Measuring Optimiser Performance on a Conical Barrier Tree Benchmark

Itshak Tkach
its.tkach@gmail.com
Dept. of Computing, Goldsmiths, University of London
London, UK

Tim Blackwell
t.blackwell@gold.ac.uk
Dept. of Computing, Goldsmiths, University of London
London, UK

ABSTRACT

The common method for testing metaheuristic optimisation algorithms is to benchmark against problem test suites. However, existing benchmark problems limit the ability to analyse algorithm performance due to their inherent complexity. This paper proposes a novel benchmark, BTB, whose member functions have known geometric properties and critical point topologies. A given function in the benchmark is a realisation of a specified barrier tree in which funnel and basin geometries, and values and locations of all critical points are predetermined. We investigate the behaviour of two metaheuristics, PSO and DE, on the simplest manifestations of the framework, ONECONE and TWOCONES, and relate algorithm performance to a downhill walker reference algorithm. We study success rate, defined as the probability of optimal basin attainment, and inter-basin mobility. We find that local PSO is the slowest optimiser on the unimodal ONECONE but surpasses global PSO in all TWOCONES problems instances below 70 dimensions. DE is the best optimiser when basin difference depths are large but performance degrades as the differences become smaller. LPSO is the superior algorithm in the more difficult case where basins have similar depth. DE consistently finds the optimum basin when the basins have equal size and a large depth difference in all dimensions below 100D; the performance of LPSO falls away abruptly beyond 70D.

CCS CONCEPTS

• **Theory of computation → Theory of randomized search heuristics**.

KEYWORDS

Optimization, Problem benchmarks, Particle swarm optimization, PSO, Differential evolution, DE

ACM Reference Format:
Itshak Tkach and Tim Blackwell. 2022. Measuring Optimiser Performance on a Conical Barrier Tree Benchmark . In *Genetic and Evolutionary Computation Conference (GECCO '22), July 9–13, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3512290.3528842

1 INTRODUCTION

Evolutionary algorithms (EAs) are being developed and exploited to solve real-world problems [28],[7] [26] where an optimal solution is required. Evaluation tools and parameters are required to measure the effectiveness of these algorithms [6]. However, current approaches for the analysis of the performances of EAs are very complex [10], [31]. The performance of an EA is expected to strongly depends on the nature of the problem. The performance evaluation of an EA on all possible even within a restricted class is too expensive to be considered practical. Numerous artificial benchmark functions such as the IEEE Congress on Evolutionary Computation (CEC) problem suites have been designed and widely applied [20], [12] for EA evaluation.

The Particle Swarm Optimisation (PSO) is one of the most known population optimsers. PSO has been widely applied and evaluated on many CEC problem functions, but theoretical results are sparse without structured and principled matching of the problem to algorithm apart from general advice (e.g. apply PSO with local communication to a multi-modal problem).

This paper aims to understand the workings of an algorithm by employing a benchmark generator (BG) whose test function instances have known geometry. The BG can implement any barrier tree (e.g. any combination of funnels and basins). The behaviour of three algorithms on the simplest BG functions - unimodal and bimodal problems - are reported in this work.

The proposed BG possesses several advantages. The problem can be controlled for systematic and principled problem-algorithm investigation and algorithm behaviour can be related to geometric properties and to barrier tree complexity. The values and locations of all critical points are known and the BG can generate a huge variety of problems.

The remainder of this paper is organised as follows. Background work is described in Section 2. Next, in Section 3, the barrier tree benchmark (BTB) is defined. Section 4 details experiment setup and section 5 presents results for particle swarm optimisation algorithms with local and global communication networks (L/GPSO) and DE on ONECONE and TWOCONES BTB instances. The paper concludes with an overview of the main empirical findings.

2 BACKGROUND

Benchmark problems for real-world applications such as the travelling salesman problems, and the knapsack problems among others [4], have been used to evaluate EAs. However, the high costs associated with these benchmarks limit their use for a more comprehensive analysis [21]. Consequently, benchmarks with artificial problem suites have been proposed and widely adopted.

The CEC problem suites have provided a series of standard benchmark functions and their variants are convenient in terms of comparison and implementation [12]. A set of twenty-five single-objective optimisation test functions was assembled for a competition at the 2005 Congress on Computational Intelligence. This benchmark consists of five unimodal functions, seven basic multimodal functions, two multimodal complex functions, and eleven hybrid functions, which were weighted sums of ten basic test problems [25].

This test suite became the standard for comparing algorithms, and in subsequent years possible improvements were identified, resulting in a new benchmark suite being developed for CEC 2013 [15]. This collection of twenty-eight functions expanded on the 2005 composite functions and added new test problems. It includes five unimodal functions, fifteen basic multimodal functions, and eight composition functions, and has become the more recent default standard for testing and comparing optimization algorithms.

Another set of benchmark problems was designed at the BBOB competition [11]. Generators of random problems were introduced for improving existing benchmarks. A polynomial test problem generator (NGLI) with known basins and saddle points has been proposed [17]. It allowed the control of the difficulty levels and the modality, but there was a strong relation between the dimensionality and modality of the NGLI. A max set of Gaussians (MSG) controlling the number of basins has also been put forward [10]. A general framework for generating test functions with controllable properties, in which the structures of the Gaussian functions were modified also been proposed [16].

A modified Gaussian fitness landscape generator to compare the performances of optimisation algorithms [14] and a nonseparable test problem generator (N-Peaks) of different geometries with randomly distributed basins [29] have also been considered.

Benchmarking on these problem suites may result in overfitting and customisation of the algorithms, limiting their performance generalisation to other problems [18]. Also, the complexity of these benchmarks may limit the understanding of the strengths and weaknesses of these algorithms [21]. Thus, a simple and fully known benchmark generator is needed to understand the behaviour of the optimisation algorithms.

There have been a handfull of studies of PSO performance in multi-funnel environments [27] [9], [22]. The general view is that PSO struggles in multi-funnel scenarios. Relative attractor distance and heights has been found to be more pertinent to PSO performance [30] than modality. These studies are sporadic but suggestive: we propose that a systematic study of barrier tree complexity and function geometry and topography is required. This paper seeks to outline how such a process might proceed.

Despite numerous studies trying to optimise the benchmark functions, to the best of our knowledge, no research has been conducted to achieve a comprehensive understanding of the relationship between optimiser performance and geometric quantities such as basin size, landscape topography and barrier tree complexity.

3 A BARRIER TREE BENCHMARK

A barrier tree is a hierarchical representation of a fitness landscape: leaves correspond to local minimum values and internal nodes represent saddles [24]. The representation obscures geometric features (positions of critical points and shape and extent of basins and funnels) in favour of a topological description. Figure 1 depicts a barrier tree with two minima of function values $d_{1,2}$ and a saddle of value d_{12}.

