off-line error. For example, it is possible to average the PR values calculated at the end of each time step, as employed in [12] and [16]. However, Calculation

Table 3: Criteria for comparison of the dynamic multimodal benchmark generators

CID	Description of the Criterion
C1	Can the generated test problem have multiple global minima?
C2	Does the distance between basins change over time?
C3	Is the dynamic non-uniformity in the relative distribution of minima controllable? In other words, is it possible that the user makes sure that in some time steps, some minima are closely packed while the rest are far from each other?
C4	Can the sizes of basins of minima change over time?
C5	Can the basins of minima have different shapes?
C6	Can the shapes of basins change over time?
C7	Can the dynamic mechanism simulate complex (but not random or irregular) patterns in the movement of a minimum? For example, moving along a straight line is a simple pattern.
C8	Can the dynamic mechanism simulate a controllable amount of irregularity/randomness in the pattern of the change?
C9	Can the fitness landscape have an adequate global structure around each desired minimum? A landscape that consists of several independent subparts lacks this structure.
C10	Will a solution evaluation be cheap if the problem has several minima/basic functions? Calculation of multiple basic functions or employing transformations for each basic function (e.g., rotation) results in a substantial amount of extra computation.
C11	Is the generated problem deterministic? Randomness in the objective function definition introduces additional statistical uncertainty in the measured performance and makes an exact reproducibility of results hard.
C12	Can the dynamic mechanism in the benchmark generator be easily applied to existing test problems for multimodal optimization? There are already well-developed test problem and benchmark generators for static multimodal optimization.
C13	How complex is the formulation of the benchmark generator? Having many equations, complex equations, and extra user-tuned parameters increases the benchmark generator's complexity.

Table 4: Comparison of the proposed test suite generator (DDRB) with other benchmark generators using the criteria defined in Table 3

CID	M-MPB	RCDB	RRDB	RRCDB	GMPB	DDRB
C1	Yes	No	No	No	No	Yes
C2	Yes	Yes	Yes	Yes	yes	Yes
C3	No	No	No	No	No	Yes
C4	Yes	No	Yes	No	Yes	Yes
C5	No	Yes	No	Yes	Yes	Yes
C6	No	No	No	No	Yes	Yes
C7	No	No	Partially ^a	Yes ^b	No	Yes ^c
C8	yes	Yes	Yes	Yes	yes	Yes
C9	No	Yes	No	Yes	Yes	Yes
C10	No	No	No	No	No	Yes
C11	No	No	No	No	No	Yes
C12	No	No	No	No	No	Yes
C13	Low	High	High	High	High	High

^a Rotation only

^b Combined rotation and translation

^c Combined rotation and nonlinear scaling

435 of PR requires the user to define a tolerance for the objective value (ϵ_f) and a preset niche radius R_{nich} . If these values are not set properly, the detected global minima may be counted incorrectly. For example, if the basins are large, a small R_{nich} may result in a calculated PR greater than the actual one.

Conversely, a large R_{nich} can result in a PR which is lower than the actual
 440 one if the global minima are close to each other. If the global minima are drastically different in size, it might be even impossible to find a proper value for R_{nich} . This becomes even more challenging for DMMO, since the relative distances between global minima change over time.

Setting ϵ_f to a small value can address the issues associated with R_{nich} if
 445 R_{nich} is set to a sufficiently small value. However, this may ignore the global optima that had been *approximately* detected, e.g., the algorithm could get close to the global minimum but could not converge to its exact location. This could be problematic since PR uses a binary measure to check whether the algorithm has found a global minimum or not. A small ϵ_f can thus calculate a lower
 450 PR than the actual one, whereas a large one may perceive two basin-sharing solutions as distinct global minima.

It is also possible to utilize the hill-valley test [55] to check whether two solutions share the same basin without the need to define the niche radius. This test was used in [13] to count the number of detected minima. However, the
 455 hill-valley test is only a heuristic and may result in an incorrect outcome.

