# Differential Evolution with Thresheld Convergence

Antonio Bolufé-Röhler
School of Mathematics and Computer Science
University of Havana
Havana, Cuba
bolufe@matcom.uh.cu

Suilan Estévez-Velarde
School of Mathematics and Computer Science
University of Havana
Havana, Cuba
s.estevez@lab.matcom.uh.cu

Alejandro Piad-Morffis
School of Mathematics and Computer Science
University of Havana
Havana, Cuba
a.piad@lab.matcom.uh.cu

Stephen Chen
School of Information Technology
York University
Toronto, Canada
sychen@yorku.ca

James Montgomery
Research School of Computer Science
Australian National University
Canberra, Australia
james.montgomery@anu.edu.au

***Abstract*—During the search process of differential evolution (DE), each new solution may represent a new more promising region of the search space (exploration) or a better solution within the current region (exploitation). This concurrent exploitation can interfere with exploration since the identification of a new more promising region depends on finding a (random) solution in that region which is better than its target solution. Ideally, every sampled solution will have the same relative fitness with respect to its nearby local optimum – finding the best region to exploit then becomes the problem of finding the best random solution. However, differential evolution is characterized by an initial period of exploration followed by rapid convergence. Once the population starts converging, the difference vectors become shorter, more exploitation is performed, and an accelerating convergence occurs. This rapid convergence can occur well before the algorithm's budget of function evaluations is exhausted; that is, the algorithm can converge prematurely. In *thresheld convergence*, early exploitation is "held" back by a threshold function, allowing a longer exploration phase. This paper presents a new adaptive *thresheld convergence* mechanism which helps DE achieve large performance improvements in multi-modal search spaces.**

***Keywords—thresheld convergence; differential evolution; exploration; exploitation; crowding; niching; multi-modal optimization***

## I. INTRODUCTION

When optimizing in multi-modal search spaces, the effectiveness of a heuristic search technique depends on its ability to both detect good attraction basins and to find the best solutions within these basins. An attraction basin represents the region of a search space that will lead to the same (local) optimum when greedy local search is used. The process of detecting the best basins can be called exploration, and the process of finding the local optimum within a basin can be called exploitation.

Measuring the precise fitness of a basin is not possible without using local search to find the actual optimum. Thus, to estimate the promise of a basin, heuristic search is usually based on (random) sample solutions. If a new sample solution is found to be better than the existing representative of an already explored basin, then the search process is likely to focus on the newly discovered basin. However, it is hard to determine when a new solution is from an unexplored basin. If the new solution happens to be from an already known basin, then it can be considered exploitation. As a result, many population-based heuristic search techniques simultaneously perform exploration and exploitation, i.e. the two processes are not performed in separate phases. Importantly, concurrent exploration and exploitation may interfere with each other.

The fitness of an attraction basin can be estimated through the fitness of solutions sampled from within that basin. If different basins are sampled randomly, there can be reasonable expectations that the best solutions correspond to the best basins. However, if one basin is sampled more than the others, then above-average solutions can be expected to be found from that basin. In a comparison of an above-average solution from one basin and random solutions from other basins, the expectation that the fittest of these solutions is from the best basin may no longer be reasonable. Importantly, exploitation is likely to find above average solutions from a basin. When

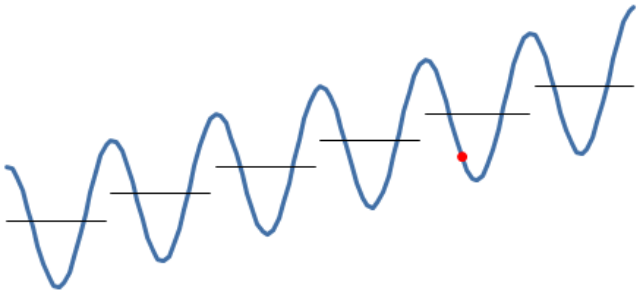

Fig. 1. The horizontal lines represent the average/expected fitness of random sample solutions in each attraction basin. If an attraction basin is represented by a better-than-average solution (see dot), a random solution from a better attraction basin may no longer have a better expected fitness.

compared against (sample solutions from) other basins, the (over) exploited basin (represented by an above-average solution) may look more promising than basins which are actually better (see Fig. 1). This "uneven" comparison can cause the search process to focus on poor basins/regions of the search space.

Differential evolution (DE) [1] is not immune to this interference between exploration and exploitation. In a typical configuration, DE's search is characterized by an initial period of exploration. The effectiveness of the exploration will strongly depend on the diversity of its population. However, its elitist selection mechanism rapidly focuses the population around the best found basins. Once the population starts converging, the difference vectors become shorter, more exploitation is performed, and an accelerating convergence occurs. This rapid convergence can occur well before the algorithm's budget of function evaluations is exhausted; that is, the algorithm can converge prematurely.

Several techniques such as crowding and niching have been developed in the past decades to slow or prevent convergence on multi-modal functions [2][3]. In niching, the population is split across different parts of the search space by using multiple, minimally-interacting subpopulations [4]. In crowding, each new solution is compared against a subset of the population, and it replaces the most similar member from this subset if the new solution is better. However, a small population subset can cause "replacement errors", and larger population subsets can cause large increases to the required computational effort.