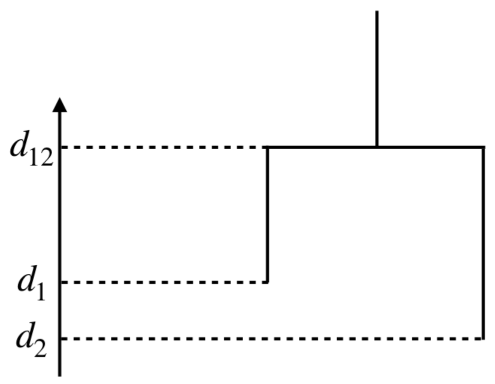


Figure 1: A bimodal barrier tree depicting a function with two minima of values d_1, d_2 and a saddle of value d_{12}. The horizontal axis represents the entire search space.

A barrier tree can be fleshed out by providing optimum and saddle locations, and by specifying basin and funnel extent and shape i.e. by providing an aligning spatial description. Extent can be specified by an arbitrary subdivision of the search space into disjoint regions and shape by assigning unimodal functions to regions such that function minimum and saddle values correspond to the desired barrier tree. A point x would then be evaluated by first identifying the containing region and then applying the associated region function. A benchmark (BTB) would consist of realisations of a collection of barrier trees.

The intuitive notion of a basin, as a region surrounding an attracting point, and a funnel, as a region surrounding two or more basins, can be tightened by considering *downhill walks*. A downhill walk, also know as an adaptive walk, is a terminating succession of points of finite separation and decreasing fitness. The basin of an attractor or optimum can then be defined as a region $\mathcal{B}$ such that all downhill walks starting in $\mathcal{B}$ terminate at x. A funnel $\mathcal{F}$ is a region in which all downhill walks starting in $\mathcal{F}$ enter two or more basins. (We avoid an ambiguity in funnel/basin definition for landscapes with neutral subregions such as plateau and non-isolated attractors by considering landscapes without neutral regions.)

The construction of a barrier tree test function (BTF) from a barrier tree is arbitrary but subject to the requirement that the ordering of saddle values and basin depths is preserved. Consider a barrier tree representing two basins $\mathcal{B}_{1,2}$ of depths $d_{1,2}$ touching at a saddle with value d_{12}. The basins are surrounded by a funnel $\mathcal{F}_{12}$; the entire space (the search space) is $X = \mathcal{B}_1 \cup \mathcal{B}_2 \cup \mathcal{F}_{12}$.

Unimodal functions $f_{1,2}$ with optima $x_{1,2}$ and optimal values $d_{1,2}$ are associated with basins $\mathcal{B}_{1,2}$. Unimodal f_{12} with optimum x_{12} and optimal value d_{12} provides funnel topography. The requirement $x_1 \neq x_2 \neq x_{12}$ and, from the barrier tree, the condition $d_{1,2} < d_{12}$, ensure the correct basin-funnel relationships $f_1(\mathcal{B}_1), f_2(\mathcal{B}_2) \leq d_{12}$ and $f(\mathcal{F}_{12}) \geq d_{12}$.

A BTF is instantiated by defining regions, choosing critical points and picking unimodal functions. For example, 'TWOCONES', a bimodal BTF corresponding to Fig. 1 can be constructed as follows. Choose D-dimensional balls $B_{1,2}(x_{1,2}, r_{1,2})$ of radii $r_{1,2}$ and centres $x_{1,2}$ and place the D-balls, touching at x_{12}, in a search space X. The basins are the regions $\mathcal{B}_{1,2} = B_{1,2} \setminus x_{12}$ and the funnel $\mathcal{F}$ is the region $\mathcal{F} = X \setminus \mathcal{B}_1 \cup \mathcal{B}_2$. (The saddle x_{12} is not strictly in either basin because a downhill walk starting at x_{12} can proceed into either basin i.e. $x_{12} \in \mathcal{F}$.) Cone functions $f_i = m_i|x - x_i| + d_i$, $i = 1, 2, 12$, of gradients $m_i > 0$, depths d_i and centres x_i are assigned to basins $\mathcal{B}_{1,2}$ and $\mathcal{F}$ such that $x_{1,2} \in \mathring{B}_{1,2}$ (centres in ball interiors), $x_3 = x_{12}$ (funnel centred on the saddle). $m_{1,2}$ are chosen so that $f(\mathring{B}_{1,2} < f_3(x_s))$; m_3 is arbitrary (but positive). Fig. 2 depicts a 2D TWOCONES BTF.

A saddle surface $S = \partial B_1 \cup \partial B_2$ can be defined as the level set $f(S) = f(x_{12})$. A point x is evaluated:

$$f(x) = \begin{cases} f_{1,2}(x) & \text{if } x \in \mathring{B}_{1,2} \text{ (basin interior)}, \\ d_{12} & \text{if } x \in S \text{ (basin boundary or saddle)}, \\ f_{12}(x) & \text{if } x \in \mathcal{F} \text{ (funnel)}. \end{cases}$$

The simplest barrier tree function is any unimodal function placed in a search space i.e. a single basin and no funnels. For example, choosing cones again, provides a 'ONECONE' function and an entire barrier tree benchmark 'MANYCONES' can be constructed by furnishing a barrier tree collection with cones. The cone construction is not, however necessary, and region functions can be unimodal functions of arbitrary difficulty, for example high-conditioned ellipsoids.

More complex BTFs can be built in a similar way by a hierarchical structuring of basins and funnels. These constructions would not yield smooth functions (in the above example there is a cliff edge at $\partial\mathcal{B}_1 \cup \partial\mathcal{B}_2$): the benchmark is intended for non gradient optimisers, that is, algorithms that rely solely on function value and do not require smoothness.

The advantage of the barrier tree banchmark is that all geometric (basin and saddle optimum locations and funnel and basin shapes, areas and volumes) and topological properties (numbers and values of optimums and saddles) are known. A principled investigation of the relationship between algorithm performance and function property (geometric and topological) can then be pursued.

The BTB suggests several metrics of function difficulty e.g. relative volume and surface of the optimal basin, number of funnels and barrier height. A metric that differentiates functions of differing topographies but with identical barrier tree configurations can be motivated with the notion of a downhill 'slider' (DHS). The downhill walker, as previously defined, proceeds by a succession of N steps x_t, $t = \{1, 2...N\}$ such that $f(x(t+1) < f(x_t)$. In the limit $|x_{t+1} - x_t| \to 0, N \to \infty$ such that walk length $L = \Sigma_{t=1}^{N-1}|x_{t+1} - x_t|$ remains finite and non-zero, the walk outlines a continuous path in X. Suppose that an idealised continuous path following downhill slider is launched from any point in X. The probability that the DHS will arrive at the global optimum will be related to the surface area of the optimal basin.

