An Analysis of Minimum Population Search on Large Scale Global Optimization

Antonio Bolufé-Röhler
School of Mathematical and Computational Sciences
University of Prince Edward Island
Charlottetown, Canada
aboluferohler@upei.ca

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

Dania Tamayo-Vera
Thinking Big Inc.
Charlottetown, Canada
dania@thinkingbig.net

***Abstract*—Minimum Population Search is a recently developed metaheuristic specifically designed to optimize high dimensional multi-modal functions. In this paper, Minimum Population Search is evaluated on the test functions provided for the 2013 LSGO competition in the IEEE Congress of Evolutionary Computation (CEC 2013). Furthermore, an analysis using some recent categorizations for exploitation and exploration, and especially the effects of selection, sheds light on how MPS performs better in high dimensions than popular metaheuristics such as Differential Evolution and Particle Swarm Optimization.**

***Index Terms*—minimum population search, large scale global optimization, exploration, multimodality**

I. Introduction

With the rise of big data, Large Scale Global Optimization (LSGO) has become prevalent in many real world applications such as the Internet of Things, bioinformatics, logistics and many others [1]. With increasing dimensionality, the performance of most optimization algorithms rapidly deteriorates, and this effect is known as the curse of dimensionality. A natural way to tackle the effect caused by large dimensions is to exploit separability, i.e. to partition a large problem into a number of sub-components and then solving them independently. Decomposition techniques have been successfully applied to many large scale problems and are undoubtedly one of the key research lines in LSGO, but they may be limited when facing non-separable problems [2]. In this paper, we bring attention to two other aspects that can play a role in LSGO: efficiency and exploration.

In continuous domains, as dimensions increase the volume of the search space grows exponentially. However, for practical reasons, the number of function evaluations (a measure of computational resources) that is budgeted to explore this search space usually increases linearly. Other parameters such as population size also often increase linearly [3]. The magnitude of the divergence between these growths is sometimes hard to grasp, so a useful visualization of this effect is presented in [4] (p.514):

> A simple guideline from differential evolution (DE) is to use a population size (n) that is ten times the dimensionality (d) of the search space [3]. However, going from d = 2 to d = 42 can increase the size of the search space by a factor of about $2^{40} \approx 10^{12}$. Whereas we might imagine 20 birds being able to fully explore a 10 m · 10 m courtyard, we should have a different visualization of 420 birds in an area similar to the size of the Atlantic Ocean.

A direct way to tackle this issue is to increase the efficiency in the use of allotted resources, such as the focus we place on function evaluations (FEs) in this study. In population-based metaheuristics, efficiency is strongly related with the key parameter of population size. While larger populations generally lead to a better coverage of the search space, they also limit the number of iterations an algorithm can perform given a fixed budget of FEs. This is a well known trade-off between the need to explore different regions of the search space, and the need to perform many iterations to converge towards a local/global optimum. The algorithm we analyze in this study derives its name precisely from this issue. Minimum Population Search (MPS), was specifically designed to use a relatively small population to explore the search space efficiently and to achieve a final convergence towards an optimum.

The other aspect that we want to discuss from an LSGO perspective is exploration. When optimizing multimodal functions, exploration is in charge of detecting the best regions of the search space while exploitation is tasked with converging to the corresponding optima. In previous work [5], we have provide a precise definition of exploration based on dividing a multi-modal search space into attraction basins which each have a single local optimum. This definition has allowed us to analyze the effects of concurrent exploration and exploitation and to define new concepts such as failed exploration. Through the analysis of MPS in this paper, we show how large dimensions impact several exploration-related issues such as an increase in failed exploration.

MPS represents a good comparative point since it was designed to focus on multi-modal functions and to consider from the beginning the issues that may arise when scaling to LSGO. To limit the negative effects of concurrent exploration and exploitation, MPS has Thresheld Convergence (TC) as an integral part of its design [6]. TC is a diversification technique which separates exploration and exploitation, and it has been succesfully integrated into mutliple metaheuristic leading to improved results on multimodal functions [5].

