Evolution Strategies with Thresheld Convergence

Alejandro Piad-Morffis*, Suilan Estévez-Velarde*, Antonio Bolufé-Röhler*, James Montgomery† and Stephen Chen‡
* Faculty of Math & Computer Science
University of Havana
Havana, Cuba
Email: {apiad,sestevez,bolufe}@matcom.uh.cu
† School of Engineering and ICT
University of Tasmania
Hobart, Australia
Email: james.montgomery@utas.edu.au
‡ School of Information Technology
York University
Toronto, Canada
Email: sychen@yorku.ca

***Abstract*—When optimizing multi-modal spaces, effective search techniques must carefully balance two conflicting tasks: exploration and exploitation. The first refers to the process of identifying promising areas in the search space. The second refers to the process of actually finding the local optima in these areas. This balance becomes increasingly important in stochastic search, where the only knowledge about a function's landscape relies on the relative comparison of random samples. Thresheld convergence is a technique designed to effectively separate the processes of exploration and exploitation. This paper addresses the design of thresheld convergence in the context of evolution strategies. We analyze the behavior of the standard (μ, λ)-ES on multi-modal landscapes and argue that part of it's shortcomings are due to an ineffective balance between exploration and exploitation. Afterwards we present a design for thresheld convergence tailored to ES, as a simple yet effective mechanism to increase the performance of (μ, λ)-ES on multi-modal functions.**

I. Introduction

Effective optimization in multi-modal fitness landscapes requires a fine-tunned balance between two conflicting processes, namely exploration (diversification) and exploitation (intensification) [1]. The explorative process attempts to detect regions of the fitness landscape with high quality solutions, whereas the exploitative process attempts to find the best local optima within these high quality regions, generally using some form of gradient descent. The concurrent execution of both processes may bias the search towards sub-optimal regions leading insufficient time being allocated for exploration. However, many heuristic search algorithms don't explicitly separate both processes.

Thresheld convergence (TC) is a technique that attempts to separate the processes of exploration and exploitation through the use of a threshold function. By enforcing a minimum search step, its goal is to avoid over-sampling specific regions of the search space at early stages of the search process. Thresheld convergence has proven useful to increase the search performance when applied to rapidly converging search techniques such as simulated annealing (SA) [2], particle swarm optimization (PSO) [3] and differential evolution (DE) [4].

This paper addresses the design of thresheld convergence in the context of evolutions strategies. Evolution strategies (ES) [5] are a family of heuristic search techniques based on a sampling process guided by the automatic adaptation of a set of endogenous parameters that define the sampling schema. One of the simplest adaptation techniques for (μ, λ)-ES is the so-called log-normal learning strategy. It adapts a single endogenous parameter, the variance (σ) of the Gaussian distribution used for mutating the new solutions. We analyze behavior of $(\mu + \lambda)$-ES with log-normal learning on multi-modal spaces and argue that part of its shortcomings lies in its inability to perform and unbiased sampling.

Modern variants of ES incorporate a larger number of endogenous parameters and more advanced self-adaptation strategies. We take an alternative route, attaching (μ, λ)-ES with a mechanism to restrict the bias introduced by its elitist selection criteria. However, the main motivation for this research is not merely to improve the performance of ES in multi-modal landscapes, but to provide evidence to support our claim that the log-normal learning rule biases the search.

We hypothesize that the minimum search step enforced by TC's threshold function can provide a more uniform sampling and improve the exploration process. Our research problem is how to integrate thresheld convergence with the ES self-adaptation rule. By analyzing the behavior of ES with thresheld convergence, we aim to understand the impact of each of the design parameters of TC, and to provide better insight for applying thresheld convergence to other, more complex, search techniques.

The paper is organized as follows: In Section II we introduce both the state of the art of thresheld convergence and the standard design of (μ, λ)-ES. In Section III we perform a detailed analysis of (μ, λ)-ES on multi-modal landscapes and argue the main shortcomings of ES in this scenario. In Section IV we present a variant of thresheld convergence applied to ES, and analyze it's design decisions. In Section V we perform an experimental comparison of our proposal with

978-1-4799-7492-4/15/$31.00

standard (μ, λ)-ES. Section VI presents the results and conclusions, and provides new insights into the design of thresheld convergence. Finally, Section VII summarizes the main ideas and results of the paper.

II. Background

A. Evolution Strategies

The standard formulation for evolution strategies, introduced in [5], can be described with the notation $(\mu/\rho\ \{+,\}\ \lambda)$-ES. In this notation, μ stands for the number of parents and λ for the number of offspring. Each sample μ_i consists of three sets of values: a vector $\boldsymbol{x_i} \in \mathrm{R}^n$ in the function domain, that represents a possible solution, a vector $\boldsymbol{y_i} \in \mathrm{R}^m$ in the parameter space that represents the set of m endogenous parameters that guide the search, and the the fitness value $f_i = F(\boldsymbol{x_i})$ for the corresponding solution. Every offspring is generated by combining a random subset of ρ parents out of the possible μ, and adding a mutation factor controlled by the endogenous parameters $\boldsymbol{y}$. The signs $+$ and , are used to denote the strategy for combining one generation with the next, either by merging the μ and λ samples $(+)$, or by completely replacing the μ parents with the λ offspring $(,)$. In either case, after each generation, only the best μ samples are kept. The special case of $\rho = 1$ is simply denoted $(\mu\{+,\}\lambda)$-ES.