In the TWOCONES example, suppose the DHS starts at an arbitrary point in $\mathcal{F}$ and that the saddle is at the centre of X. The DHS will move down the funnel, eventually meeting the level set S. The slider will now, depending on its location on S, proceed into a basin and will find one of two optima $x_{1,2}$. If $r_1 = r_2$, the probability of the slider reaching the boundary of basin 1 or basin 2, when averaged over all slider initialisations, will be $\frac{1}{2}$.

The probability that the DHS will optimise a general BTF can be calculated by multiplying probabilities e.g. in a two funnel scenario, by multiplying the probability of finding the optimal funnel by the probability of finding the optimal basin within that funnel. This probability is calculable for BTFs built from dual cones and funnels as described above; otherwise a downhill walker DHS implementation can serve as a reference algorithm.

The BTB as described is not unambiguous because a given barrier tree within a search space of given dimension can be realised in many ways: the relative positions of the basin optima, the basin sizes, the shapes of the unimodal region functions and the form of the saddle surfaces are arbitrary. Algorithm performance, however, is not expected to depend only on the barrier tree (in which only the depths and topological relationships of the the basins are specified). For example, BTB instances with high conditioned basin functions and saddle surfaces might prove more challenging than more symmetric instances derived from the same barrier tree. Specific realisations of the BTB must be defined in any exhaustive study of algorithm performance. Here, we propose, for ease of implementation, a conical BTB with conical basin shapes and spherical saddle surfaces and we demonstrate comparative algorithm behaviour for BTB instances with varying basin size.

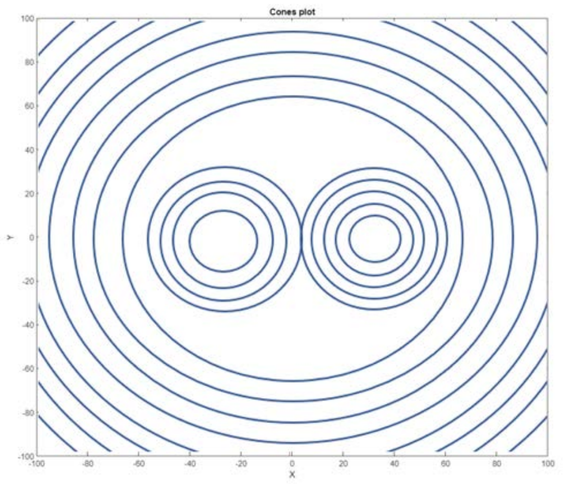


Figure 2: A contour map of the 2D TWOCONES function. A figure of eight saddle contour separates the two basins. The rightmost minimum is deeper; funnel contours are spherical surfaces of increasing value away from the basins.

4 METHODS

A series of ONECONE and TWOCONES experiments was performed with three population algorithms in order to test the feasibility of the BTB approach. The optimisers, local and global particle swarm optimisation (L/GPSO) and differential evolution were chosen for their popularity and similarity.

A standard PSO [13, 19, 23] with update rule

$$\begin{aligned} v_i(t+1) &= wv_i(t) + cu_1 \circ (n_i(t+1) - x_i(t)) \\ &\quad + cu_2 \circ (p_i(t+1) - x_i(t)) \\ x_i(t+1) &= x_i(t) + v_i(t+1) \end{aligned} \tag{1}$$

was chosen. Here, x_i, v_i, p_i are particle position, velocity and historical best position of particle i, $u_{1,2} \sim U(0,1)$ are uniform random variables in $[0,1]^D$ and $\circ$ is the Hadamard (entry-wise) product, n_i is the historical best position of the best neighbour in i's social network (an arbitrary choice is made in the case of a tie). The neighbourhood is fully connected in GPSO i.e. $n_i = g$, the swarm's best ever position. LPSO neighbourhoods are localised: we ran experiments with the ring neighbourhood in which each particle has access to two other particles.

The DE/best/1 version DE variant was chosen because of its competitive and robust performance [5]. A trial update for particle i at x_i is found according to

$$\begin{aligned} &\text{if } u \sim U(0,1) < CR \text{ or } d == r \\ &\quad y_d = g_d + F(x_{jd} - x_{kd}) \\ &\text{else} \\ &\quad y_d = x_{id} \end{aligned} \tag{2}$$

where g is the best particle, j, k are random particle indices such that $i \neq j \neq k$ and $r = U(\{1, 2, \ldots D\})$ is a random component. y replaces x if $f(y) \leq f(x_i)$.

The search space for all experiments was $X = [-100, 100]^D$. Each function instance of ONECONE has a single cone of unit gradient and depth -10. The cone centre, different for each ONECONE instance, is chosen from the uniform distribution on a hypersphere centred on O^D and of radius 10.

TWOCONES is as specified in Sec. 3. x_1 was a uniform random point on the hypersphere of radius r_1 centred on x_{12}; x_2, the global optimum, was positioned diametrically opposite to x_1 such that the basins touched at a random point (x_{12}) at a distance $2(r_1 + r_2)$ from O^D. Each $x_1, x_2, x_{12}, r_1, r_2, f_1, f_2$ configuration corresponds to a TWOCONES instance.

Mean results and standard errors in mean (SEMS) were gathered in 100 runs on 30 function instances for a variety of basin sizes, depths and dimensions and results. Each run for a random algorithm initialisation was terminated at 150000 evaluations.

Population positions were initialised in X (ONECONE) and in $\mathcal{F}$ (TWOCONES); PSO particles were initialised with zero velocity. Potentially outflying particles during runs were allowed to move but were not evaluated.

4.1 Parameter settings.

G/LPSO parameters N, w and c were tuned for 30 initialisations of 30 instances of the $r_1 = r_2 = 1, d_1 = -1.0, d_2 = -10.0$ TWOCONES problem. Success probabilities and 95% confidence limits were collected for $N \in [10, 200]$, $w \in [0, 1.0)$ and $c = \{w + \frac{1}{2}, w + 0.75, w + 1\}$. The ranges of w and c correspond to theoretical convergence limits [1]. Recommended w, c values (0.729844 and 1.49618 respectively [2, 8]) and $N = 100$ were found to be in the optimal parameter range, as determined by the confidence limits, for LPSO and were close to optimal for GPSO. These values were chosen for the following PSO experiments in order to provide comparisons with results in the literature.

DE parameters $N \in [10, 200]$, $F \in [0.1, 5]$ and $CR \in [0, 1.0]$ were tuned in a similar set of training runs. A range of optimal settings for success probability was discovered: $N = 50, F = 0.8$ and $CR = 0$ was found to be optimal when mean error was considered and these settings were used for the experiments reported below.