This study proposes Robust Peak Ratio (RPR) for measuring the performance of a multimodal optimization method. In contrast to PR, it gives partial credit for *approximately* identified global minima and besides, it automatically sets the niche radius for each global minimum. Algorithm 1 presents the pseudo-
 460 code for the calculation of *RPR*.

According to Algorithm 1, first, the niche radius for each global minimum (R_{nich_k}) is calculated, which corresponds to half of the distance from that global minimum to the closest one (lines 3-5). Then, the subset of candidate solutions related to each global minimum is identified (line 7). If this subset is not empty,
 465 the best solution in the subset is selected as the representative one (line 9).

Algorithm 1: Calculation of RPR

Data: Set of candidate solutions $\mathbb{X} = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N\}$ and the global minima $\mathbb{X}^* = \{\mathbf{X}_1^*, \mathbf{X}_2^*, \dots, \mathbf{X}_{N_{\text{gmin}}}^*\}$, minimum and maximum threshold for global minimum value ($0 < \epsilon_{f_{\text{min}}} \leq \epsilon_{f_{\text{max}}}$)

Result: RPR

```
1 initialization ;
2  $count \leftarrow 0$  ;
3 for  $k \leftarrow 1$  to  $N_{\text{gmin}}$  do
4    $R_{\text{nich}_k} \leftarrow 0.5 \left( \min_{\mathbf{X}_j^* \in \mathbb{X}^*, j \neq k} \|\mathbf{X}_k^* - \mathbf{X}_j^*\| \right)$  ;
5 end
6 for  $k \leftarrow 1$  to  $N_{\text{gmin}}$  do
7   Let  $\mathbb{S} \subseteq \mathbb{X}$  be the set of all solutions in  $\mathbb{X}$  whose distance to  $\mathbf{X}_k^*$  is
   less than  $R_{\text{nich}_k}$  ;
8   if  $\mathbb{S} \neq \emptyset$  then
9     Let  $\tilde{f}^*$  be the minimum value of the solutions in  $\mathbb{S}$  ;
10     $count \leftarrow count + \min \left\{ 1, \max \left\{ 0, \frac{\ln(\epsilon_{f_{\text{max}}}) - \ln(\tilde{f}^* - f^*)}{\ln(\epsilon_{f_{\text{max}}}) - \ln(\epsilon_{f_{\text{min}}})} \right\} \right\}$  ;
11  end
12 end
13  $RPR \leftarrow count / N_{\text{gmin}}$  ;
```

Then, the counted number of detected global minima is increased by a continuous value between 0 and 1 (inclusive) based on the quality of the representative solution, which is its value when compared to $\epsilon_{f_{\min}}$ and $\epsilon_{f_{\max}}$ (line 10). At the end, the counted number of minima is divided by the N_{gmin} to calculate RPR (line 13).

RPR can be easily extended to dynamic problems as follows:

$$MRPR = \frac{1}{1 + t_{\max}} \sum_{t=0}^{t_{\max}} RPR(t), \quad (14)$$

in which $RPR(t)$ is the calculated RPR at the end of time step t and $MRPR$ is the Mean Robust Peak Ratio, which is our proposed performance indicator for DMMO.

7. Numerical Results

This section provides numerical results of a simple DMMO method on the set of dynamic test problems presented in Table 2 combined with the proposed dynamic simulation procedure proposed in Section 5. The numerical results can serve as a reference for future studies that use DDRB.

7.1. Dynamic Scenarios

Different scenarios are considered in this study. Starting from the base scenario, each subsequent scenario emphasizes one specific type of challenge associated with DMMO. Table 5 presents the scenarios considered in this study and the challenge emphasized in each scenario. In all scenarios, the first change occurs after $4000DN_{\text{gmin}}$ evaluations.

7.2. Dynamic Multimodal Optimization Method

This section benchmarks a dynamic variant of Covariance Matrix Self-Adaptation Evolution Strategy with Repelling Subpopulations (DRS-CMSA-ES). It combines RS-CMSA-ES [51] with a simple strategy to utilize previous information. When a change occurs, a new restart of RS-CMSA-ES is run in which the

Table 5: Scenarios considered in this study and their characteristic features.