Recent work [6][7][8][9] has identified a new way to efficiently implement a convergence control mechanism similar to crowding. Named "*thresheld convergence*", its goal is to delay local search and thus prevent "uneven" sampling from attraction basins. Convergence is "held" back as (local) search steps that are less than a threshold function are disallowed. As the threshold function decays towards zero, greedier local search steps are allowed. Conversely, until the threshold is sufficiently small, the search technique is forced to focus on the global search aspect of finding the best region/attraction basin of the search space in which a local optimum will eventually be found.

The optimal threshold size is directly related to the number and size of the attraction basins. On unimodal functions, for instance, no control mechanism is needed and the threshold can be zero. Conversely, more exploration and a stronger control of convergence are required on highly multi-modal functions, so a larger and/or more slowly decreasing threshold may be appropriate. Depending on the topology of different search regions, the optimal threshold may also vary at different times during the search process on the same function. This paper presents a new adaptive threshold function which improves the performance and robustness of *thresheld convergence* when used with DE to solve multi-modal functions.

A brief background on the development of threshold functions and other diversification techniques is provided in Section II. Differential evolution with *thresheld convergence* is presented in Section III. The experimental design and first results are presented in Section IV. A simple and robust adaptive threshold function is described in Section V. In Section VI, a new design of *thresheld convergence* for DE which further improves the performance is presented. Finally, all of these developments are discussed in Section VII before a summary is given in Section VIII.

## II. BACKGROUND

Differential evolution is a population-based evolutionary algorithm. In each generation, every member in a population of *Np* solutions is considered as a *target* for replacement by a newly generated trial point. How this new solution is generated varies on the variant of DE. The remainder of this work considers only the highly common and frequently effective variant labeled DE/rand/1/bin [10]. This version generates a new solution (trial point) in two steps: first a new point $v$ is generated using difference vectors (1), and then this point $v$ is recombined with the *target* using uniform crossover (2).

$$v = x_{base} + F \cdot (x_{r1} - x_{r2}) \tag{1}$$

In (1), $x_{r1}$, $x_{r2}$, and $x_{base}$ (or simply *base*) are distinct, randomly selected solutions from $P \backslash \{x_{target}\}$, and $F$ is the scaling factor, typically in (0, 1]. Uniform crossover is performed on $v$ based on the parameter $Cr \in [0, 1)$ in (2) where $R \in [0, 1)$ is a uniform random number and $I$ is the randomly selected index of a component that must be mutated to ensure that $u$ is distinct from $x_{target}$. The *target* is replaced if the new solution is as good or better.

$$u^j = \begin{cases} v^j & \text{if } R^j \le Cr \text{ or } j = I, \\ x^j_{target} & \text{if } R^j > Cr \text{ and } j \ne I, \end{cases} \tag{2}$$

DE's search behavior results from a complex interaction of population size *Np*, scale factor *F*, and crossover rate *Cr* [11]. Assuming a fixed budget of function evaluations (FEs), as is common in experimental and practical use of heuristic search techniques, a smaller population allows more (potential) updates to each individual. With more generations, DE is more likely to converge, but at the cost of reduced exploration and an increased risk of premature convergence. Conversely, larger populations lead to higher exploration but also to fewer generations and slower convergence.

Small values of *F* shorten the difference vectors and can also lead to premature convergence. Larger values of the scaling factor *F* lead to larger steps away from the *base* vector and a more explorative behavior. With low values of *Cr*, premature convergence is not a concern since DE members will effectively conduct independent searches making small moves from the *target*. This type of global search is not likely to be successful in complex multi-modal search spaces as little information about distant areas is transmitted among individuals. Therefore, a high *Cr* is usually required to guarantee convergence in multi-modal search spaces (e.g. $Cr \approx 0.9$) [11]. Previous work has also shown that a large *F* is particularly important when *Cr* is high [12].

When *Cr* is high, "moves" in DE are largely conducted from the *base*, so new candidate solutions are typically closer to *base* than the *target* they replace – this leads to convergence.

Once convergence starts, nearby solutions can create short difference vectors which lead to the creation of more solutions close to the existing solutions – i.e. crowding begets more crowding. This "cascading convergence" is a characteristic of DE, and it can cause the algorithm to convergence prematurely [7].

Crowding [5] and niching [3] techniques can be applied to prevent solutions from becoming too close in solution space. In DE, niching includes speciation-based methods [13] which split the population by identifying "dominant" individuals and then determining clusters of similar individuals near them. An alternative approach is multi-population DE [14] which divides the search space into a number of non-overlapping subspaces with separate subpopulations for each.

The purpose of crowding in DE is to prevent solutions from becoming too close in solution space. It slows the rate of convergence and may help the identification of multiple attraction basins. In [5], two algorithms are described which use different mechanisms to achieve this: SharingDE and CrowdingDE. In SharingDE, the reported quality of a solution diminishes with its proximity to other solutions. In CrowdingDE, instead of comparing a new candidate solution against the *target*, the solution is compared to the population member nearest to it in the search space. If the new solution is better, it replaces that nearby solution immediately.

The main weaknesses with crowding is that it is either slow (requiring as many distance calculations as members in the population to find the nearest neighbor) or prone to "replacement errors" (if crowding is applied to only a small subset of the population) [5]. Another disadvantage of each of these techniques is that they work on solutions after they have been generated and evaluated. In many real-world problems of interest, solution evaluation can be computationally expensive, so a convergence control technique that can intervene prior to evaluation could be beneficial.

Considering the natural convergence behavior of DE, an alternative approach that tackles both disadvantages has been proposed in [7]. Given relatively high values of *Cr*, new solutions are likely to be generated near to the *base* used in its construction. By requiring new solutions to be a minimum distance from their *base*, it is possible to avoid crowding. This technique only requires a single distance measurement to be calculated. New solutions which do not fulfill this requirement are not evaluated.