5 RESULTS

5.1 ONECONE

ONECONE is a unimodal, symmetric, function, identical to the square root of Sphere with error, $f(x) - f(x_1)$ equal to the distance between x and the optimum, x_1. The error ϵ of the optimiser best position, and, for the population optimisers, diversity, defined as the mean particle separation from the population centroid, was recorded at steps of 1000 evals for each run on a single function instance and averaged over runs and instances.

5.1.1 LPSO, GPSO, DE. Figs. 3 and 4 shows mean error and diversity in 30 and 80D.

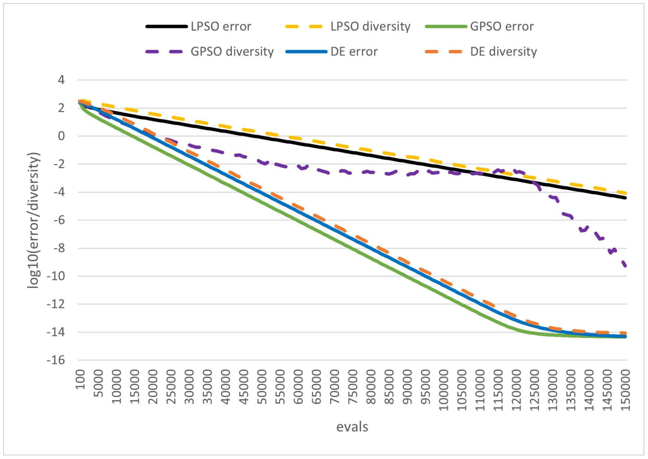


Figure 3: 30D ONECONE convergence and diversity plots for the three population optimisers.

DE and GPSO have the fastest convergence rates in 30D but GPSO is the clear fastest optimiser in 80D. Population diversity scales with optimum separation in each case except for GPSO in 30D; in 80D, GPSO diversity remains high for a greater part of the optimisation indicating that a small subswarm is likely responsible for the convergence. LPSO is, as expected, the slowest optimiser of this unimodal problem, [2]. The local information sharing network favours diversity - as indicated on the plots - but diversity does not enhance unimodal performance.

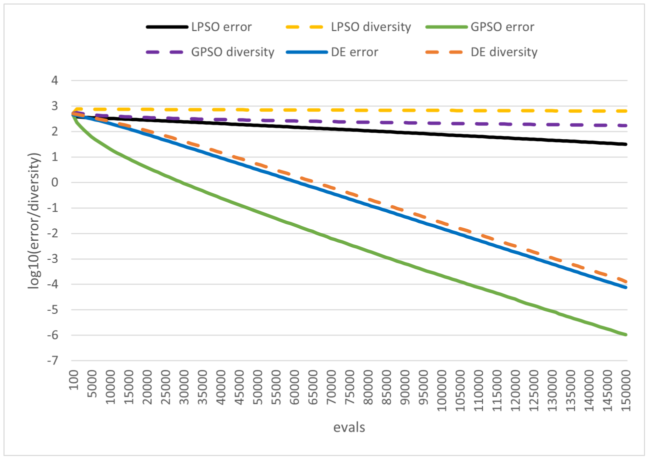


Figure 4: 80D ONECONE convergence and diversity plots for the three population optimisers.

The exponential error convergence $\epsilon(t) = \epsilon(0)10^{-\lambda t}$, in agreement with earlier findings [3]. Linear regression fits are given in Table 1. Since $\epsilon(t)$ is identical to the distance r to the cone tip, the swarms converge as $r = r(0)10^{-\lambda t}$ where r is the distance of the population best position to the optimum. The regression analysis was taken between $t = 1000$ (to allow the swarm settle) and either $t = 150000$ or the first eval such that the mean error dropped below $1e-14$ (hitting the limit of FP arithmetic) i.e. $t_{end} = \min(\{t : f(t) < 1e-14\}, 150000)$.

5.1.2 DHW. A simple realisation of a downhill walker is given in Algorithm 1. Points are picked from the uniform distribution on a hypersphere centred on the current position x of the walker and of radius equal to the current step length until a position is found that is not worse than x. The step length is reduced after a preset number by multiplication by a scale factor in $(0, 1)$. This DHW implementation was trialled on ONECONE with $initial\ step\_length = 10$, $tries = 10$ and $scale\_factor = 0.9$.

Algorithm 1 DHW implementation

```
$attempts \leftarrow 0$
**do**
    $attempts$++
    **if** $attempts > tries$ **then**
        $step\_length *= scale\_factor$
    **end if**
    $y \leftarrow random\_point\_on\_hypersphere(x, step\_length)$
**while** $f(y) > f(x)$
$x \leftarrow y$
```

Figure 5 reports DHW convergence and Table 1 shows regression analysis over the straight part of the convergence leading to convergence at $< 1e-14$ (30D) and $< 1e-13$ (80D); the latter termination condition was chosen because no improvement below an error of $1e-14$ was found. Furthermore, 11 runs in the total of 3000 did not converge in 80D. These failures were removed from the analysis. A comparison of the decay rate λ and the end points of the figures show that the DHW converges much faster then either population optimiser. The DHW realisation was not tuned (the stated values of $initial\ step\_length$, $tries$ and $scale\_factor$ remained at the first guesses) and possibly faster and guaranteed 80D convergence could be achieved with careful parameter selection.

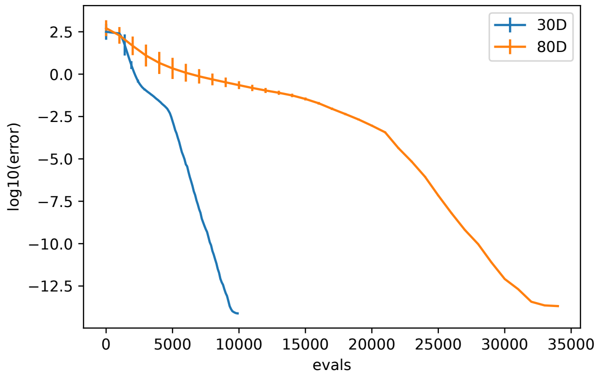
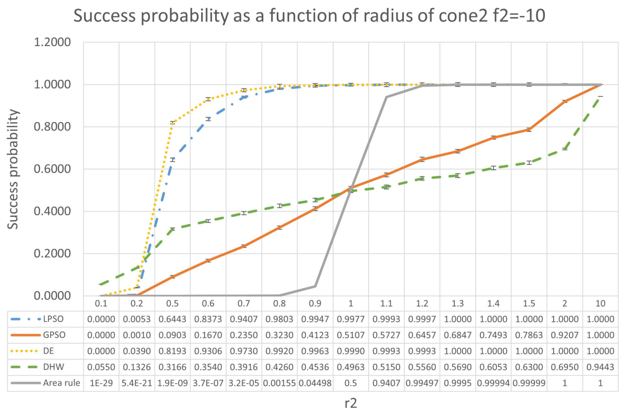


Figure 5: 30D and 80D DHW ONECONE convergence plots. Bars depict standard error in mean.