Scenario	h_{GO}	e_c	n_{tr}	n_{ti}	$\frac{\tau_t}{DN_{gmin}}$	t_{max}	Feature
I	0.3	0.5	30	∞	2000	40	Base Scenario
II	0.6	0.5	30	∞	2000	40	Hard global optimization
III	0.3	0.1	30	∞	2000	40	Hard niching problem
IV	0.3	0.5	10	∞	2000	40	Severe changes
V	0.3	0.5	30	5	2000	40	Irregular changes
VI	0.3	0.5	30	∞	500	40	Fast-changing problem

center of the subpopulations is one of the archived solutions in the previous time step, and the initial global step size is set to half of the default value. If all the previously archived solutions have been tried, the center of a new population is determined according to the default setting in RS-CMSA-ES. The number of subpopulations and the subpopulation size increase factor are set to 1 and 1.05, respectively.

7.3. Results and discussion

Table 6 presents the MRPR of DRS-CMSA-ES for each problem in each scenario, calculated over 50 independent runs when $\epsilon_{f_{max}} = 0.1$ and $\epsilon_{f_{min}} = 10^{-5}$. The average RMPR in each scenario, calculated over 10 problems, is also provided. It can be observed that:

- A considerable reduction in MRPR is observed in Scenario II since a higher h_{GO} has increased the difficulty of the global optimization. Similarly, a lower e_c (Scenario III) has increased the difficulty of the DMMO problem by intensifying the hardness of its niching aspects (e.g., severe changes in shapes, sizes and relative distances of global minima). This shows that these two parameters can reliably control the challenges associated with both global optimization and multimodal optimization, allowing the researcher to scrutinize pros and cons of a DMMO method in coping with these two challenges in isolation.

- 510 • A significant performance drop can be observed for the fast-changing problems (Scenario VI) as well, demonstrating an increase in the intensity of challenges associated with dynamic aspects of the problems.
- The change severity (Scenario IV) and irregularity of changes (Scenario V) have little impact on the performance of DRS-CMSA-ES, which practically does not employ any prediction method. This may account for the insensitivity of its performance to these two factors. Utilization of a prediction method can improve the performance when the changes are more regular and less severe, but it may deteriorate it otherwise.
- 515 • For most problems, DRS-CMSA-ES can reach a high RPR in the base scenario. It can be deduced that by increasing the change frequency, RS-CMSA-ES can reach $MRPR \cong 1$ at least in the base scenario. This means that there is at least one method that can detect and track all global minima if the change frequency is reasonably high.
- 520 • Since a robust DMMO method should be reliable in diverse scenarios, researchers are encouraged to adopt the average MRPR when comparing different methods. The average MRPR of DRS-CMSA-ES is 0.764 according to Table 6.
- 525

8. Summary and Conclusions

This study has proposed a method for generating tunable test problems for evaluating dynamic multimodal optimization (DMMO) methods in the continuous domain. The method, called Deterministic Distortion and Rotation Benchmark (DDRB), was employed to generate ten DMMO test problems in which the user can control the intensity of each type of challenge associated with DMMO problems. The proposed scaling function simulates dynamic behavior by distorting the fitness landscape, which changes the shapes, sizes, locations, and relative distribution of the global minima over time. Besides, this study

Table 6: Mean Robust Peak Ratio (MRPR) of DRS-CMSA-ES on 10 DMMO test problems in different scenarios