978-1-7281-2153-6/19/$31.00 ©2019 IEEE

This paper begins with a background on exploration and Thresheld Convergence before describing MPS's design in section III. Section IV presents measurements and analysis of exploration in large dimensions, while section V presents the results on the LSGO contest. The paper finalizes with a discussion and conclusions in section VI .

II. Background

A. An analysis of Exploration

Exploration plays a critical role in the performance of metaheuristics. Two key exemplars of metaheuristics that we use as benchmarks to compare against MPS are Particle Swarm Optimization (PSO) [7] and Differential Evolution (DE) [8]. The role of exploration in these techniques has been studied extensively as can be seen in recent surveys [9], [10]. However, the ability to measure exploration (and exploitation) is hindered by their lack of precise definitions [11].

Our definitions for continuous domains are based on dividing a multi-modal search space into attraction basins which each have a single local optimum. Each point in an attraction basin has a monotonic path of increasing (for maxima) or decreasing (for minima) fitness to its local optima. A new search point is then defined to be performing "exploration" if it is in a different attraction basin than its reference solution(s), and it is defined to be performing "exploitation" if it is in the same attraction basin as (one of) its reference solution(s). We define a reference solution as a solution against which the new search point is compared. For instance, in PSO the *pbest* is a reference solution, and in DE the *target* solution would be the reference solution.

Recent research [12], [13] has built upon the above definitions to highlight the crucial effects of selection on the exploratory activities of metaheuristics. In this paper we will analyze the effects of "failed exploration" in LSGO and its relation to concurrent exploration and exploitation. We propose failed exploration as a sub-category of exploration that refers to a search solution in a new and fitter attraction basin that is rejected by selection.

In general, the comparison of an exploratory search solution against a reference solution can lead to four possible outcomes: successful exploration, successful rejection, deceptive exploration, and failed exploration [14]. These four categories occur because selection is based on the fitness of the reference solution and the fitness of the search solution, and does not take into account the fitness of the new sampled attraction basin. If we define the fitness of an attraction basins as the fitness of its local optimum, then these four possible outcomes are described as follows for a minimization problem:

- Successful exploration: A fitter attraction basin represented by a fitter exploratory search solution is accepted to replace the less fit reference solution from a less fit attraction basin.
- Successful rejection: A less fit attraction basin represented by a less fit exploratory search solution is rejected to keep the fitter reference solution from a fitter attraction basin.
- Deceptive exploration: A less fit attraction basin represented by a fitter exploratory search solution is accepted to replace the less fit reference solution from a fitter attraction basin.
- Failed exploration: A fitter attraction basin represented by a less fit exploratory search solution is rejected to keep the fitter reference solution from a less fit attraction basin.

B. Thresheld Convergence

Failed exploration tends to occur more frequently when the reference solutions are close to the local optimas of their attraction basins, and this situation is usually the result of exploitation. Thus, a key idea regarding the balance of exploration and exploitation is that early exploitation can bias the search towards sub-optimal regions. Consider a multi-modal function where all of the attraction basins have similar shapes and sizes, e.g. the popular Rastrigin function. If exactly one random solution could be sampled from each attraction basin, then there is a reasonable expectation that the fittest solutions will represent the fittest attraction basins. The process of detecting the fittest attraction basins (exploration) would then consist of selecting the fittest sample solutions. However, if some attraction basins are oversampled (e.g. due to exploitation), then the chances that worse attraction basins are represented by fitter solutions will increase, and this can bias the exploration towards the initially found (less fit) regions of the search space.

The negative effects of exploitation in known attraction basins are presented through a detailed study on the Rastrigin function in [12]. For a reference solution that has not experienced any local optimization the study shows that failed exploration occurs less than 50% of the time. However, if the reference solution is moved as little as 50% of the distance towards its local optimum (e.g. through exploitation), the rate of failed exploration can increase to over 99%.