## III. Differential Evolution with Thresheld Convergence

With recommended parameters of DE (i.e. $Cr \approx 0.9$ and $F < 1$), new solutions are likely to be generated near the *base* [15]. However, if the new solution (or simply *new*) is better than the *target*, it won't replace *base* – it will replace the *target* instead. As a result, *new* and *base* will start to form a crowd. Once solutions start crowding, difference vectors can become shorter. Shorter difference vectors can generate more new solutions even nearer to *base*. The nearer the new solutions get to the *base*, the more crowding that will occur and the shorter the difference vectors can become. This autocatalytic process, where crowding begets more crowding, leads to a "cascading convergence" in DE.

Fig. 2 illustrates the effect of cascading convergence. It shows a plot of the maximum, average, and minimum length of difference vectors in a standard DE implementation (DE/rand/1/bin). The plot is the average value of 25 independent runs on the Rastrigin function in a 20 dimensional search space. Parameters were set as recommended in [16] to have $Np = 20$, $F = 0.8$, and $Cr = 0.9$. It can be observed that before the system reaches generation 500 (i.e. 10,000 FEs), the lengths of the difference vectors have all dropped near zero. When this situation occurs, the system will perform no further exploration. In related works (e.g. [6][7][17]), experiments are done with a fixed budget of 5000*DIM function evaluations. In a search space with 20 dimensions (DIM = 20), this budget would allow a total of 5000 generations, but standard DE has largely converged before using even 10% of the available function evaluations.

To avoid this "cascading convergence", a simple control mechanism may be added. In [7], three alternate convergence control schemes are presented. In each, a decreasing threshold attempts to control the distance between *new* and *base*.

1) Prevent moves produced by difference vectors whose magnitude ($\| x_{r1} - x_{r2} \|$) is less than the threshold.

2) Prevent moves where the actual distance between *new* and *base* is less than the threshold, i.e. $\| new - base \| < threshold$.

3) As in (2), but instead of preventing the move, if $f(new) < f(base)$, have *new* replace *base*.

Scheme (1) is indirect since it attempts to control the distance between *new* and *base* by controlling the magnitude of the difference vector. The third scheme is more similar to crowding, but it is also potentially complex. In DE, *base* is also a *target* for replacement, and its own new candidate solution

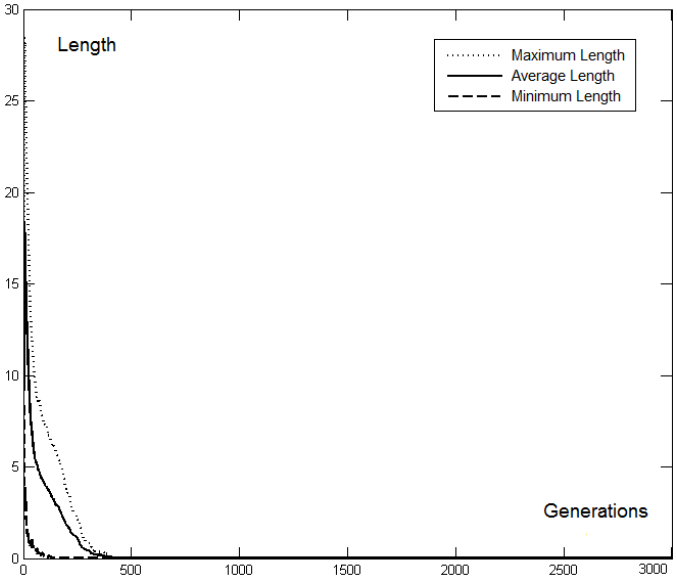

Fig. 2. Maximum, average, and minimum length of difference vectors in each generation of standard DE

TABLE I
BBOB FUNCTIONS

| Set | fn | Function Name | Attribute | | |
|---|---|---|---|---|---|
| | | | s | u | gs |
| 1 | 1 | Sphere | X | X | X |
| | 2 | Ellipsoidal, original | X | X | X |
| | 3 | Rastrigin | X | | X |
| | 4 | Büche-Rastrigin | X | | X |
| | 5 | Linear Slope | X | X | |
| 2 | 6 | Attractive Sector | | X | |
| | 7 | Step Ellipsoidal | | | X |
| | 8 | Rosenbrock, original | | | |
| | 9 | Rosenbrock, rotated | | | |
| 3 | 10 | Ellipsoidal, rotated | | X | X |
| | 11 | Discus | | X | X |
| | 12 | Bent Cigar | | X | |
| | 13 | Sharp Ridge | | X | |
| | 14 | Different Powers | | X | |
| 4 | 15 | Rastrigin, rotated | | | X |
| | 16 | Weierstrass | | | X |
| | 17 | Schaffers F7 | | | X |
| | 18 | Schaffers F7, moderately ill-conditioned | | | X |
| | 19 | Composite Griewank-Rosenbrock F8F2 | | | X |
| 5 | 20 | Schwefel | | | |
| | 21 | Gallagher's Gaussian 101-me Peaks | | | |
| | 22 | Gallagher's Gaussian 21-hi Peaks | | | |
| | 23 | Katsuura | | | |
| | 24 | Lunacek bi-Rastrigin | | | |

Names and selected attributes of the 24 functions in the BBOB problem set – separable (s), unimodal (u), global structure (gs).

may be better than the *new* with which this scheme would use to replace *base*. Given the complexities of implementing scheme (3) under a standard DE generational model and the indirect nature of scheme (1), the second approach was selected for implementation and testing in [7].