The mutation strategy applied depends on the parameters vector $\boldsymbol{y}_i$. In the simplest case, the parameter vector contains a single value, $\boldsymbol{y}_i = (\sigma_i)$. In this case, the mutation strategy consists of adding a univariate normal variable $r_i \sim N(0, \sigma_i)$, and the endogenous parameters control the strength of the mutation (variance of the random distribution).

The parameter vector of the new offspring $\boldsymbol{y}_i$ is also mutated, using a different rule. Each of the μ_i parents carries it's own σ_i. After each of the λ_i is generated, the parameter σ_i' inherited from the parent is mutated applying the log-normal rule

$$\sigma_i' \leftarrow \sigma_i \cdot e^{\tau \cdot N(0,1)}$$

where the learning parameter τ controls the strength of the mutation, and $N(0, 1)$ is a standard univariate normal variable. The generally recommended learning rate is $\tau = 1/\sqrt{N}$ where N is the dimension of the search space. For highly multi-modal functions, values such as $\tau = 1/\sqrt{2N}$, or even smaller, are recommended. A general outline is presented in Algorithm 1.

ES can be seen as performing optimization in two different levels: the solution space, where the algorithm tries to find the best input vectors, and the parameter space, where the algorithm tries to find the optimal set of parameters that guide the search. Hence, as the search progresses, the algorithm attempts to find its own optimal state for searching in that specific function.

The difference between the $+$ and , variants is that, since ES has an elitist selection criteria, in $(\mu + \lambda)$-ES the fittest individuals can survive for a large number of generations. In contrast, (μ, λ)-ES exhibits a more explorative behavior, and performs better in multi-modal spaces [6]. In either case, the elitist selection strategy can lead to premature convergence. Dynamic niching [7] has been suggested to address this problem. At the beginning of every generation, after the fitness is updated, the solutions are classified in niches, represented by a subset of the population members, such that the distance between every pair of solutions is greater than some fixed threshold. Recombination, mutation and selection are then performed independently for every niche sub-population, and then the resulting sub-populations are merged back.

Algorithm 1 (μ, λ)-ES

$P \leftarrow \{\ \}$
for λ times **do**
  $\boldsymbol{x}_k \leftarrow$ random starting point
  $y_k \leftarrow f(\boldsymbol{x}_k)$
  $\sigma_k \leftarrow$ random starting variance
  $P \leftarrow P \cup \{(\boldsymbol{x}_k, y_k, \sigma_k)\}$
end for
for all generations **do**
  $Q \leftarrow$ best μ solutions in P
  $P \leftarrow \{\ \}$
  for all parent Q_k **do**
    for λ times **do**
      $(\boldsymbol{x}_k, y_k, \sigma_k) \leftarrow Q_i$
      $\boldsymbol{x}_k' \leftarrow \boldsymbol{x}_k + \langle N(1, \sigma_i)\rangle^n$
      $y_k' \leftarrow f(\boldsymbol{x}_k')$
      $\sigma_k' \leftarrow \sigma_k \cdot e^{\tau \cdot N(0,1)}$
      $P \leftarrow P \cup \{(\boldsymbol{x}_k', y_k', \sigma_k')\}$
    end for
  end for
end for
return $\boldsymbol{x}_k \in P$ with minimum y_k

B. Thresheld Convergence

Threshed convergence is a high level technique designed to guide a stochastic optimization algorithm towards a balanced sampling regime. TC is built upon the hypothesis that a concurrent execution of exploration and exploitation may bias the exploration process. Let's take for instance a multi-modal function where the attraction basins have proportional sizes and shapes. If only one random solution could be sampled from each attraction basin, then there is a reasonable expectation that the best solutions correspond to the best basins. The process of detecting the best attraction basins (exploration) will be simplified to selecting the best samples. However, if some attraction basins are over-sampled (exploitation) then the chances that worse attraction basins are represented by better solutions will increase. This is what we call a *biased exploration* (Fig. 1).

In general, obtaining a precise measurement of the fitness of an attraction basin is not possible until local search has been used to identify the actual local optimum. Thus, in many heuristics, exploration and exploitation are merged during the search process, i.e. large (explorative) and small (exploitative) search steps are indistinguishably made by the algorithm. Since an algorithm's ability to detect the best attraction basins ultimately depends on the comparison of random samples, the process of detecting the best regions may thus be compromised, and the selection pressure may lead the search to focus on sub-optimal regions (attraction basins).

Thresheld convergence aims to solve this problem by controlling the distance (search step) between a parent and

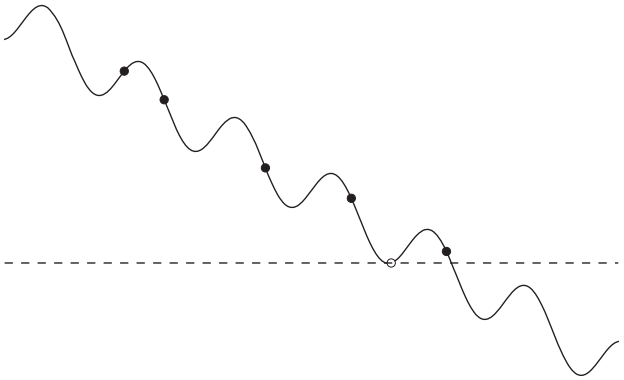

Fig. 1. Effects of over-sampling an attraction basin. When compared to random solutions (black dots), an exploited solution (white dot) from a sub-optimal attraction basin exhibits better fitness than a random solution from a better attraction basin.