The design of many metaheuristics does not take this biasing of exploration into account. Specifically, large explorative and small exploitative steps are often indistinguishably made during the early (explorative) stage of the search process. Thresheld Convergence aims to address this weakness by controlling the distance (search step) between a parent and its offspring solution. Thus, TC attempts to separate the processes of exploration and exploitation through the use of a "threshold" function to establish a minimum search step. Managing this step makes it possible to influence the transition from exploration to exploitation, convergence is thus "held" back until the last stages of the search process.

If the size of the search step is large enough, then the new sample solutions are more likely to represent different attraction basins. The threshold is initially set to a fraction of the search space diagonal, and it is updated over the execution of a metaheuristic by following a decay rule Equation 1. In this equation, $diagonal$ is the main diagonal of the search space, $totalFEs$ is the total number of function evaluations, and FEs is the amount of evaluations performed so far. The

α parameter determines the initial threshold and γ controls the decay rate.

$$threshold = \alpha * d * ([totalFEs - FEs]/totalFEs)^{\gamma} \quad (1)$$

Thresheld Convergence has been successfully applied to several metaheuristics such as Differential Evolution [15], Estimation Distribution Algorithms [16], Evolution Strategies [17], among others [5]. TC is applied by modifying the sampling or selection mechanism of a given metaheuristic in a way that guarantees enforcing a minimum search step. In all cases it has led to significant improvement in the optimization of multi-modal functions.

III. Minimum Population Search

Minimum Population Search is a recently developed metaheuristic specifically designed to focus on multi-modal functions and to consider from the beginning the issues that may arise when scaling to LSGO. The key ideas were initially developed for two dimensional problems using only two population members in [18], later generalized for standard dimensions in [6], and scaled towards large scale problems in [19]. One of its most relevant characteristics is its ability to perform a methodical exploration, which makes it an excellent technique for hybridizing with more exploitative heuristics such as Nelder-Mead and CMA-ES [1]. A comprehensive third-party study [20] has rated MPS-CMAES as the second best of 15 tested methods for Large-Scale Global Optimization (LSGO).

To improve performance on multimodal functions, the sampling method of MPS includes Thresheld Convergence as an integral part of its original design. As a result MPS can achieve a strong exploration with reduced levels of failed exploration. To improve scalability, MPS uses a population of the same size as the search space.

If the population size (n) becomes smaller than the dimensionality of the problem (d), then its population will define an n-1 dimensional hyperplane. New solutions generated strictly from the line segments formed among the population members will get trapped inside this n-1 dimensional hyperplane, which is a subset of the complete search space. Line segments are frequently used in many metaheuristics, e.g. difference vectors in DE, attraction vectors in PSO and mid-point crossover in Genetic Algorithms. To address this issue MPS uses a population size equal to the dimensionality of the problem $(n = d)$, new solutions are generated using difference vectors to be in a d-1 dimensional hyperplane. Full coverage of the search space is then achieved by taking a subsequent step that is orthogonal to this hyperplane.

$$trial_i = x_i + F_i * (x_i - x_c) + FOrt\_i * VOrt\_i \quad (2)$$

In each generation a new solution $trial_i$ is created from each population member x_i in two steps. First, from each parent solution x_i a step inside the $d - 1$ hyperplane (formed by the n population members) is performed. Secondly, an orthogonal step is made to search into the missing dimension. The "in-plane" step is made using the (normalized) difference vector between the parent solution x_i and the centroid of the current population x_c. The orthogonal step is made taking a random vector orthogonal $VOrt\_i$ to the parent-centroid difference vector (Fig. 1). This two-step process for generating the new trial solutions $trial_i$ is represented in Equation 2. The direction and size of the difference and the orthogonal vectors are determined by the scaling factor F_i and $FOrt\_i$, respectively.