By working off of the distance between *new* and *base*, *thresheld convergence* can prevent crowding before evaluating the candidate new solution. If the distance between *new* and *base* is smaller than the threshold, *new* is rejected and an alternative move is generated for the same *target* (randomly selecting a new *base*, $x_{r1}$, and $x_{r2}$). If no acceptable move can be generated after five attempts, no child solution is produced for that *target* in that generation. However, in the current implementation (as in [7]), this solution still counts as a function evaluation.

The threshold is initially set to a fraction of the search space diagonal, and it is updated (decreased) over the course of a run by following a decay rule (3). In (3), $d$ is the diagonal of the search space, $n$ is the total number of generations, and $i$ is the current generation. The parameter $\alpha$ determines the initial threshold and $\gamma$ controls the decay rate.

$$threshold = \alpha * d * ([n-i]/n)^{\gamma} \quad (3)$$

## IV. EXPERIMENTAL DESIGN

The following analysis of differential evolution with *thresheld convergence* focuses on two sets from the Black-Box Optimization Benchmarking (BBOB) functions [18]: set 4, multi-modal functions with adequate global structure, and set 5, multi-modal functions with weak global structure. In Table I, some key attributes of the functions (fn) are indicated, e.g. whether or not they are separable (s), unimodal (u), or have (adequate) global structure (gs). Different instances can be generated for each function, e.g. each instance has a different optimal value (shifted in f-space). To be consistent with related results (e.g. [6][7][17]), the experiments perform 25 independent trials on each function (5 trials on each of the first 5 instances) with a fixed limit of 5000*DIM function evaluations. These experiments also use DIM = 20 dimensions. It should be noted that *thresheld convergence* is a technique specifically designed to improve performance on multi-modal functions. Therefore, experiments focus on BBOB sets 4 and 5 – multi-modal functions with strong and weak global structure, respectively.

The initial set of experiments was designed to analyze the performance of differential evolution with *thresheld convergence* (DETC) using different values for $\alpha$ and $\gamma$. A pair-wise comparison is presented in Tables II and III. The reported values are the relative improvements (%-diff = (a-b)/a) achieved by DETC versus standard DE with *F=0.8* and *Cr=0.9*. A positive value indicates that DETC (b) outperforms standard DE (a). Table II shows the improvement of DETC with $\gamma$ = 3 and $\alpha$ = 0.01, 0.02, 0.05, 0.1, 0.2, and 0.5. Experiments were also conducted for $\gamma$ = 2 and $\gamma$ = 1 (the best results in [7] were with $\gamma$ = 1, 2, or 3), but they were

TABLE II
EFFECT OF INITIAL THRESHOLD SIZE

| BBOB | $\alpha$ | | | | | |
|---|---|---|---|---|---|---|
| fn | 0.01 | 0.02 | 0.05 | 0.1 | 0.2 | 0.5 |
| 15 | 43.5% | 49.6% | 58.8% | **70.7%** | 65.8% | 67.9% |
| 16 | 55.0% | 57.9% | 65.5% | 64.2% | **66.3%** | 46.9% |
| 17 | 58.8% | 74.4% | 79.6% | 81.9% | **85.0%** | 82.3% |
| 18 | 53.3% | 66.2% | 70.5% | 63.0% | 65.6% | **76.7%** |
| 19 | **23.2%** | 19.7% | 21.8% | 15.8% | 17.8% | 15.6% |
| 15-19 | 46.7% | 53.6% | 59.2% | 59.1% | 60.1% | 57.9% |
| 20 | **24.0%** | 16.1% | 14.8% | 13.6% | 18.1% | 19.9% |
| 21 | -64.6% | -66.8% | **22.8%** | -22.9% | -25.1% | -23.1% |
| 22 | 32.8% | 19.4% | 43.9% | 34.7% | 27.9% | **73.8%** |
| 23 | 30.9% | **36.7%** | 31.3% | 30.7% | 28.6% | 30.1% |
| 24 | 32.6% | 28.4% | **34.3%** | 30.7% | 30.1% | 25.6% |
| 20-24 | 11.2% | 6.4% | 29.4% | 17.4% | 15.9% | 25.3% |

Mean relative improvement (%-diff) of DETC over standard DE, with different values of $\alpha$ and $\gamma$=3. The optimum $\alpha$ varies widely depending on the structure of the search space.

TABLE III
EFFECT OF DECAY RATE

| BBOB | $\gamma$ | | | | |
|---|---|---|---|---|---|
| fn | 1 | 2 | 3 | 4 | 5 |
| 15 | 61.4% | 56.1% | 58.8% | 59.2% | **61.6%** |
| 16 | 65.7% | 66.9% | 65.5% | **67.4%** | 59.6% |
| 17 | 71.9% | 79.7% | 79.6% | **87.3%** | 81.7% |
| 18 | 56.3% | 65.7% | 70.5% | 67.4% | **71.2%** |
| 19 | 16.4% | 18.2% | 21.8% | 23.1% | **23.8%** |
| 15-19 | 54.3% | 57.3% | 59.2% | 60.9% | 59.6% |
| 20 | 13.1% | **24.0%** | 14.8% | 16.6% | 15.6% |
| 21 | -74.1% | -3.2% | **22.8%** | 0.8% | -6.1% |
| 22 | 50.3% | **63.7%** | 43.9% | 58.5% | 52.5% |
| 23 | 31.7% | 27.6% | 31.3% | 35.2% | **41.8%** |
| 24 | 13.8% | 24.4% | **34.3%** | 28.4% | 32.6% |
| 20-24 | 7.0% | 27.3% | 29.4% | 27.9% | 27.3% |

Mean relative improvement (%-diff) of DETC over standard DE, with different values of $\gamma$ and $\alpha$=0.05. The optimum $\gamma$ shows less variation than the optimum $\alpha$ in Table II.

consistently a little worse, so they have been omitted for clarity and brevity. Results in Table II show that the best performance of DETC varies widely with $\alpha$.