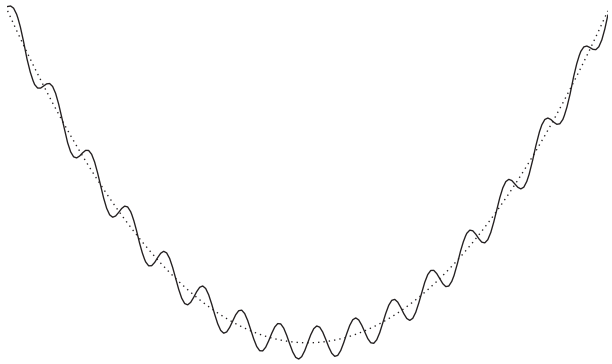

Fig. 2. Example of a function with well behaved global structure: a high-frequency sinusoidal superimposed on a quadratic slope. The dotted line shows the underlying quadratic gradient.

an offspring solution. If the size of the search step is large enough, then new solutions are likely to be sampled from a different attraction basin. This is achieved by fixing a minimum search step (threshold) which decays as the search progresses —convergence is thus "held" back until the last stages of the search process. Exploration is guaranteed to be performed during the early iterations, and a gradual transition to exploitation (local search) is ensured once the best regions have been found. 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 (1). In (1), d is the diagonal of the search space, n is the total number of generations, and i is the current generation. The parameter α determines the initial threshold and γ controls the decay rate.

$$threshold = \alpha \cdot d \cdot \left(\frac{n-i}{n}\right)^{\gamma} \tag{1}$$

Previous work has applied TC to particle swarm optimization and differential evolution. In PSO, the minimum step is enforced between the local and personal attractors (best solutions found). A new solutions replaces the personal attractor only if its distance to the local attractor is larger than the threshold [3]. In DE, a new solution is evaluated if its distance to the base (parent solution) is larger than the threshold. Otherwise, the solution is "pushed" a threshold distance away from its parent [4].

Thresheld convergence achieves its best results on multi-modal functions with a well behaved global structure. A function is said to have "good" global structure if there exists a search step size such that the function's landscape, when sampled with such a scale, behaves as an unimodal function. An illustrative case is a sinusoidal function superimposed on a quadratic function (Fig. 2). The quadratic component provides a single "global" attraction basin, in which smaller "local" attraction basins are superimposed. These types of functions are said to have two search scales. It remains an active area of research to design a variant of thresheld convergence that can effectively deal with more than two search scales or very chaotic and dissimilar attraction basins [8].

III. Analysis of Evolution Strategy on Multi-Modal Landscapes

In order to gain a better understanding of the behavior of (μ, λ) on multi-modal landscapes, we analyze its behavior using the methodology proposed in [9]. This analysis consists on observing the behavior of ES in the well-known Rastrigin's function, as defined in [10] (hereafter Rastrigin):

$$f(\boldsymbol{x}) = An + \sum_{i=1}^{n} \left[x_i^2 - A\cos\left(2\pi x_i\right)\right]$$

where n is the dimension of the space, $A = 10$ and $\boldsymbol{x} \in [-5.12, 5.12]^n$. This function has a global optima located at $\boldsymbol{x} = \mathbf{0}$ with $f(\mathbf{0}) = 0$, and 11^n local optima located at the integer values of a regular grid of size 1. Hence, the minimum and maximum distance between two adjacent local optima are 1 and $\sqrt{n}$ respectively.

By rounding all components of a given sample $\boldsymbol{x}$ to the nearest integer, we can effectively identify the local optimum $\boldsymbol{x}^*$ in whose basin $\boldsymbol{x}$ is located. That is, if pure gradient descent is performed starting at $\boldsymbol{x}$ the solution will converge to $\boldsymbol{x}^*$. This information will prove useful to analyze the behavior of (μ, λ)-ES on multi-modal spaces, since it allows us to quantify the balance of exploration and exploitation as the algorithm progresses. Rastrigin is a classic example of a non-linear, multi-modal function with a well behaved global structure and two clearly defined search scales.

Fig. 3 shows the average performance of $(10, 100)$-ES in terms of generations, up to a maximum of $3,000$. The learning parameter τ was chosen as $1/(4 \cdot \sqrt{N})$ after an initial experimentation with various possibilities. The solid line shows the average fitness of the current sample $\boldsymbol{x_k}$ in generation k, while the dashed line shows the fitness of the corresponding local optimum $\boldsymbol{x_k^*}$. Thus, changes in the dashed

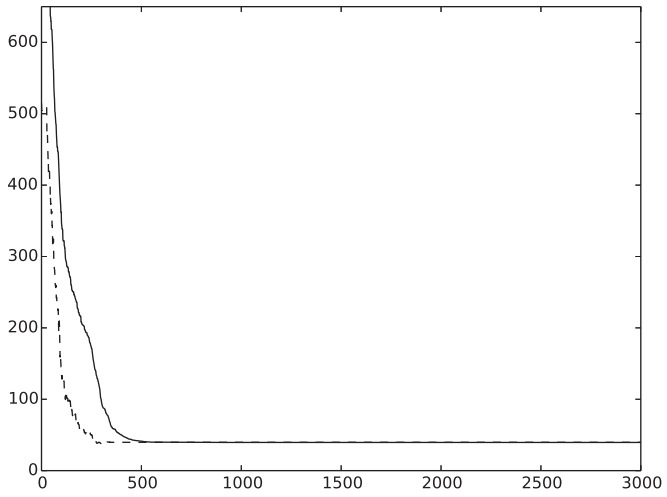

Fig. 3. Results of $(10, 100)$-ES on Rastrigin with $n = 30$. The plot shows the average performance over 51 trials. Mean performance is $f(\boldsymbol{x}_{best}) = 39.5$.