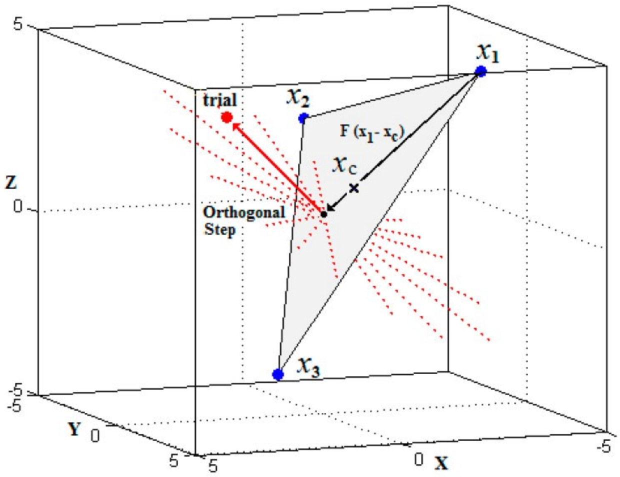

Fig. 1. Visualization of MPS search process in three dimensions. Centroid vector and orthogonal step.

Thresheld Convergence is used to enforce that a trial solution is a minimum min_step threshold distance away from its parent solution. To avoid new solutions from being sampled too far away from the best-found regions, MPS also enforces a maximum search threshold which is the double of the minimum step: $max\_step = 2 * min\_step$. In Equation 2, x_i and x_c are the parent and the centroid, respectively. The F_i factor is drawn with a uniform distribution from $[-max\_step, max\_step]$ ($x_i - x_c$ is normalized before scaling). To ensure that the distance from the new trial solution ($trial_i$) to its parent solution (x_i) stays within the acceptable $[min\_step, max\_step]$ threshold range, the $FOrt\_i$ factor is selected with a uniform distribution from $[min\_ort_i, max\_ort_i]$.

The min_ort_i and max_ort_i values are calculated using (3) and (4), respectively. The difference vector $x_i - x_c$ and the orthogonal vector $VOrt\_i$ are normalized before scaling. The $min\_step(threshold)$ values are updated using the standard Thresheld Convergence decay rule (Equation 1), with parameters $\alpha = 0.1$ and $\gamma = 3$. Once the new solutions are created, clamping is performed if necessary, and the best n solutions among the parents and offspring survive into the next generation.

$$min\_ort_i = \sqrt{max(min\_step_i^2 - F_i^2, 0)} \quad (3)$$

$$max\_ort_i = \sqrt{max(max\_step_i^2 - F_i^2, 0)} \quad (4)$$

To ensure good spacing the initial solutions are selected to be on the diagonals of the search space. Equation 5 is used to generate the initial solutions, assuming that the search space is bounded by the same lower and upper bound in each dimension: s_k is the k^{th} population member, rs_i are random numbers which can be -0.5 or 0.5, and $bound$ is the lower and upper bound in each dimension.

$$S_k = (rs_1 * bound/2, rs_2 * bound/2, \ldots, rs_n * bound/2) \quad (5)$$

IV. Experiments in LSGO

This section presents some comparative results between Minimum Population Search, Differential Evolution and Particle Swarm Optimization in large dimensions. We begin by analyzing how exploitation and exploration (including the four categories presented in section II-A) behave as dimensions increase. In order to quantitatively measure exploration and exploitation we use the Rastrigin function shown in Equation 6. The Rastrigin function has a regular fitness landscape in which every point with integer values in all dimensions is a local optimum, and every other point belongs to the attraction basin of the local optimum that is determined by rounding each solution term to its nearest integer value. These features make it possible to quickly and easily determine the attraction basin of a search point and the fitness of the local optimum of this attraction basin.

$$f(x) = 10n + \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i)\right) \quad (6)$$