5.2 TWOCONES

5.2.1 Varying depth and radius. Experiments were conducted in 30D for varying configurations. Basin 1 was held fixed at $r_1 = 1.0, d_1 = -1.0$ and the saddle value was set to $d_{12} = 0.0$ whilst the radii of basin 2 was varied between 0.1 and 10. Three cone 2 depths were investigated.

Figs 6-8 shows algorithm success probability, where success is gauged by optimisation of the optimal basin. The geometric basin surface area measure is included in these plots for comparative purposes.

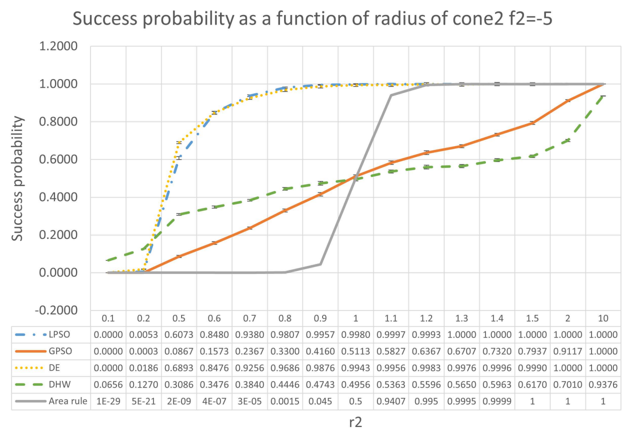


	0.1	0.2	0.5	0.6	0.7	0.8	0.9	1	1.1	1.2	1.3	1.4	1.5	2	10
LPSO	0.0000	0.0053	0.6443	0.8373	0.9407	0.9803	0.9947	0.9977	0.9993	0.9997	1.0000	1.0000	1.0000	1.0000	1.0000
GPSO	0.0000	0.0010	0.0903	0.1670	0.2350	0.3230	0.4123	0.5107	0.5727	0.6457	0.6847	0.7493	0.7863	0.9207	1.0000
DE	0.0000	0.0390	0.8193	0.9306	0.9730	0.9920	0.9963	0.9990	0.9993	0.9993	1.0000	1.0000	1.0000	1.0000	1.0000
DHW	0.0550	0.1326	0.3166	0.3540	0.3916	0.4260	0.4536	0.4963	0.5150	0.5560	0.5690	0.6053	0.6300	0.6950	0.9443
Area rule	1E-29	5.4E-21	1.9E-09	3.7E-07	3.2E-05	0.00155	0.04498	0.5	0.9407	0.99497	0.9995	0.99994	0.99999	1	1

Figure 6: TWOCONES success probabilites, 30D, optimal cone depth $d_2 = -10$.

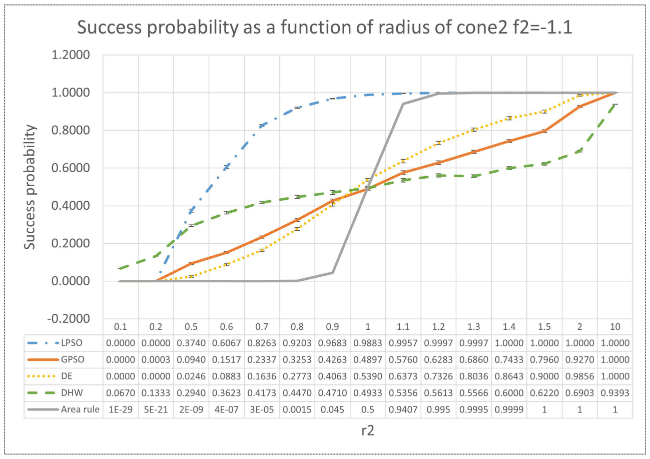
Table 1: One Cone, $m = 1$. Linear regression coefficients for fit to $\log_{10}\epsilon(t) = \log_{10}\epsilon(0) - \lambda t$. * converged trials only

D	Algo	λ	$\log_{10}\epsilon(0)$	r	final regr. eval
30	LPSO	-4.33e-05	2.08	-1.00	150000
	GPSO	-1.32e-04	1.90	-1.00	124000
	DE	-1.22e-04	2.00	-0.995	150000
	DHW	-2.49e-03	N/A	-1.00	8700
80	LPSO	-7.20e-06	2.60	-1.00	150000
	GPSO	-5.21e-05	1.61	-0.997	150000
	DE	-4.62e-05	2.82	-1.00	150000
	DHW	$-9.30e-04^*$	N/A	-0.998^*	32000^*



	0.1	0.2	0.5	0.6	0.7	0.8	0.9	1	1.1	1.2	1.3	1.4	1.5	2	10
LPSO	0.0000	0.0053	0.6073	0.8480	0.9380	0.9807	0.9957	0.9980	0.9997	0.9993	1.0000	1.0000	1.0000	1.0000	1.0000
GPSO	0.0000	0.0003	0.0867	0.1573	0.2367	0.3300	0.4160	0.5113	0.5827	0.6367	0.6707	0.7320	0.7937	0.9117	1.0000
DE	0.0000	0.0186	0.6893	0.8476	0.9256	0.9686	0.9876	0.9943	0.9956	0.9983	0.9976	0.9996	0.9990	1.0000	1.0000
DHW	0.0656	0.1270	0.3086	0.3476	0.3840	0.4446	0.4743	0.4956	0.5363	0.5596	0.5650	0.5963	0.6170	0.7010	0.9376
Area rule	1E-29	5E-21	2E-09	4E-07	3E-05	0.0015	0.045	0.5	0.9407	0.995	0.9995	0.9999	1	1	1

Figure 7: TWOCONES success probabilites, 30D, optimal cone depth $d_2 = -5$.



	0.1	0.2	0.5	0.6	0.7	0.8	0.9	1	1.1	1.2	1.3	1.4	1.5	2	10
LPSO	0.0000	0.0000	0.3740	0.6067	0.8263	0.9203	0.9683	0.9883	0.9957	0.9997	0.9997	1.0000	1.0000	1.0000	1.0000
GPSO	0.0000	0.0003	0.0940	0.1517	0.2337	0.3253	0.4263	0.4897	0.5760	0.6283	0.6860	0.7433	0.7960	0.9270	1.0000
DE	0.0000	0.0000	0.0246	0.0883	0.1636	0.2773	0.4063	0.5390	0.6373	0.7326	0.8036	0.8643	0.9000	0.9856	1.0000
DHW	0.0670	0.1333	0.2940	0.3623	0.4173	0.4470	0.4710	0.4933	0.5356	0.5613	0.5566	0.6000	0.6220	0.6903	0.9393
Area rule	1E-29	5E-21	2E-09	4E-07	3E-05	0.0015	0.045	0.5	0.9407	0.995	0.9995	0.9999	1	1	1

Figure 8: TWOCONES success probabilites, 30D, optimal cone depths $d_2 = -1.1$.