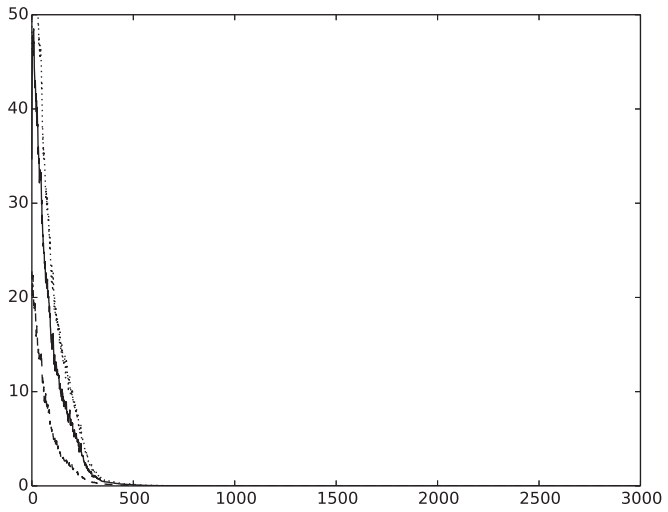

Fig. 4. Maximum (dotted), average (solid) and minimum (dashed) step size of $(10, 100)$-ES on Rastrigin with $n = 30$, over 51 trials. The minimum and maximum distance between two neighboring attraction basins is respectively 1 and $\sqrt{n} \simeq 5.5$.

line represent progress of the global search (exploration), while the convergence of the solid line towards the dashed line denotes evidence of local search (exploitation). In this landscape, (μ, λ)-ES shows a strong tendency towards local search, even with a very small value for τ. The global search phase lasts only for the first 10% of the generations. After this point, the algorithm is unable to find a better attraction basin, and only local search is performed. Local search also ends early on, at 25% of the generations, after which the algorithm no longer produces improving results.

To better understand this phenomenon of fast convergence, its useful to look at the actual size of the search steps. Fig. 4 shows the minimum, average and maximum distances between the parent solution and the offspring for each iteration. As soon as the first 10% of the generations, the average step size reaches the maximum distance between neighboring attraction basins. From this point on, new solutions begin to fall with higher probability inside the same attraction basins than in new attraction basins. At around the first 20% of the iterations, the maximum step size is actually lower than the minimum distance between neighboring attraction basins. After this point, new solutions can only be generated inside a single attraction basin. Hence, after this point, only local search is performed. The best attraction basin was determined with only 20% of the alloted function evaluations.

Fig. 5 shows the average value of the endogenous parameter σ^* for the best solution $\boldsymbol{x}^*$ in every generation. Theoretically, the log-normal rule has the potential to increase the endogenous parameter if needed. However, in the case of the Rastrigin function, this rarely happens. The selection criteria for (μ, λ) is highly exploitative, even when all parents are replaced each generation. To achieve better exploration, the learning factor τ needs to be adjusted to a very low value, beyond the minimum recommended for multi-modal spaces. However, if the learning factor is very low, then local exploration will be delayed, and the algorithm may fail to converge. The right value for τ appears to be highly correlated with the landscape being optimized.

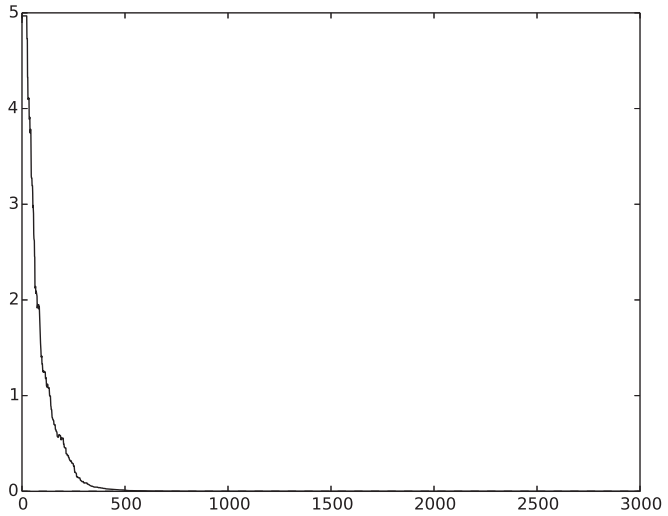

Fig. 5. Average value of the endogenous parameter σ for the best solution in every generation. Contrary to the expected behavior, this value rarely increases, instead dropping very fast towards zero.

IV. Evolution Strategy with Thresheld Convergence

In multi-modal search spaces, (μ, λ)-ES behaves extremely exploitatively. Based on this observation, we hypothesize that the addition of a minimum search step might mitigate the effects of early convergence. The original formulation of TC consists of a scheduled threshold function which decays exponentially with each generation. To apply such a formulation to ES, the algorithm needs to be modified to reject solutions that are closer to the corresponding parent than the current threshold. This rejection strategy has a negative impact in the performance of the algorithm. If the rejected solutions are counted as evaluated, the algorithm wastes solutions that could lead to better local optima. On the other hand, if the rejected solutions are not counted as evaluated, then there is no upper

bound for how many generations the algorithm will take. In either case, performance, in the sense of fitness or in the sense of speed, is harmed.