These experiments use a standard implementation of MPS as described in the previous section, with a population size of $p = d$. For PSO a standard version [21] with a ring topology is used. The key parameters specified from this standardization are $\chi = 0.72984$, and $c_1 = c_2 = 2.05$ for the velocity updates given in Equation 7. Additional implementation details are the use of $p = 50$ particles [21], zero initial velocities [22] and "Reflect-Z" for particles that exceed the boundaries of the search space (i.e. reflecting the position back into the search space and setting the velocity to zero) [23].

$$\begin{aligned} v_{i+1,d} = \chi\{&v_{i,d} + c_1\epsilon_1(pbest_{i,d} - x_{i,d}) \\ &+ c_2\epsilon_2(lbest_{i,d} - x_{i,d})\} \end{aligned} \quad (7)$$

Our experiments with DE use an implementation of DE/rand/1/bin with typical parameters of population size $p = 50$, crossover $Cr = 0.9$, and scale factor $F = 0.8$ [24], [25]. Each population member x_i is considered as a *target* for replacement by a candidate solution that is constructed in two steps: creation of an intermediate solution and crossover with x_i. During the creation of an intermediate solution y_i from three distinct random solutions r_1, r_2, and r_3 in Equation 8, the scale factor F affects the "step size" from r_1 taken in the direction of the "difference vector" created with r_2 and r_3.

$$y_i = r_1 + F(r_2 - r_3) \quad (8)$$

Equation 9 defines how this intermediate solution is then crossed term-by-term in each dimension d of the search space with the target solution x_i to produce a new search/offspring solution x_i'.

$$x'_{i,d} = \begin{cases} y_{i,d} & u_d \le Cr \\ x_{i,d} & u_d > Cr \end{cases} \quad (9)$$

A. An Analysis of Exploration

The degradation in the performance of metaheuristics as dimensions increase is often attributed to the exponential increases in search space volumes. However, the sphere function is well-studied in high dimensions, and the effect of exponentially increasing search space volumes for the sphere function are not particularly severe [4]. Thus we hypothesize that the effect of large dimensions is strongly related to the exploration-exploitation balance and particularly to the increase of failed exploration.

This experiment aims to analyze how solutions are distributed among exploitation and the four exploration cases. In MPS there is no specific reference solutions against which a new solution is compared, instead the best solutions are selected among the current and the new population. Thus we will consider a new solution to perform exploitation if it is located in the same basin as any solution of the current population (otherwise it performs exploration). We will consider it a successful or deceptive exploration if an exploratory solution is accepted to the new population, successful exploration if its basins is better than the worst accepted basin, deceptive exploration otherwise. An exploratory solution that is not kept in the new population will be failed exploration if its basin is better than the worst basin in the current population, otherwise it is a successful rejection.

Figure 2 shows (percents) of how new solutions are distributed among these five categories in PSO, DE and MPS. The figure shows how these distributions vary as dimensions increase. A clear trend in PSO and DE is the predominance of exploitation as dimensions increase, but in MPS exploration slightly increases with dimensions. On the other hand, failed exploration increases with dimensions. In 10 dimension failed exploration represents less than 20% of total exploration (for all three algorithms), and in 1000 dimensions it represents a much higher percent of exploration (over a 50% for PSO and DE). It is also worth noticing that the levels of successful exploration in MPS remain steady as dimensions increase.

These results suggest that with increasing dimension, it becomes more difficult to achieve the balance between exploration and exploitation. It also shows that failed exploration becomes more dominant, most likely because of the increase of exploitation during early stages of the search process. It also shows that MPS is able to keep a higher exploration rate and less failed exploration than DE and PSO in large dimensions.