PID	Scenario					
	I	II	III	IV	V	VI
1	0.953 ± 0.0024	0.444 ± 0.0067	0.813 ± 0.0046	0.949 ± 0.0027	0.947 ± 0.0026	0.500 ± 0.0057
2	0.980 ± 0.0025	0.816 ± 0.0059	0.868 ± 0.0039	0.987 ± 0.0022	0.980 ± 0.0024	0.739 ± 0.0052
3	0.973 ± 0.0020	0.774 ± 0.0050	0.795 ± 0.0047	0.973 ± 0.0025	0.965 ± 0.0031	0.823 ± 0.0058
4	0.822 ± 0.0053	0.571 ± 0.0068	0.525 ± 0.0036	0.844 ± 0.0045	0.845 ± 0.0047	0.482 ± 0.0058
5	0.726 ± 0.0045	0.332 ± 0.0039	0.489 ± 0.0032	0.803 ± 0.0037	0.763 ± 0.0034	0.381 ± 0.0049
6	0.936 ± 0.0024	0.821 ± 0.0021	0.874 ± 0.0024	0.930 ± 0.0026	0.927 ± 0.0023	0.595 ± 0.0025
7	0.919 ± 0.0027	0.882 ± 0.0030	0.713 ± 0.0037	0.935 ± 0.0023	0.927 ± 0.0021	0.726 ± 0.0042
8	0.932 ± 0.0028	0.835 ± 0.0027	0.717 ± 0.0031	0.942 ± 0.0029	0.928 ± 0.0024	0.712 ± 0.0033
9	0.657 ± 0.0026	0.404 ± 0.0023	0.376 ± 0.0021	0.656 ± 0.0019	0.659 ± 0.0022	0.290 ± 0.0022
10	0.712 ± 0.0036	0.415 ± 0.0028	0.468 ± 0.0023	0.738 ± 0.0022	0.718 ± 0.0032	0.241 ± 0.0024
Avg.	0.861 ± 0.0010	0.629 ± 0.0014	0.664 ± 0.0011	0.876 ± 0.0009	0.866 ± 0.0009	0.549 ± 0.0014

formulated a simple deterministic way to control the regularity of the change pattern and proposed the Robust Peak Ratio to overcome the sensitivity of the well-known Peak Ratio indicator to the user-defined tolerance for the global minimum value and niche radius by giving partial credit for the *approximately* identified global minima and determining the corresponding solutions for each global minimum. Finally, it has provided the results of a simple DMMO method for these ten test problems in six dynamic scenarios as a reference for future studies.

When compared with existing benchmark generators, DDRB is deterministic, which allows for reproducible and reliable comparison of results across different platforms and software. The user has control over dynamic variation in the non-uniformity of the distribution of global minima, a well-known challenge associated with multimodal optimization. The resultant test problems are computationally cheap. Finally, the dynamic mechanism of DDRB allows for easy generalization of existing and well-developed static multimodal optimization test problems to DMMO.

Tunability of the proposed test problems allows the user to have complete control over the significance of each type of challenge associated with DMMO problems (i.e., global optimization, niching, and dynamic optimization) with only a few control parameters. Such a control allows for studying the pros and cons of a DMMO method in coping with each type of challenge. For benchmarking and comparing DMMO methods, it allows setting a reasonable balance between different aspects of the DMMO problem. For example, in our case, more challenging niching features have resulted in a significant drop in performance.

The employed methodology in this study can be generalized to other types of problems involving distinct challenges. For example, a tunable benchmark generator for dynamic multiobjective optimization should be able to easily control the difficulty of i) global optimization (finding a solution in the global efficient set), ii) diversity preservation (ideal spread of population over the global efficient set), and iii) dynamic optimization (tracking the global efficient set). Besides, the change in the objective values may necessitate multimodal optimization as

well because a local efficient set may become the global one after some time. This challenge, which has not been adequately explored in the literature, highlights the importance of multimodal optimization for dynamic multiobjective optimization.

Acknowledgment

This work was supported by Australian Research Council (Discovery Project DP190102637). The last author acknowledges support from CONACyT project no. 2016-01-1920 (*Investigación en Fronteras de la Ciencia 2016*) and from a 2018 SEP-Cinvestav grant (application no. 4). He was also partially supported by the Basque Government through the BERC 2018-2021 program by the Spanish Ministry of Science. Computational work in this study was performed at National Computational Infrastructure of Australia. The interested readers are encouraged to email the corresponding author to receive the code of the proposed benchmark generator in MATLAB and/or Python.