In [4], a simple strategy is proposed to mitigate the effects of rejection. Solutions that don't pass the threshold test are "pushed" towards the threshold, in the direction determined by the difference vector between each offspring and its parent. However, pushing rejected solutions to the threshold may have undesired effects in the behavior of the algorithm, because it changes the characteristics of the sampling scheme. If many solutions are rejected, they will all lie on the surface of a hypersphere, at the same distance from the parent. Thus, the algorithm looses its ability to search in one of the dimensions, and the sampling becomes biased in the rest of the dimensions. Instead of pushing, we propose reflecting a rejected solution with respect to the threshold, such that no function evaluation is wasted, and search remains unbiased. The new solution ($\boldsymbol{x}''$) is obtained by adding a vector in the line determined by the solution ($\boldsymbol{x}'$) and its parent ($\boldsymbol{x}$), towards the threshold, twice the size of the gap between the threshold and the distance of the solution to its parent (2).

$$\boldsymbol{x}'' = \boldsymbol{x}' + 2 \cdot (threshold - ||\boldsymbol{x} - \boldsymbol{x}'||) \cdot (\boldsymbol{x}' - \boldsymbol{x}) \quad (2)$$

Algorithm 2 shows a general outline of the resulting technique, (μ, λ)-ES+TC. The differences with standard (μ, λ)-ES can be observed in line 11 where the value of threshold is calculated, according to (1), and in line 17 where a new solution that fails the threshold test is reflected according to (2).

Algorithm 2 (μ, λ)-ES+TC

1: $P \leftarrow \{\ \}$
2: **for** λ times **do**
3: $\quad \boldsymbol{x}_k \leftarrow$ random starting point
4: $\quad y_k \leftarrow f(\boldsymbol{x}_k)$
5: $\quad \sigma_k \leftarrow$ random starting variance
6: $\quad P \leftarrow P \cup \{(\boldsymbol{x}_k, y_k, \sigma_k)\}$
7: **end for**
8: **for all** generations **do**
9: $\quad Q \leftarrow$ best μ solutions in P
10: $\quad P \leftarrow \{\ \}$
11: $\quad T \leftarrow \alpha \cdot d \cdot \left(\frac{n-i}{n}\right)^{\gamma}$
12: $\quad$ **for all** parents Q_k **do**
13: $\quad\quad$ **for** λ times **do**
14: $\quad\quad\quad (\boldsymbol{x}_k, y_k, \sigma_k) \leftarrow Q_i$
15: $\quad\quad\quad \boldsymbol{x}'_k \leftarrow \boldsymbol{x}_k + \langle N(1, \sigma_i) \rangle^n$
16: $\quad\quad\quad$ **if** $||\boldsymbol{x}_k - \boldsymbol{x}'_k|| < T$ **then**
17: $\quad\quad\quad\quad \boldsymbol{x}'_k \leftarrow \boldsymbol{x}'_k + 2 \cdot (T - ||\boldsymbol{x}_k - \boldsymbol{x}'_k||) \cdot (\boldsymbol{x}'_k - \boldsymbol{x}_k)$
18: $\quad\quad\quad$ **end if**
19: $\quad\quad\quad y'_k \leftarrow f(\boldsymbol{x}'_k)$
20: $\quad\quad\quad \sigma'_k \leftarrow \sigma_k \cdot e^{\tau \cdot N(0,1)}$
21: $\quad\quad\quad P \leftarrow P \cup \{(\boldsymbol{x}'_k, y'_k, \sigma'_k)\}$
22: $\quad\quad$ **end for**
23: $\quad$ **end for**
24: **end for**
25: **return** $\boldsymbol{x}_k \in P$ with minimum y_k

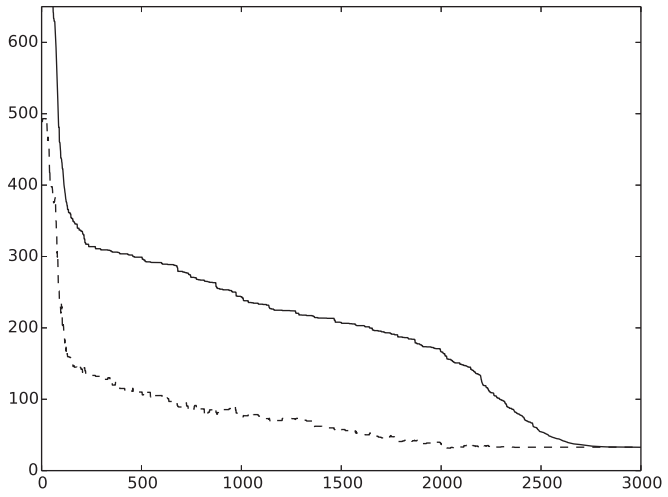

Fig. 6. Results of $(10, 100)$-ES+TC on Rastrigin with $n = 30$, with a scheduled threshold function. The plot shows the average performance over 51 trials. Mean performance is $f(\boldsymbol{x}_{best}) = 32.8$.

Fig. 6 shows the plot of the fitness of $(10, 100)$-ES+TC on Rastrigin's function. Initial results show an improvement of approximately 17% in performance. More importantly, the global search phase has increased to around 50% of the generations, and local search occurs in the final 33% of generations. Once the value of the threshold is low enough, search steps become smaller and the algorithm converges naturally. This is aided by the fact that (μ, λ)-ES is highly exploitative on its own.