Table III shows a similar experiment but with varying values of $\gamma$ = 1, 2, 3, 4, and 5. The parameter α was set to 0.05 (which led to the best results in Table II). Results in Table III are more consistent, and the best results are achieved for $\gamma$ = 3, 4, and 5. From these results, it can be noticed that $\alpha$ is a more important parameter when *thresheld convergence* is added to DE. The large variation in performance for different values of $\alpha$ (see Table II) suggests that *thresheld convergence* could benefit from an adaptive mechanism that adjusts the threshold value in a more problem-specific manner.

## V. An Adaptive Threshold Function

From the results in Tables II and III, it can be concluded that the key parameter affecting the performance of *thresheld convergence* is $\alpha$. Each of the tested values reports the best results on at least one function (see Table II). It is hypothesized that the ideal threshold value is directly related to the size of attraction basins and their location in the search space (i.e. the topology of the function). If so, it may then be possible to improve the average performance through the design of an adaptive threshold function which would require little or no parameter tuning.

An ideal threshold function should promote exploration during the early stages of the search and steadily decrease during the later generations to permit a final convergence. Due to the strong convergence behavior exhibited by standard DE, an adaptive threshold must first keep a high enough threshold to promote exploration. A high value should be kept as long as the exploration leads to the discovery of new (unexplored) attraction basins. Once exploration stops detecting new promising regions, the threshold should be decreased to allow DE to converge.

A simple characteristic which can help determine when a given threshold has an adequate value is the number of replacements in the population during a generation. As long as DE keeps finding solutions which improve at least one member of the population, the search can be considered successful. Based on this idea, an adaptive threshold strategy can be implemented simply by reducing the threshold value if no replacements occur in a given generation. In (4), $i$ represents the current generation.

$$threshold_i = \begin{cases} threshold_{i-1} & \text{if } replacements_{i-1} > 0 \\ threshold_{i-1} \cdot \beta & \text{if } replacements_{i-1} = 0 \end{cases} \quad (4)$$

The decay factor $\beta$ becomes an important parameter to adjust in this strategy. A good selection of $\beta$ should promote exploration in early generations and allow convergence at the end. To allow convergence, the decay factor $\beta$ has to assure that the threshold reaches zero (or a very small value) during the final generations. If it is assumed that after each generation the threshold will be updated, then this decay rate can be estimated using (5), in which $n$ is the total number of generations and $\varepsilon$ is a value small enough to allow convergence. For instance, with $\varepsilon$ =1e-10, $\alpha$=0.01, and $n$=5000 generations (as in the current experimental design), $\beta$ will be equal to 0.9951. In general, good results were obtained with a "decay factor" of $\beta$ = 0.995 – the threshold decreases by 0.5% after any generation in which no improvements are made. Unreported experiments also tried decay factors of $\beta$ = 0.9975 and $\beta$ = 0.99, but worse results were achieved.

$$\beta = \left( \frac{\varepsilon}{initial\ threshold} \right)^{\frac{i}{n}} \quad (5)$$

## VI. A New Design for Thresheld Convergence

In differential evolution, new solutions can be generated near the *base*. Thus, the addition of *thresheld convergence* to DE aims to guarantee a minimum distance between *base* and *new*. To achieve this minimum distance, five attempts are performed with different *base*, $x_1$, and $x_2$ vectors. If no acceptable move can be generated after the five attempts, then no child solution is produced for that *target* in that generation. However, in the previous [7] and current implementation, this solution still counts as if it had been evaluated. A consequence of this approach is that a large number of function evaluations may not be performed. This "waste" of available FEs (from a fixed budget) can decrease the benefits of *thresheld convergence*. Another disadvantage is that allowing up to four additional attempts to create an acceptable solution adds to the processing time of the algorithm.

A new design of *thresheld convergence* for DE aims to avoid the generation of additional solutions. In DE, the relative direction of *new* with respect to *base* is determined by its internal mechanisms, i.e. the difference vectors and uniform crossover operations. The size is mainly determined by the length of the difference vector. To control the distance between *new* and *base*, it is not strictly necessary to reject solutions which are too close. Instead, the new solution can be "pushed" a threshold distance away from *base* while keeping the same direction with respect to *base*.

In the new design of *thresheld convergence*, if *new* is not a threshold distance away from *base*, no more attempts are performed (see Algorithm 1). Instead, the direction from *base* to *new* is calculated (subtracting *base* from *new*), and the directional vector is normalized. The new solution is then generated by taking a step from *base* along this direction. The length of this step is set equal to the current threshold value. This modification keeps the original search direction of DE mechanisms (and thus its search logic), but it adjusts the step size once crowding starts to occur.