References

- [1] T. T. Nguyen, S. Yang, J. Branke, Evolutionary dynamic optimization: A survey of the state of the art, *Swarm and Evolutionary Computation* 6 (2012) 1–24.
- [2] M. Mavrovouniotis, C. Li, S. Yang, A survey of swarm intelligence for dynamic optimization: Algorithms and applications, *Swarm and Evolutionary Computation* 33 (2017) 1–17.
- [3] K. Deb, U. B. Rao, S. Karthik, Dynamic multi-objective optimization and decision-making using modified nsga-ii: a case study on hydro-thermal power scheduling, in: *International conference on evolutionary multi-criterion optimization*, Springer, 2007, pp. 803–817.

- 595 [4] D. Rohde, B. R. Knudsen, T. Andresen, N. Nord, Dynamic optimization of control setpoints for an integrated heating and cooling system with thermal energy storages, *Energy* 193 (2020) 116771.
- [5] L. T. Bui, Z. Michalewicz, E. Parkinson, M. B. Abello, Adaptation in dynamic environments: A case study in mission planning, *IEEE Transactions on Evolutionary Computation* 16 (2) (2012) 190–209.
- 600 [6] M. Okulewicz, J. Mańdziuk, The impact of particular components of the pso-based algorithm solving the dynamic vehicle routing problem, *Applied Soft Computing* 58 (2017) 586–604.
- [7] B. Sun, Y. Yang, J. Shi, L. Zheng, Dynamic pick-up and delivery optimization with multiple dynamic events in real-world environment, *IEEE Access* 7 (2019) 146209–146220.
- 605 [8] Z. Deng, M. Liu, H. Chen, W. Lu, P. Dong, Optimal scheduling of active distribution networks with limited switching operations using mixed-integer dynamic optimization, *IEEE Transactions on Smart Grid* 10 (4) (2018) 4221–4234.
- 610 [9] T. Blackwell, J. Branke, Multi-swarm optimization in dynamic environments, in: *Workshops on Applications of Evolutionary Computation*, Springer, 2004, pp. 489–500.
- [10] D. Yazdani, M. N. Omidvar, J. Branke, T. T. Nguyen, X. Yao, Scaling up dynamic optimization problems: A divide-and-conquer approach, *IEEE Transactions on Evolutionary Computation* 24 (1) (2019) 1–15.
- 615 [11] J. Branke, T. Kaußler, C. Smidt, H. Schmeck, A multi-population approach to dynamic optimization problems, in: *Evolutionary design and manufacture*, Springer, 2000, pp. 299–307.
- [12] S. Kundu, S. Biswas, S. Das, P. N. Suganthan, Crowding-based local differential evolution with speciation-based memory archive for dynamic mul-

- 620 multimodal optimization, in: Proceedings of the 15th annual conference on Genetic and evolutionary computation, ACM, 2013, pp. 33–40.
- [13] W. Luo, X. Lin, T. Zhu, P. Xu, A clonal selection algorithm for dynamic multimodal function optimization, *Swarm and Evolutionary Computation* 50 (2019) 100459.
- 625 [14] S. Cheng, H. Lu, Y.-n. Guo, X. Lei, J. Liang, J. Chen, Y. Shi, Dynamic multimodal optimization: A preliminary study, in: 2019 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2019, pp. 279–285.
- [15] R. Azzouz, S. Bechikh, L. B. Said, Dynamic multi-objective optimization using evolutionary algorithms: A survey, in: Recent advances in evolutionary multi-objective optimization, Springer, 2017, pp. 31–70.
- 630 [16] S. Cheng, H. Lu, W. Song, J. Chen, Y. Shi, Dynamic multimodal optimization using brain storm optimization algorithms, in: International Conference on Bio-Inspired Computing: Theories and Applications, Springer, 2018, pp. 236–245.
- 635 [17] N. Hansen, S. Finck, R. Ros, A. Auger, Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions, Tech. rep. (2009).
- [18] X. Li, A. Engelbrecht, M. G. Epitropakis, Benchmark functions for cec’2013 special session and competition on niching methods for multimodal function optimization, Tech. rep., RMIT University (2013).
- 640 [19] A. Ahrari, K. Deb, A novel class of test problems for performance evaluation of niching methods, *IEEE Transactions on Evolutionary Computation* 22 (6) (2017) 909–919.
- [20] S. Jiang, M. Kaiser, S. Yang, S. Kollias, N. Krasnogor, A scalable test suite for continuous dynamic multiobjective optimization, *IEEE transactions on cybernetics* 50 (6) (2019) 2814–2826.
- 645