Fig. 7 shows the distances from newly generated solutions to the corresponding parents. Approximately during the first 33% of the generations, the minimum search steps remain bigger than the size of the attraction basins. A large transitioning phase from global to local search can be appreciated between generation 1000 and generation 2000. In the final 33% of the generations, the search steps decrease below the size of attraction basins and local search is performed. The specific intervals at which global and local search are performed are only valid for the Rastrigin function. However, we hypothesize that in any function with a well behaved global structure, there will exist an interval of generations where the threshold will be suitable, if initially large enough.

Fig. 8 shows the values of the threshold and the endogenous parameter σ for the best solution in each generation. Due to the exploitative nature of (μ, λ)-ES, the initially large σ decays rapidly when solutions are generated farther away than the threshold. Before the first 5% of the generations are completed, the algorithm finds an optimal value for σ that generates solutions just above the threshold. Afterwards, σ decreases linearly to maintain this optimal sampling regime.

V. Experimental Results

To compare the improvement achieved when adding thresheld convergence to (μ, λ)-ES, both techniques are run on the standard CEC 2013 benchmark [11]. This benchmarks consists of a set of 28 unimodal and multi-modal functions with various characteristics. The functions are divided in three sets: unimodal functions (1 to 5), basic multi-modal functions (6 to

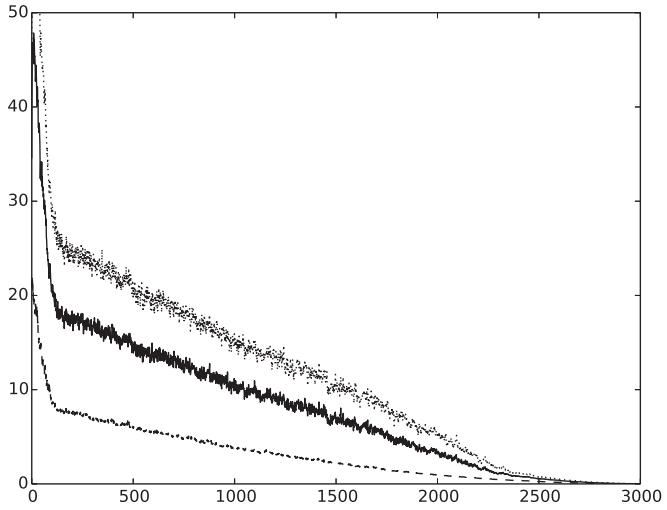

Fig. 7. Maximum, average and minimum distance between each solution and its corresponding parent.

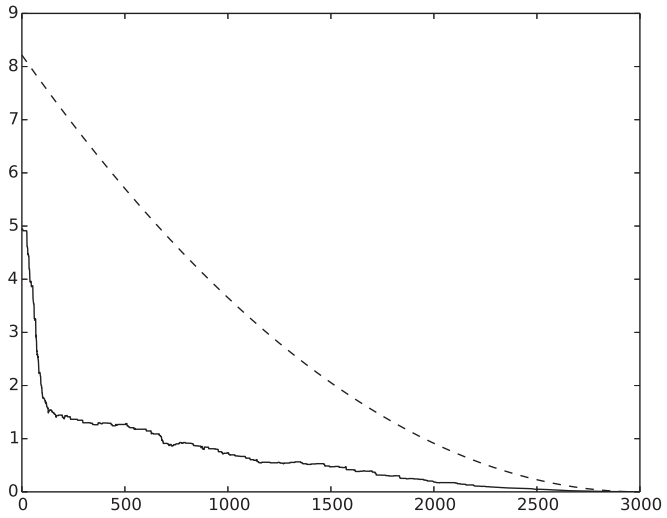

Fig. 8. Average value of the threshold (dashed line) and the endogenous parameter σ (solid line) for the best solution in each generation.

20) and composite multi-modal functions (21 to 28). The second group presents the most interesting functions in terms of landscape characteristics, since thresheld convergence has been specifically designed for optimizing these kinds of functions. All functions have known global optima, and many present characteristics such as rotations and asymmetric perturbation to increase their difficulty. The experimental setup consists on 51 independent runs on each function on 30 dimensions with a maximum of $300,000$ function evaluations.

Performance is measured by comparing the mean error. The relative difference of the performance of the algorithms is also reported, calculated as $(p_2 - p_1)/\max(p_1, p_2)$ where p_1 is the performance of the original ES, and p_2 is the performance of ES with thresheld convergence. Hence, positive results means an improvement when applying TC, and *vice versa*. Table I shows a comparison of $(10, 100)$-ES+TC with respect to standard $(10, 100)$-ES. The implementation of thresheld

TABLE I. COMPARISON OF $(10, 100)$-ES WITH AND WITHOUT TC.