The new adaptive *thresheld convergence* systematically avoids crowding by controlling the distance between *base* and

**Algorithm 1** New thresheld convergence implementation

**if** ||*new*-*base*|| < *threshold*

  *direction* = Normalize(*new* – *base*)

  *new* = *base* + *threshold* * *direction*

**end if**

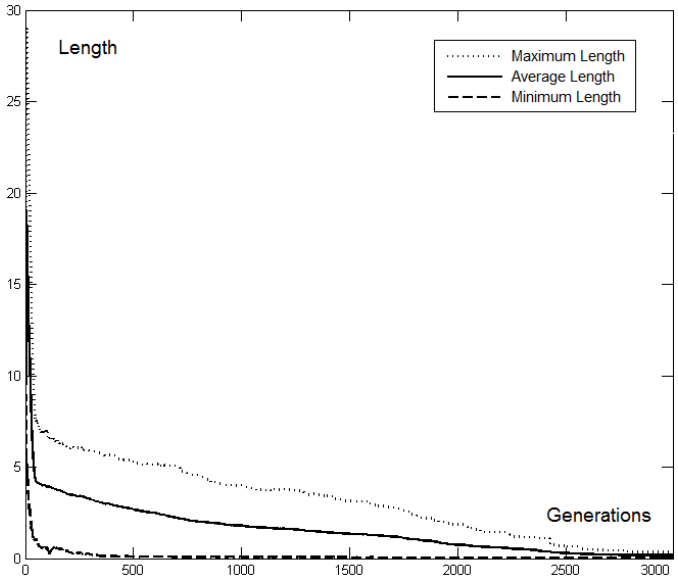

Fig. 3. Maximum, average, and minimum length of difference vectors in each generation of DE with the new adaptive *thresheld convergence*.

*new*. Fig. 3 shows that by reducing crowding, it is possible to slow down the "cascading convergence" of DE. It can be noticed that the length of the difference vectors does not approach zero until generation 3000. Compared to Fig. 2, this new search behavior clearly shows more opportunity for exploration and better use of the available computational resources (i.e. function evaluations).

The results in Table IV again show the percent difference (%-diff = (b-a)/b) in the mean performance between differential evolution with the new adaptive *thresheld convergence* (a) and standard DE (b). Results are shown with different initial threshold values of $\alpha$ = 0.01, 0.02 0.05, 0.1, 0.2, and 0.5, and a "decay factor" of $\beta$ = 0.995. It can be seen that DE with the new design of adaptive *thresheld convergence* shows a lower variation in performance with respect to the values of $\alpha$. The largest improvements are achieved with $\alpha$=0.1 and $\alpha$=0.2. This new design leads to better results than the original version of DE with *thresheld convergence*, and it has a lower processing time.

TABLE IV
EFFECT OF INITIAL THRESHOLD SIZE

| BBOB | $\alpha$ | | | | | |
|---|---|---|---|---|---|---|
| fn | 0.01 | 0.02 | 0.05 | 0.1 | 0.2 | 0.5 |
| 15 | 50.0% | 47.5% | 43.2% | 42.0% | **67.0%** | 63.7% |
| 16 | 79.8% | 78.3% | **83.0%** | 78.2% | 81.0% | 82.8% |
| 17 | 64.7% | 73.0% | 66.2% | 77.2% | **91.3%** | 71.7% |
| 18 | 51.6% | 46.4% | 59.0% | 60.5% | **78.9%** | 59.3% |
| 19 | 46.7% | 42.4% | 35.7% | **82.1%** | 36.2% | 27.3% |
| 15-19 | 58.6% | 57.5% | 57.4% | 63.4% | 70.9% | 61.0% |
| 20 | 15.0% | 9.0% | **27.0%** | 20.0% | 26.4% | 7.3% |
| 21 | 6.0% | -35.7% | 8.6% | **16.2%** | -40.6% | -34.0% |
| 22 | 26.2% | 34.8% | 36.3% | 11.2% | 37.8% | **41.2%** |
| 23 | 81.4% | 80.7% | **83.0%** | 80.7% | 79.1% | 65.6% |
| 24 | 69.9% | 69.5% | 65.1% | 66.1% | **75.5%** | 74.1% |
| 20-24 | 39.7% | 31.7% | 44.0% | 38.9% | 35.7% | 30.8% |

Mean relative improvement (%-diff) through the addition of the new adaptive *threshold convergence* to standard DE.

Overall, the new design of *thresheld convergence* (with the adaptive threshold function) provides significant improvements on 14 of 24 functions when compared to standard DE on the full BBOB set (see Table V). As expected, larger improvements are achieved on multi-modal functions. The *thresheld convergence* technique has been specifically design to slow convergence and allow a more thorough exploration of the search space. On unimodal functions, a fast convergence is both safe and desired once the gradient direction has been identified. Nevertheless, in difficult unimodal functions (e.g. ill-conditioned, not separable, or with plateaus), *thresheld convergence* may avoid stalling and thus increase performance.

The final overall results in Table V are for a single set of parameters ($\alpha$ = 0.1 and a decay factor of $\beta$ = 0.995). Since no single set of parameters can provide the best results in all scenarios, some tuning can be expected [19]. However, this new version simplifies the parameter tuning by providing a meaningful method to select $\beta$ and a smaller interval of recommended values for $\alpha$ (and more consistent results across all values of $\alpha$ than the previous design). This improved consistency is an indicator that the adaptive strategy efficiently exploits some characteristics of the function's landscape.