DE is the best algorithm for the larger cone depths but LPSO has a higher probability of finding the optimal cone in the more difficult case when the optimal and sub-optimal cone depths are similar: the superiority of LPSO over DE and GPSO can be attributed to a local communication mechanism which promotes exploration at the cost of slower convergence, as is evident from the ONECONE results.

No algorithm can optimise very small optimal basins ($r_2 = 0.1$). All population algorithms perform better than a solitary downhill walker when the optimal basin is larger than the sub-optimal basin; otherwise, GPSO always underperforms a walker. The $p = 0.5$ result at equi-sized basins suggests that GPSO optimises whichever basin it first happens upon - this suggestion is further explored in the mobility experiments reported below. The similarity of the LPSO and GPSO plots for varying radii at the three depths indicate that the behaviour of these algorithms is linked to problem geometry, but does not depend strongly on the relative catchment areas, as indicated by comparison with the surface area metric. The almost linear relationship to basin radius for GPSO (all depths) shows that this optimiser is sensitive to the spatial extent of the optimal basin and not its area or volume. It seems a kind of line search is taking place, but not, due to the weak area dependence, a progressive line search in which the swarm rolls, like a walker, downhill. The scaling of success with basin linear size suggests that the population is converging on the basins from several sides so that individuals are traversing the critical region. The populations are effectively engaged in line searches that span the basins; this picture would explain the linear dependence on the radius of the optimal basin.

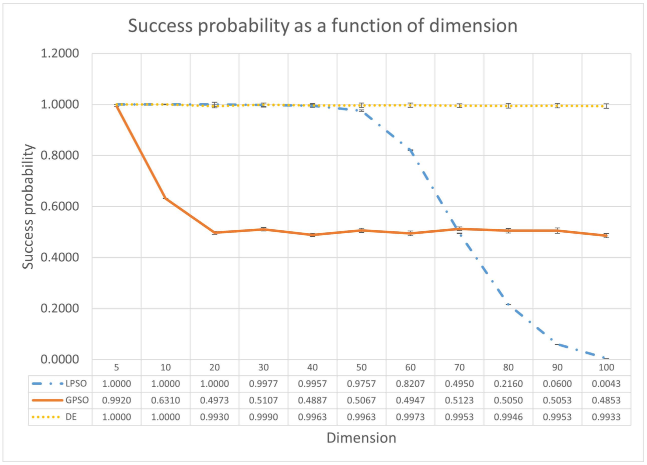
5.2.2 *Varying dimension.* Fig. 9 illustrates the success probability of choosing the optimal cone as a function of the dimension for dimensions ranging from 5 to 100. LPSO and DE almost always find the optimal cone for $D \leq 50$ but the performance of DE, in terms of success probability alone, does not diminish over the range of tested dimensions. GPSO performance falls to a 50% success rate at 20D. LPSO struggles to find the cones above 70D, and in most cases is stuck in the funnel.

Fig. 10 shows diversity and error for a single LPSO run in 90D. Swarm diversity remains high throughout the run. The final error of 26 shows that at termination the swarm has not left the funnel. The swarm will see the funnel as a unimodal problem at these scales and we know that LPSO converges very slowly in higher dimension (single cone results, Fig. 3, 4).

DE and GPSO move more quickly in the funnel, perceived as unimodal, and performance is steady in the range $D = 30 - 100$; the algorithms are not able, however, to distinguish the two cones.

Performance does improve in lower dimensions, though they can only successfully optimise TWOCONES in 5D.

LPSO's high diversity strategy fails in higher dimensions: the swarm possibly requires more particles to cope with the larger volume, scaling as 200^D, of the search space.



Dimension	5	10	20	30	40	50	60	70	80	90	100
LPSO	1.0000	1.0000	1.0000	0.9977	0.9957	0.9757	0.8207	0.4950	0.2160	0.0600	0.0043
GPSO	0.9920	0.6310	0.4973	0.5107	0.4887	0.5067	0.4947	0.5123	0.5050	0.5053	0.4853
DE	1.0000	1.0000	0.9930	0.9990	0.9963	0.9963	0.9973	0.9953	0.9946	0.9953	0.9933

Figure 9: Success probability as a function of dimension. $r_1 = r_2 = 1, d_1 = -1, d_2 = -10$.

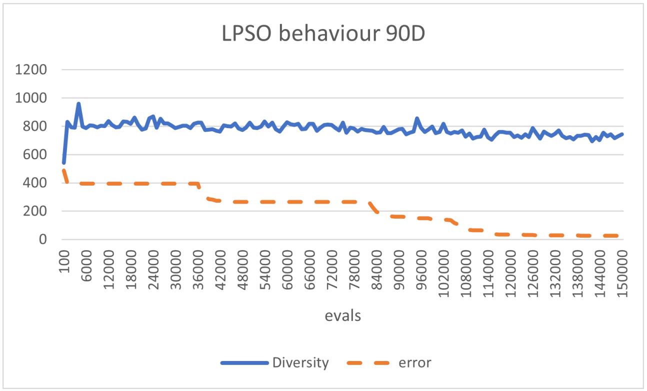


Figure 10: Error and diversity plot for a single LPSO run in 90D. $r_1 = r_2 = 1, d_1 = -1, d_2 = -10$

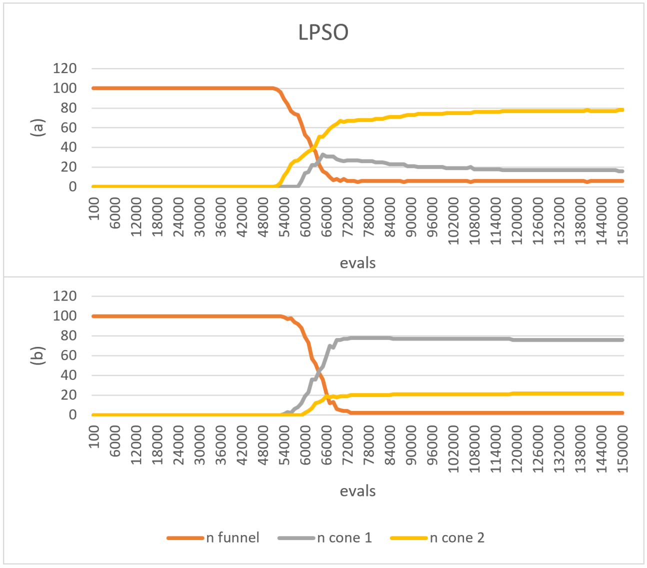


Figure 11: Number of LPSO particles in the funnel and cones for two representative runs, 30D, $r_1 = r_2 = 1, d_1 = -1, d_2 = -10$**.**