- [21] S. Jiang, S. Yang, Evolutionary dynamic multiobjective optimization: Benchmarks and algorithm comparisons, *IEEE transactions on cybernetics* 47 (1) (2017) 198–211.
- 650 [22] Y. Wang, J. Yu, S. Yang, S. Jiang, S. Zhao, Evolutionary dynamic constrained optimization: Test suite construction and algorithm comparisons, *Swarm and Evolutionary Computation* 50 (2019) 100559.
- [23] R. Azzouz, S. Bechikh, L. B. Said, W. Trabelsi, Handling time-varying constraints and objectives in dynamic evolutionary multi-objective optimization, *Swarm and evolutionary computation* 39 (2018) 222–248.
- 655 [24] J. Branke, Memory enhanced evolutionary algorithms for changing optimization problems, in: *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99* (Cat. No. 99TH8406), Vol. 3, IEEE, 1999, pp. 1875–1882.
- 660 [25] F. Oppacher, M. Wineberg, The shifting balance genetic algorithm: Improving the ga in a dynamic environment, in: *Proceedings of the genetic and evolutionary computation conference*, Vol. 1, 1999, pp. 504–510.
- [26] D. Parrott, X. Li, Locating and tracking multiple dynamic optima by a particle swarm model using speciation, *IEEE Transactions on Evolutionary Computation* 10 (4) (2006) 440–458.
- 665 [27] C. Li, S. Yang, A general framework of multipopulation methods with clustering in undetectable dynamic environments, *IEEE Transactions on Evolutionary Computation* 16 (4) (2012) 556–577.
- [28] F. B. Ozsoydan, A. Baykasoğlu, Quantum firefly swarms for multimodal dynamic optimization problems, *Expert Systems with Applications* 115 (2019) 189–199.
- 670 [29] X.-W. Luo, Z.-J. Wang, R.-C. Guan, Z.-H. Zhan, Y. Gao, A distributed multiple populations framework for evolutionary algorithm in solving dynamic optimization problems, *IEEE Access* 7 (2019) 44372–44390.

- 675 [30] J. K. Kordestani, A. E. Ranginkaman, M. R. Meybodi, P. Novoa-Hernández, A novel framework for improving multi-population algorithms for dynamic optimization problems: A scheduling approach, *Swarm and evolutionary computation* 44 (2019) 788–805.
- [31] C. Li, T. T. Nguyen, M. Yang, M. Mavrovouniotis, S. Yang, An adaptive multipopulation framework for locating and tracking multiple optima, 680 *IEEE transactions on evolutionary computation* 20 (4) (2015) 590–605.
- [32] J. Branke, H. Schmeck, Designing evolutionary algorithms for dynamic optimization problems, in: *Advances in evolutionary computing*, Springer, 2003, pp. 239–262.
- 685 [33] K. Trojanowski, Z. Michalewicz, Searching for optima in non-stationary environments, in: *Proceedings of the 1999 congress on evolutionary computation-CEC99* (Cat. No. 99TH8406), Vol. 3, IEEE, 1999, pp. 1843–1850.
- [34] A. Ahrari, S. Elsayed, R. Sarker, D. Essam, C. A. C. Coello, Towards a 690 more practically sound formulation of dynamic problems and performance evaluation of dynamic search methods, in: *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, IEEE, 2020, pp. 1387–1394.
- [35] I. Moser, R. Chiong, Dynamic function optimization: the moving peaks benchmark, in: *Metaheuristics for Dynamic Optimization*, Springer, 2013, 695 pp. 35–59.
- [36] A. Sharifi, J. K. Kordestani, M. Mahdaviani, M. R. Meybodi, A novel hybrid adaptive collaborative approach based on particle swarm optimization and local search for dynamic optimization problems, *Applied Soft Computing* 32 (2015) 432–448.
- 700 [37] S. Yang, C. Li, A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments, *IEEE Transactions on Evolutionary Computation* 14 (6) (2010) 959–974.