In general, the results with *thresheld convergence* are better and more consistent on the multi-modal functions with global structure (set 4 – BBOB 15-19) than without global structure (set 5 – BBOB 20-24). Final results in Table V show that the overall results for set 4 improve by 63.4% and 38.9% for set 5. These results confirm that the performance of differential evolution

TABLE V
FINAL RESULTS

| | standard DE | | with thresholds | | %-diff | p-value |
|---|---|---|---|---|---|---|
| | mean | std dev | mean | std dev | | |
| 1 | 0.00e+0 | 0.00e+0 | 0.00e+0 | 0.00e+0 | 0.0% | |
| 2 | 0.00e+0 | 0.00e+0 | 0.00e+0 | 0.00e+0 | 0.0% | |
| 3 | 5.59e+1 | 2.15e+1 | **2.68e+1** | 1.28e+1 | **52.1%** | 0.00 |
| 4 | 8.17e+1 | 3.19e+1 | **4.24e+1** | 1.54e+1 | **48.0%** | 0.00 |
| 5 | 2.02e+0 | 7.11e+0 | 0.00e+0 | 0.00e+0 | 100.0% | 0.08 |
| 1-5 | | | | | 25.0% | |
| 6 | 1.87e+1 | 3.86e+1 | **7.06e−6** | 2.67e−5 | **100.0%** | 0.01 |
| 7 | 1.34e+1 | 8.04e+0 | **3.68e+0** | 2.41e+0 | **72.5%** | 0.00 |
| 8 | 3.35e+0 | 3.27e+0 | 6.97e+0 | 5.06e+0 | -108.3% | 0.00 |
| 9 | 1.26e+1 | 2.70e+0 | 1.46e+1 | 1.18e+1 | -16.1% | 0.20 |
| 6-9 | | | | | 12.0% | |
| 10 | 2.60e+2 | 2.03e+2 | 3.31e+2 | 2.09e+2 | -27.2% | 0.11 |
| 11 | 3.08e+0 | 3.29e+0 | **1.79e+0** | 1.43e+0 | **42.1%** | 0.04 |
| 12 | 1.22e+4 | 6.09e+4 | 5.39e+0 | 9.82e+0 | 100.0% | 0.16 |
| 13 | 1.03e+1 | 1.45e+1 | **4.65e+0** | 5.51e+0 | **54.9%** | 0.04 |
| 14 | 4.86e−5 | 1.46e−5 | 8.13e−5 | 3.54e−5 | -67.2% | 0.00 |
| 10-14 | | | | | 20.5% | |
| 15 | 6.41e+1 | 2.84e+1 | **3.72e+1** | 1.30e+1 | **42.0%** | 0.00 |
| 16 | 1.81e+1 | 4.82e+0 | **3.95e+0** | 2.46e+0 | **78.2%** | 0.00 |
| 17 | 1.19e+0 | 9.25e−1 | **2.71e−1** | 3.56e−1 | **77.2%** | 0.00 |
| 18 | 3.07e+0 | 1.59e+0 | **1.21e+0** | 9.27e−1 | **60.5%** | 0.00 |
| 19 | 5.06e+0 | 6.58e-1 | **9.06e-1** | 7.51e−1 | **82.1%** | 0.00 |
| 15-19 | | | | | 63.4% | |
| 20 | 1.35e+00 | 2.94e-1 | **1.08e+0** | 2.50e−1 | **20.0%** | 0.00 |
| 21 | 4.10e+00 | 4.56e+0 | 3.43e+0 | 3.56e+0 | 16.2% | 0.28 |
| 22 | 1.05e+01 | 1.26e+1 | 9.28e+0 | 1.37e+1 | 11.2% | 0.38 |
| 23 | 2.90e+00 | 5.12e−1 | **5.60e−1** | 3.13e−1 | **80.7%** | 0.00 |
| 24 | 1.37e+02 | 1.77e+1 | **4.65e+1** | 1.01e+1 | **66.1%** | 0.00 |
| 21-24 | | | | | 38.9% | |

When added to standard DE, the new adaptive threshold function leads to consistent and mostly significant improvements (%-diff > 10% and p < 0.05 for the ***t***-test) on BBOB sets 4 and 5.

on multi-modal functions can be improved through the addition of a simple and effective threshold function for controlling convergence.

## VII. Discussion

Differential evolution is characterized by fast convergence. Slowing the convergence rate can therefore be beneficial on multi-modal functions. Several techniques to control the convergence of DE and improve its performance already exist, such as crowding and niching. However, these techniques inherently increase the cost and complexity of the algorithm. Existing crowding techniques, for example, examine solutions after they have been evaluated and may need to examine a large number of pairs of solutions for similarity. An important advantage of *thresheld convergence* is its simplicity. A single distance measurement before evaluating the new solution guarantees a minimum (threshold) distance between *base* and *new*.

Hypothetically, the size of the threshold should be large enough for new solutions to explore new attraction basins. The parameter $\alpha$ should ideally be related to the size of the basins and their location in the search space. However, in different functions, attraction basins may have different sizes and distributions. It is thus necessary to adjust accordingly the parameters of *thresheld convergence*, i.e. the initial threshold size ($\alpha$) and the decay rate. To achieve this without *a priori* knowledge of the function's landscape is difficult, so the use of an adaptive threshold function is useful.

The original implementation of *thresheld convergence* for differential evolution is based on rejecting solutions which do no pass the threshold restriction [7]. This approach makes the process of adapting the parameters more difficult. If the parameters are not correctly adjusted, many solutions can be rejected. A steady rejection of solutions will waste possible function evaluations and lead to a drop in performance. With the new design of *thresheld convergence*, the rejection of solutions is avoided and more robust results are achieved.