	$(10, 100)$-ES+TC		$(10, 100)$-ES			
No.	Mean	Stdev	Mean	Stdev.	%-diff	t-test
1	2.562e-04	2.815e-04	2.273e-13	0.000e+00	-99.9 %	0.02
2	1.754e+05	1.024e+05	1.829e+06	9.048e+05	**90.4 %**	0.00
3	6.733e+05	1.818e+06	1.865e+09	1.133e+09	**99.9 %**	0.00
4	1.903e+04	8.323e+03	1.438e+05	3.333e+04	**86.7 %**	0.00
5	2.880e-03	5.405e-04	1.730e-03	2.722e-04	-39.9 %	0.00
1-5					27.4 %	
6	1.083e+01	1.453e+00	3.066e+01	2.445e+01	**64.6 %**	0.04
7	1.087e+00	1.432e+00	1.076e+04	2.442e+04	99.9 %	0.22
8	2.082e+01	1.352e-01	2.087e+01	8.853e-02	0.2 %	0.37
9	8.525e+00	3.113e+00	3.740e+01	3.101e+00	**77.2 %**	0.00
10	7.461e-03	5.305e-03	5.681e-01	6.569e-01	**98.6 %**	0.03
11	4.158e+01	9.405e+00	9.720e+02	2.341e+02	**95.7 %**	0.00
12	2.875e+01	7.750e+00	1.179e+03	1.282e+02	**97.5 %**	0.00
13	7.299e+01	3.551e+01	1.580e+03	2.791e+02	**95.3 %**	0.00
14	1.222e+03	2.293e+02	4.225e+03	7.076e+02	**71.0 %**	0.00
15	1.569e+03	2.896e+02	4.451e+03	5.529e+02	**64.7 %**	0.00
16	9.429e-01	1.133e+00	1.483e-01	7.320e-02	-84.2 %	0.06
17	5.748e+01	4.620e+00	1.677e+03	9.507e+02	**96.5 %**	0.00
18	5.787e+01	6.492e+00	9.831e+02	9.486e+02	**94.1 %**	0.02
19	2.653e+00	8.350e-01	1.692e+01	2.208e+01	84.3 %	0.08
20	1.445e+01	1.112e+00	1.476e+01	3.981e-01	2.0 %	0.45
6-20					63.8 %	
21	3.089e+02	7.756e+01	3.030e+02	1.008e+02	-1.9 %	0.89
22	1.649e+03	2.417e+02	5.866e+03	9.603e+02	**71.8 %**	0.00
23	1.903e+03	4.079e+02	6.038e+03	7.524e+02	**68.4 %**	0.00
24	2.087e+02	3.642e+01	5.613e+02	2.869e+02	**62.8 %**	0.00
25	2.326e+02	1.138e+01	4.359e+02	6.539e+01	**46.6 %**	0.00
26	2.751e+02	6.166e+01	4.169e+02	2.578e+02	34.0 %	0.14
27	5.603e+02	2.701e+01	1.479e+03	1.978e+02	**62.1 %**	0.00
28	4.939e+02	3.867e+02	7.349e+03	2.146e+03	**93.2 %**	0.00
21-28					54.6 %	

convergence used is based on the results of section IV, i.e., a minimum search step controlled by an scheduled threshold function. Parameters such as initial threshold, initial σ and initial τ are also equal to those used in previous sections. A standard t-test is performed to measure statistical significance. Results with statistically significant improvements (t-test < 0.05) of our version over the baseline ES are highlighted.

When compared with standard $(10, 100)$-ES, thresheld convergence provides a clear improvement in 19 out of the 28 benchmark functions tested. On the subset of unimodal functions, results are inconsistent. However, thresheld convergence is not designed to perform optimization on these types of functions. On the subset of basic multi-modal functions, thresheld convergence provides a consistent improvement over the standard ES. The mean relative difference for these functions where a statistically significant difference exists is 85.5%. On the subset of composite multi-modal functions, results are also consistent, with an average of 67.5% performance improvement.

VI. DISCUSSION

It is hypothesized that thresheld convergence is not a suitable mechanism for unimodal optimization, because it explicitly attempts to delay convergence. In unimodal spaces, once a gradient is identified, the more efficient course of

action consists on a greedy search down the gradient. The only concern lies in adjusting the search step to avoid overshooting the global optimum. More exploitative search techniques, such as standard (μ, λ)-ES, the $+$ variant, and even $(1 + \lambda)$-ES with the 1/5-th success probability rule [12] can provide better results.

In multi-modal functions, thresheld convergence provides a clear advantage. The largest improvements with respect to $(10, 100)$-ES are obtained in functions with a consistent global structure: Schaffer's F7 function (function 7), Griewank's function (10), Rastrigin functions (11 to 13) and the Lunacek bi-Rastrigin variants (functions 17 and 18). Function 7 has a huge number of local optima, hence its very easy for ES to get side tracked without a proper global search. However, on a larger scale, it presents a clearly defined spherical gradient to follow towards the region with the best local optima. Functions 10 to 13 also have a very consistent global structure. Three of these are Rastrigin variants, with rotation and asymmetric perturbations. Griewank's function, on the other hand, consists of a sinusoidal superimposed on an elliptic landscape.

Functions 17 and 18 are composites of two quadratic penalization functions and a sinusoidal, with additional features (asymmetrical perturbations and rotation) for increased difficulty. However, both functions have two clearly defined search scales and mostly similar attraction basins. In both cases, the largest quadratic function leads to a region of suboptimal attraction basins, whereas a smaller quadratic region that is harder to find leads to global optima. All of these functions exhibit characteristics similar to the basic Rastrigin function: two clearly defined search scales, and a global gradient that can be followed to the most promising attraction basins.

In the composite multi-modal set, the best result is achieved for function 28, which is a composite of functions such as Griewank's and Schaffer's F7, where TC performs best. The global optima is located in a large spherical basin, which is easy to explore with a large enough threshold.

The worst results where obtained on function 16 (rotated Katsuura). This function contains a huge number of very deep local optima. It is a continuous function in its entire domain, but differentiable nowhere. At a large scale the function's landscape is nearly flat. We hypothesize that for this kind of functions global search provides almost no advantage versus a very exploitative algorithm that can at least find one local optimum. TC wastes most of the generations trying to explore a gradient that is almost non-existent. When search steps become small, there are not enough function evaluations left to perform an intense local search.