- [38] S. Dennis, A. Engelbrecht, A review and empirical analysis of particle swarm optimization algorithms for dynamic multi-modal optimization, in: 2020 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2020, pp. 1–8.
- [39] U. Halder, S. Das, D. Maity, A cluster-based differential evolution algorithm with external archive for optimization in dynamic environments, IEEE transactions on cybernetics 43 (3) (2013) 881–897.
- [40] R. Mukherjee, G. R. Patra, R. Kundu, S. Das, Cluster-based differential evolution with crowding archive for niching in dynamic environments, Information Sciences 267 (2014) 58–82.
- [41] R. W. Morrison, K. A. De Jong, A test problem generator for non-stationary environments, in: Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Vol. 3, IEEE, 1999, pp. 2047–2053.
- [42] D. Yazdani, M. N. Omidvar, R. Cheng, J. Branke, T. T. Nguyen, X. Yao, Benchmarking continuous dynamic optimization: Survey and generalized test suite, IEEE Transactions on Cybernetics (2020). doi:10.1109/TCYB.2020.3011828.
- [43] W. Luo, B. Yang, C. Bu, X. Lin, A hybrid particle swarm optimization for high-dimensional dynamic optimization, in: Asia-Pacific Conference on Simulated Evolution and Learning, Springer, 2017, pp. 981–993.
- [44] C. Li, S. Yang, A generalized approach to construct benchmark problems for dynamic optimization, in: Asia-Pacific Conference on Simulated Evolution and Learning, Springer, 2008, pp. 391–400.
- [45] C. Li, S. Yang, T. Nguyen, E. L. Yu, X. Yao, Y. Jin, H. Beyer, P. Suganthan, Benchmark generator for cec 2009 competition on dynamic optimization, Tech. rep. (2008).

- 730 [46] A. Ahrari, S. Elsayed, R. Sarker, D. Essam, C. A. C. Coello, Adaptive
multilevel prediction method for dynamic multimodal optimization, *IEEE
Transactions on Evolutionary Computation* (2021). doi:10.1109/TEVC.
2021.3051172.
- [47] M. Jamil, X.-S. Yang, H.-J. Zepernick, Test functions for global optimiza-
735 tion: a comprehensive survey, in: *Swarm intelligence and Bio-inspired Com-
putation*, Elsevier, 2013, pp. 193–222.
- [48] M. M. Ali, C. Khompatraporn, Z. B. Zabinsky, A numerical evaluation
of several stochastic algorithms on selected continuous global optimization
test problems, *Journal of global optimization* 31 (4) (2005) 635–672.
- 740 [49] K. Deb, A. Saha, Multimodal optimization using a bi-objective evolution-
ary algorithm, *Evolutionary computation* 20 (1) (2012) 27–62.
- [50] D. Mortari, On the rigid rotation concept in n-dimensional spaces, *Journal
of the Astronautical Sciences* 49 (3) (2001) 401–420.
- [51] A. Ahrari, K. Deb, M. Preuss, Multimodal optimization by covariance ma-
745 trix self-adaptation evolution strategy with repelling subpopulations, *Evo-
lutionary computation* 25 (3) (2017) 439–471.
- [52] I. Hatzakis, D. Wallace, Dynamic multi-objective optimization with evolu-
tionary algorithms: a forward-looking approach, in: *Proceedings of the 8th
annual conference on Genetic and evolutionary computation*, ACM, 2006,
750 pp. 1201–1208.
- [53] S. Jiang, S. Yang, A steady-state and generational evolutionary algorithm
for dynamic multiobjective optimization, *IEEE Transactions on Evolution-
ary Computation* 21 (1) (2017) 65–82.
- [54] M. Rong, D. Gong, Y. Zhang, Y. Jin, W. Pedrycz, Multidirectional pre-
755 diction approach for dynamic multiobjective optimization problems, *IEEE
transactions on cybernetics* (99) (2018) 1–13.

- [55] 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.