The new adaptive *thresheld convergence* mechanism provides an effective strategy to improve the performance of a standard implementation of DE. Improvements are larger on multi-modal functions with adequate global structure (BBOB functions 15-19), than on functions with weak global structure (BBOB 20-24). Nevertheless, broad benefits across the full range of multi-modal functions (i.e. BBOB sets 4 and 5) are achieved. With the advent of hyper-heuristics which can select an appropriate heuristic for a given problem [20], a specialist technique for multi-modal search spaces is a useful addition to differential evolution. The simplicity of *thresheld convergence* makes it a promising component for hyper-heuristics since it can be easily activated or deactivated during the search.

The worst comparative results occur for functions 21 and 22 (Gallagher's Gaussian functions [21]). As specified in [18] the key property of these functions is the existence of optima "with position and height being unrelated and randomly chosen". This random feature in these search spaces presents a difficult challenge for adapting the threshold size. Of interest for future research is the design of an adaptive threshold function that can detect members of the population lying in different attraction basins, and perhaps maintain correspondingly different threshold values.

Future research will also study the effects of each parameter more closely. Although the preliminary work presented here has been quite successful, there are still large variations in performance on some functions for different parameter settings. This variation suggests that more improvements can be achieved through the development of better (adaptive) threshold functions. However, one aspect of the current adaptive threshold function that may be difficult to improve is its simplicity. The development of adaptive threshold functions to replace scheduled threshold functions is a definite improvement in terms of simplicity. The large potential benefits, computational efficiency, and general ease of adding *thresheld convergence* make improved threshold functions a promising area for further research.

## VIII. Summary

The *thresheld convergence* technique provides a simple yet effective mechanism to control early convergence in population-based heuristics. Its application to DE leads to large improvements on multi-modal functions. A new design for *thresheld convergence*, as well as an adaptive function, is proposed to replace the original scheduled function(s). The new implementation of *thresheld convergence* for DE improves both simplicity and performance. These results make *thresheld convergence* a promising technique for future research.

## References


[1] R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces," J. Global Optimization, vol. 11, pp. 341–359, 1997.

[2] K. De Jong, "An analysis of the behavior of a class of genetic adaptive systems," PhD thesis, Dept. of Computer and Communication Sciences, University of Michigan, 1975.

[3] S. W. Mahfoud, "Niching methods for genetic algorithms," PhD thesis, University of Illinois, 1995.

[4] R. Brits, A. P. Engelbrecht, and F. Van den Bergh, "A niching particle swarm optimizer," SEAL, 2002, pp. 692–696.

[5] R. Thomsen, "Multi-modal optimization using crowding-based differential evolution," IEEE CEC, 2004, pp. 1382–1389.

[6] S. Chen and J. Montgomery, "Particle swarm optimization with thresheld convergence," IEEE CEC, 2013.

[7] J. Montgomery and S. Chen, "A simple strategy to maintain diversity and reduce crowding in differential evolution," IEEE CEC, 2012, pp. 2692–2699.

[8] S. Chen, C. Xudiera, and J. Montgomery, "Simulated annealing with thresheld convergence," IEEE CEC, 2012, pp. 1946–1952.

[9] A. Bolufé-Röhler and S. Chen, "Minimum Population Search – Lessons from building a heuristic technique with two population members," IEEE CEC, 2013.

[10] E. Mezura-Montes, J. Velázquez-Reyes, and C. A. Coello Coello, "A comparative study of differential evolution variants for global optimization," GECCO 2006, pp. 485–492.

[11] J. Montgomery, "Crossover and the different faces of differential evolution searches," IEEE CEC, 2010, pp. 1804–1811.

[12] D. Zaharie, "Critical values for the control parameters of differential evolution algorithms," ICSC, 2002, pp. 62–67.

[13] X. Li, "Efficient differential evolution using speciation for multimodal function optimization," GECCO, 2005, pp. 873–880.

[14] D. Zaharie, "A multipopulation differential evolution algorithm for multimodal optimization," ICSC, 2004, pp. 17–22.

[15] J. Montgomery, “Differential evolution: Difference vectors and movement in solution space,” IEEE CEC, 2009, pp. 2833–2840.
[16] J. Rönkkönen and J. Lampinen, “On determining multiple global optima by differential evolution,” in Evolutionary and Deterministic Methods for Design, Optimization and Control, Eurogen, 2007, pp. 146–151.
[17] A. Bolufé Röhler, S. Chen, “Multi-swarm hybrid for multi-modal optimization,” IEEE CEC, 2012, pp. 1759–1766.
[18] N. Hansen, S. Finck, R. Ros, and A. Auger, “Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions,” INRIA, Technical Report RR-6829, 2009.
[19] D. H. Wolpert and W. G. Macready, “No free lunch theorems for optimization,” IEEE TEC, vol. 1(1), pp. 67–82, 1997.
[20] E. K. Burke, E. Hart, G. Kendall, J. Newall, P. Ross, and S. Schulenburg, “Hyper-heuristics: An emerging direction in modern search technology”, Handbook of Metaheuristics, Kluwer, 2003, pp. 457–474.
[21] M. Gallagher and B. Yuan, “A General-Purpose Tunable Landscape Generator”, IEEE TEC, vol. 10(5), pp. 590–603, 2006.