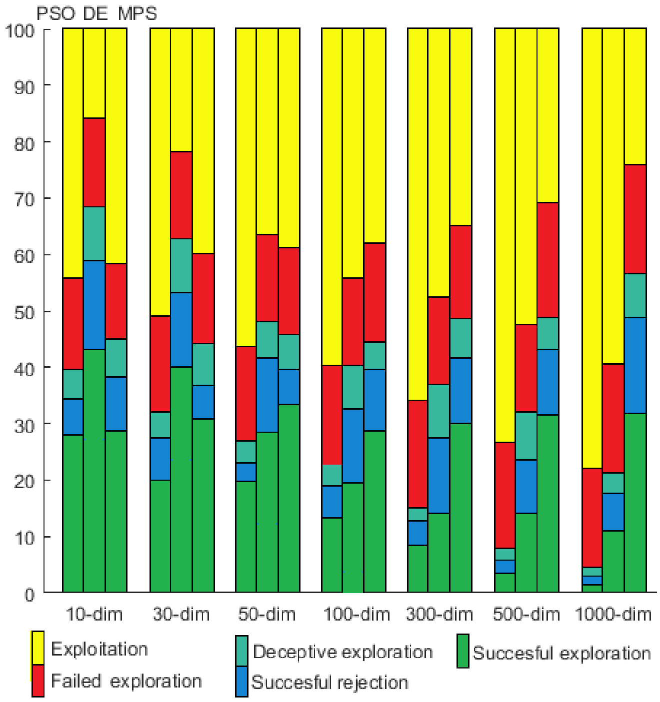

Fig. 2. Distribution of new sampled solutions among exploitation and the four categories of exploration.

B. Concurrent Exploration and Exploitation

Figure 3 shows double y plots for PSO, DE and MPS, where the left $y - axis$ represents the fitness of the convergence curve on Rastrigin and the right $y-axis$ shows the percentage of exploration and exploitation performed by each algorithm. The plots correspond to the average of 30 runs on the 1000 dimensional Rastrigin function with a total budget of $FEs = dim * 3000 = 3 \times 10^6$ function evaluations. These plots allow us to observe concurrent exploration and exploitation as the search proceeds.

The first plot shows that PSO performs a premature transition between exploration and exploitation. We can see that the point where exploitation overcomes exploration occurs very early in the search process. Although a stable amount of explorations still happens after this point, most (over 90%) of this exploration becomes failed exploration because the reference solutions (*pbest*) are already very deep in their attraction basins. All of the improvement achieved in the convergence curve after this point corresponds to exploitation of the already found attraction basins.

Differential Evolution shows less concurrent exploration and exploitation during the early stages of the search and the transition point occurs around $1 \times 10^6 FEs$. However, most of the exploration performed after $8 \times 10^5 FEs$ becomes failed exploration as just a small amount of exploitation is required for the population member to get deep into their attraction basins. In the case of MPS, Thresheld Convergence allows it to delay the transition to a very late moment of the search. This late transition leads to a higher percentage of exploration and less failed exploration as shown in Figure 2.

C. Results in LSGO

In this section we compare the performance of MPS against PSO and DE using the LSGO CEC2013 benchmark suite [26]. Based on 25 runs and a budget of $FEs = 3 \times 10^6$ function evaluations, Table I reports the mean achieved by each algorithm and the relative performances $100(a-b)/\max(a,b)$ achieved by MPS versus DE and PSO. These values indicate by what amount (percent) MPS (b) outperforms DE/PSO (a) – positive values indicate that MPS outperforms DE or PSO. A t-test between the two samples is also reported to allow a comparison on the basis of statistically significant differences at the 5% level.

In Table I mean values in bold indicate the best performance for a given function. Minimum Population Search achieves the best result in 11 of the 15 functions, while PSO and DE perform best in 2 functions each. Overall, MPS outerpforms PSO and DE by 53.4% and 58.3% respectively. If only the multi-modal functions are taken into account, then MPS outperforms PSO by 78.4%. However, the relative amount by which MPS outperforms DE is similar on multi-modal and unimodal functions. This is consistent with the higher amount of exploration achieved by DE with respect to PSO in the previous sections.

PSO achieves the best performance on functions 1 and 4, which is the Elliptic function completely (F_1) and partially separable (F_4). These results confirm that PSO performs more exploitation than DE and MPS, and it is also possible that the implemented PSO can exploit separability. In particular, neighbouring/communicating particles which are similar in a large number of dimensions can align their search trajectories along a low dimensional path that allows separability to be exploited.