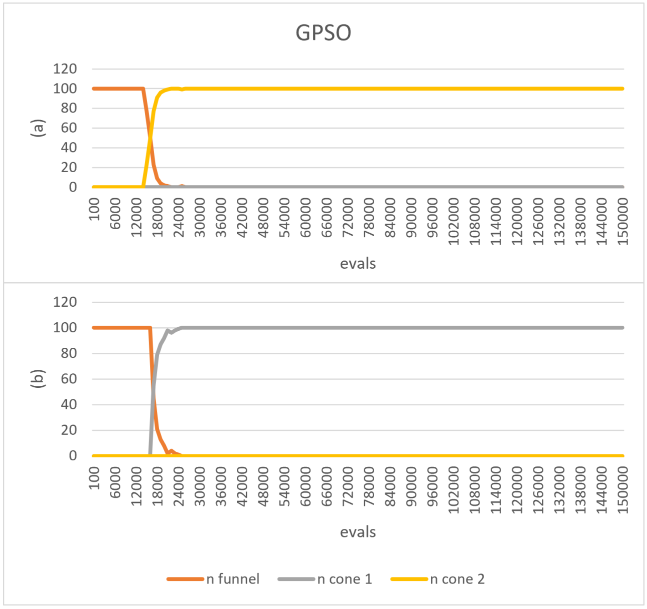


Figure 12: Number of GPSO particles in the funnel and cones for two representative runs, 30D, $r_1 = r_2 = 1, d_1 = -1, d_2 = -10$**.**

5.2.3 Mobility. Mobility is the ability of an algorithm to jump between basins [2]. The relative mobility of one algorithm with another can be gauged by the number of jumps between cones.

Figures 11 - 12 illustrate swarm mobility behaviours for representative outcomes (convergence to cone 1 or cone 2), for 30D equal-sized basins with the global optimum at the centre of cone 2.

Fig. 11 shows LPSO mobility examples when the algorithm found cone 2 (plot (a)) and a rare instance when the algorithm was fooled and settled in cone 1 (plot (b)). Plot (a): funnel particles start to depopulate at around 50000 evals when the swarm finds the region containing the cones. There is a steady rise in occupancy of cone 1 and cone 2, and then a small migration of particles form cone 1 to cone 2 at 66000 evals. The occupancy of cone 1 decreases from this point to the end of the run whilst cone 2 slowly populates, with a final occupancy of 80%. Despite the convergence, some particles (15%) remain in the funnel. Plot (b), an example of an unsuccessful optimisation shows the opposite behaviour although the lesser occupancy does not peak. These plots are evidence of swarm mobility and of population diversity since all three regions (funnel and cones) remain occupied throughout the run.

Table 2: DHW cone attainment probabilities, TWOCONES. (Standard Error in Mean.)

D	funnel prob (SEM)	cone 1 prob (SEM)	cone 2 prob (SEM)
30	0.0 (0.0)	0.503 (9.41e-03)	0.497 (9.41e-03)
80	0.002 (8.84e-04)	0.483 (8.57e-03)	0.515 (8.67e-03)

In contradistinction to LPSO mobility, Fig. 12 depicts representative GPSO behaviour for the two equally probable cases of optimisation of cone 1 or cone 2. The swarm settles entirely in either cone 1 or cone 2; no particles remain in the funnel. The selection of the optimal/suboptimal cone occurs earlier in the run at around 12000 evals and funnel depopulation is swift. The GPSO swarm has a smaller diversity LPSO, optimises faster, but makes more mistakes. The rapid depopulation of the funnel to 100% occupancy of either cone is evidence of negligible mobility.

GPSO, by virtue of its global communication strategy, is a fast unimodal optimisers in 30D (Sec. 5.1.1) but is not mobile. LPSO, with its local communication network, is more mobile - a property that wins out on this 30D bimodal problem.

5.2.4 DHW. The DHW algorithm of Sec. 5.1.2 was trialled on TWOCONES, $r_1 = r_2 = 1, d_1 = -1.0, d_2 = -10, f_{12} = 0$ in 30 and 80D in order to confirm its potential as a possible implementation of the downhill slider whose probability of attaining either basin is 0.5 in any dimension.

Table 2 shows end probabilities averaged over 30 function instances of 100 runs. The agreement with theory is excellent in 30D and good in 80D (one 80D run in 500 does not attain either cone but remains stuck in the funnel). DHW, with its zero mobility and fast unimodal convergence (Fig. 5) is intended as a reference non-population algorithm. The results show that GPSO cannot beat this reference for a simple bimodal 30D problem of equal basin size.

6 CONCLUSIONS

This paper proposes a novel barrier tree benchmark in order to systematically investigate optimiser performance on test cases of known landscape geometry and barrier tree complexity.

Three population algorithms, local and global PSO (L/GPSO) and differential evolution (DE), chosen because of their similarity, popularity and generic nature, were tested on the simplest benchmark instances, symmetric unimodal ONECONE and bimodal TWOCONES, as a demonstration of the feasibility of such a research agenda.

A reference non-population optimisation algorithm is proposed. This downhill walker retains the metaheuristic characteristic of gradient independence and sets a standard that any optimiser that seeks to exploit a collective search should surely beat. In certain cases the behaviour of an idealised walker of infinitesimal stepsize, the downhill slider, can be calculated; otherwise an implementation is necessary. A downhill walker algorithm was tested and was found to match theoretical expectations in a limited set of trials.

Empirical analysis of ONECONE focuses on speed of convergence and diversity evolution; the probability of attaining the optimum basin was employed as the chief TWOCONES measure. All algorithms converge exponentially on a single optimum; GPSO is faster than DE and LPSO, with its local communication strategy, is the slowest unimodal optimiser.

An examination of TWOCONES algorithm performance as a function of optimal basin size shows the clear dominance of LPSO over GPSO and a small superiority of DE for large basin depths whereas LPSO dominates DE for the ore difficult case when the basin depths are similar. GPSO in all cases and DE for near-equal basin depths fail to beat the downhill walker reference when the optimal basin is smaller than the sub-optimal basin. The success probability of GPSO is only 0.5 when the basins have equal size - a disturbing finding given the simplicity of the problem and the popularity of this form of the PSO algorithm. A mobility check shows that GPSO individuals do not basin-hop, an observation attributable to the lower diversity of its populations. The linear nature of GPSO success probability against optimal basis radius suggests that the population does not approach the basins from one side, as the downhill walker does, but perform line searches that traverse the basins, hence resulting in a linear, rather than area, dependence.