The smallest effects of TC were obtained in functions 8 (rotated Ackley's) and 20 (Schaffer's F6), both with a very inconsistent global structure, attraction basins of multiple sizes and non-circular shapes and no clearly identifiable gradient towards the global optimum. In these functions, the basic hypothesis of thresheld convergence fails, since the exploration phase can not correctly identify the most promising areas, because there is no large scale gradient that guides towards the best attraction basins. Without an adequate global scale, there is too little information in the structure of the function for an algorithm to learn, and search basically degenerates to a pure Monte-Carlo sampling, regardless of the threshold size. Furthermore, long search step sizes delay convergence to local optima. Hence, it is to be expected that thresheld convergence, specifically designed to exploit the larger search scale, provides little to no improvement over a very exploitative algorithm like ES. However, in contrast with function 16, TC doesn't produce negative results in these types of functions.

It is hypothesized that for the most difficult functions a single threshold is not sufficient to completely capture the right scale for global search. Functions with attraction basins of different sizes may require different threshold sizes to be correctly explored. In a population-based ES, the threshold size can be contemplated as another endogenous parameter, and self-adapted in a similar way to the sample variance. This way, effective threshold sizes can be discovered by means of self-adaptation. This strategy will be explored in future research. Also, a full analysis and comparison with state-of-the-art evolution strategies, such as CMA-ES [13], is obliged.

The proposed inclusion of thresheld convergence into the general mechanism of ES enables the control of the convergence rate, without changing the nature of ES, its general sampling and mutation mechanisms. This is an advantage to other, more invasive convergence control techniques (such as dynamic niching [7]), in the sense that more complex self-adaptation strategies for ES and different threshold functions can be easily combined without interfering with each other. The influence of the TC mechanism can be dynamically adjusted during the search process, which allows for a greater control of the convergence rate of ES. This control can be used either to completely deactivate TC at an specific generation, or to gradually reduce the effect of the convergence control during the search process. More advanced adaptive threshold functions, that can effectively identify the correct thresholds in functions with several search scales, can be easily incorporated to the current framework.

VII. Summary

Effective search techniques in multi-modal spaces need to correctly balance between local and global search, however, must heuristic search methods perform these two processes concurrently. Thresheld convergence is a high level mechanism designed to prevent concurrent exploration and exploitation. Applying thresheld convergence to a specific search technique involves analyzing when and why search steps begin to drop below the size of attraction basins. (μ, λ)-ES exhibits a very exploitative behavior on multi-modal spaces. Even with a very low learning factor, the elitist selection criteria leads the algorithm rapidly towards local search. When attached with thresheld convergence, the performance of ES increases in a broad range of multi-modal functions. Furthermore, ES provides an adequate theoretical framework for analyzing thresheld convergence and for designing more advance threshold functions.

References

[1] M. Črepinšek, S.-H. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms: a survey," *ACM Computing Surveys (CSUR)*, vol. 45, no. 3, p. 35, 2013.

[2] S. Chen, C. Xudiera, and J. Montgomery, "Simulated annealing with thresheld convergence," in *Evolutionary Computation (CEC), 2012 IEEE Congress on*, June 2012, pp. 1–7.

[3] S. Chen and J. Montgomery, "Particle swarm optimization with thresheld convergence," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*, June 2013, pp. 510–516.

[4] A. Bolufe-Rohler, S. Estevez-Velarde, A. Piad-Morffis, S. Chen, and J. Montgomery, "Differential evolution with thresheld convergence," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*, June 2013, pp. 40–47.

[5] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies – a comprehensive introduction," *Natural Computing*, vol. 1, no. 1, pp. 3–52, 2002.

[6] H.-G. Beyer, "Toward a theory of evolution strategies: Self-adaptation," *Evolutionary Computation*, vol. 3, no. 3, pp. 311–347, 1995.

[7] O. M. Shir and T. Bäck, "Niching in evolution strategies," in *Proceedings of the 2005 conference on Genetic and evolutionary computation*. ACM, 2005, pp. 915–916.

[8] J. Montgomery, S. Chen, and Y. Gonzalez-Fernandez, "Identifying and exploiting the scale of a search space in differential evolution," in *Evolutionary Computation (CEC), 2014 IEEE Congress on*, July 2014, pp. 1427–1434.

[9] Y. Gonzalez-Fernandez and S. Chen, "Identifying and exploiting the scale of a search space in particle swarm optimization," in *Proceedings of the 2014 Conference on Genetic and Evolutionary Computation*, ser. GECCO '14. New York, NY, USA: ACM, 2014, pp. 17–24.

[10] H. Mühlenbein, M. Schomisch, and J. Born, "The parallel genetic algorithm as function optimizer," *Parallel computing*, vol. 17, no. 6, pp. 619–632, 1991.

[11] J. Liang, B. Qu, P. Suganthan, and A. G. Hernández-Dıaz, "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*, vol. 201212, 2013.

[12] H.-G. Beyer, "Towards a theory of 'evolution strategies': Results for (1,+ λ)-strategies on (nearly) arbitrary fitness functions," in *Parallel Problem Solving from Nature—PPSN III*. Springer, 1994, pp. 57–67.

[13] N. Hansen, "The cma evolution strategy: a comparing review," in *Towards a new evolutionary computation*. Springer, 2006, pp. 75–102.