No attempt has been made to investigate state-of-the-art implementations and the results may also not generalise to other basin shapes, for example to ellipsoidal topographies of arbitrary conditioning.

The work presented here is offered as an alternative to competitions which pit one algorithm against the other in pursuit of the smallest error within an arbitrary and unstructured set of benchmark functions. We propose that a principled and sequential study of algorithm behaviour on problems of known geometric structure will better advance our understanding of metaheuristic optimisers.

REFERENCES

[1] Tim Blackwell. 2011. A study of collapse in bare bones particle swarm optimization. *IEEE Transactions on Evolutionary Computation* 16, 3 (2011), 354–372.

[2] Tim Blackwell and James Kennedy. 2018. Impact of communication topology in particle swarm optimization. *IEEE Transactions on Evolutionary Computation* 23, 4 (2018), 689–702.

[3] T M Blackwell. 2005. Particle swarms and population diversity. *Soft Computing* 9, 11 (2005), 793–802.

[4] Swagatam Das and Ponnuthurai N Suganthan. 2010. Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems. *Jadavpur University, Nanyang Technological University, Kolkata* (2010), 341–359.

[5] Swagatam Das and Ponnuthurai Nagaratnam Suganthan. 2011. Differential Evolution: A Survey of the State-of-the-Art. *IEEE Transactions on Evolutionary Computation* 15, 1 (2011), 4–31. https://doi.org/10.1109/TEVC.2010.2059031

[6] Stefan Droste, Thomas Jansen, and Ingo Wegener. 1999. Perhaps not a free lunch but at least a free appetizer. In *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 1*. 833–839.

[7] Miguel Duarte, Jorge Gomes, Sancho Moura Oliveira, and Anders Lyhne Christensen. 2017. Evolution of repertoire-based control for robots with complex locomotor systems. *IEEE Transactions on Evolutionary Computation* 22, 2 (2017), 314–328.

[8] Andries Petrus Engelbrecht. 2013. Particle swarm optimization: Global best or local best?. In *2013 BRICS congress on computational intelligence and 11th Brazilian congress on computational intelligence*. IEEE, 124–135.

[9] Ryan Forbes and T Nayeem Mohammad. [n. d.]. Particle swarm optimization on multi-funnel functions. *Computer Aided Optimum Design in Engineering XII* 255 ([n. d.]).

[10] Marcus Gallagher and Bo Yuan. 2006. A general-purpose tunable landscape generator. *IEEE transactions on evolutionary computation* 10, 5 (2006), 590–603.

[11] Nikolaus Hansen, Steffen Finck, Raymond Ros, and Anne Auger. 2009. *Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions.* Ph. D. Dissertation. INRIA.

[12] Tomas Kadavy, Michal Pluhacek, Adam Viktorin, and Roman Senkerik. 2020. SOMA-CL for competition on single objective bound constrained numerical optimization benchmark: a competition entry on single objective bound constrained

numerical optimization at the genetic and evolutionary computation conference (GECCO) 2020. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*. 9–10.

[13] J. Kennedy. 1999. Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. In *In: Proceedings of the 1999, Congress of Evolutionary Computation*, Vol. 3. IEEE Press, 1931–1938.

[14] Ho Min Lee, Donghwi Jung, Ali Sadollah, and Joong Hoon Kim. 2020. Performance comparison of metaheuristic algorithms using a modified Gaussian fitness landscape generator. *Soft Computing* 24, 10 (2020), 7383–7393.

[15] JJ Liang, BY Qu, PN Suganthan, and Alfredo G Hernández-Díaz. 2013. Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization. *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report* 201212, 34 (2013), 281–295.

[16] Jane-Jing Liang, Ponnuthurai Nagaratnam Suganthan, and Kalyanmoy Deb. 2005. Novel composition test functions for numerical global optimization. In *Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005*. IEEE, 68–75.

[17] Chi-Kong Ng and Duan Li. 2014. Test problem generator for unconstrained global optimization. *Computers & operations research* 51 (2014), 338–349.

[18] Adam P Piotrowski. 2015. Regarding the rankings of optimization heuristics based on artificially-constructed benchmark functions. *Information Sciences* 297 (2015), 191–201.

[19] R. Poli, J. Kennedy, and T. Blackwell. 2007. Particle Swarm Optimization: An overview. *Swarm Intellligence* 1 (2007), 33–57.

[20] KV Price, NH Awad, MZ Ali, and PN Suganthan. 2018. Problem definitions and evaluation criteria for the 100-digit challenge special session and competition on single objective numerical optimization. In *Technical Report*. Nanyang Technological University.

[21] Ronald L Rardin and Reha Uzsoy. 2001. Experimental evaluation of heuristic optimization algorithms: A tutorial. *Journal of Heuristics* 7, 3 (2001), 261–304.

[22] Maziar Salahi, Ali Jamalian, and Akram Taati. 2013. Global minimization of multi-funnel functions using particle swarm optimization. *Neural Computing and Applications* 23, 7 (2013), 2101–2106.

[23] Y. Shi and R. Eberhart. 1998. A modified particle swarm optimizer. In *Congress on Evolutionary Computation*. 69–73.

[24] Peter F Stadler. 2002. Fitness landscapes. In *Biological evolution and statistical physics*. Springer, 183–204.

[25] Ponnuthurai N Suganthan, Nikolaus Hansen, Jing J Liang, Kalyanmoy Deb, Ying-Ping Chen, Anne Auger, and Santosh Tiwari. 2005. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. *KanGAL report* 2005005, 2005 (2005), 2005.

[26] Fengyang Sun, Lin Wang, and Bo Yang. 2021. A neuro-diversified benchmark generator for black box optimization. *Information Sciences* 573 (2021), 475–492.

[27] Andrew M Sutton, Darrell Whitley, Monte Lunacek, and Adele Howe. 2006. PSO and multi-funnel landscapes: how cooperation might limit exploration. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*. 75–82.

[28] Lin Wang and Jeff Orchard. 2017. Investigating the evolution of a neuroplasticity network for learning. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 49, 10 (2017), 2131–2143.

[29] Simon Wessing, Mike Preuss, and Günter Rudolph. 2013. Niching by multiobjectivization with neighbor information: Trade-offs and benefits. In *2013 IEEE Congress on Evolutionary Computation*. IEEE, 103–110.

[30] Bin Xin, Jie Chen, and Feng Pan. 2009. Problem difficulty analysis for particle swarm optimization: deception and modality. In *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*. 623–630.

[31] Qingfu Zhang and Heinz Muhlenbein. 2004. On the convergence of a class of estimation of distribution algorithms. *IEEE Transactions on evolutionary computation* 8, 2 (2004), 127–136.