Differential Evolution achieves the best results also on partially separable functions F_8 and F_{10}. The first one is also the Elliptic function, in this case with no separable subcomponents (as opposed to F_4 which has a separable subcomponent). F_{10} is the partially separable Ackley function, and DE seems to perform relatively well on this function.

Beside showing a strong overall performance when compared to PSO and DE, MPS seems to be specially effective when optimizing multi-modal functions. This comes as no surprise since that is the goal of the original design of MPS. Furthermore, MPS also shows a strong performance on overlapping (F_{12}, F_{13} and F_{14}) and non-separable (F_{15}) functions as it achieves the best results on these functions.

V. LSGO Contest

This section presents the results of the proposed algorithm on the benchmark used for the CEC LSGO special session and competition. Figure 4 shows the convergence graphs of MPS on the six selected functions by the organizers of the special session: F_2, F_7, F_{11}, F_{12}, F_{13} and F_{14}. For each function, a single convergence curve has been plotted using the average results of 25 independent runs and a logarithmic

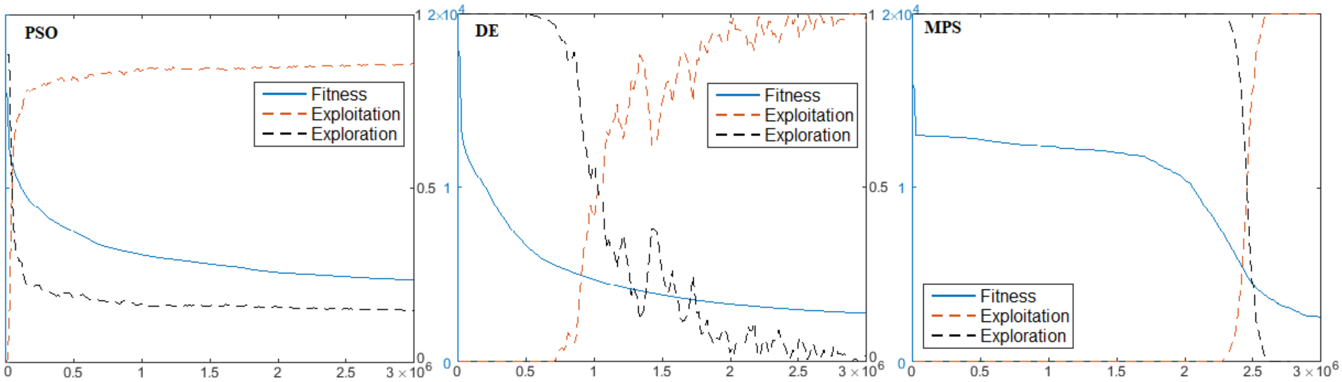

Fig. 3. Convergence curve and exploration-exploitation balance in 1000 dimensional Rastrigin function. Average over 30 runs.

TABLE I
COMPARISON OF MPS VS. PSO AND DE .

	MPS	PSO			DE		
No.	Mean	Mean	%-diff	t-test	Mean	%-diff	t-test
1	6.68e+08	**1.61e+02**	-100.0%	0.00	1.37e+09	51.2%	0.00
2	**4.20e+03**	1.25e+04	66.4%	0.00	1.52e+04	72.3%	0.00
3	**1.94e+00**	2.02e+01	90.4%	0.00	1.75e+01	88.9%	0.00
4	1.07e+11	**5.11e+10**	-52.2%	0.00	2.38e+11	55.0%	0.01
5	**1.20e+06**	5.93e+06	79.7%	0.00	9.64e+06	87.5%	0.00
6	**1.01e+03**	1.44e+05	99.3%	0.00	1.57e+03	35.6%	0.02
7	**7.19e+07**	6.33e+08	88.6%	0.00	5.90e+09	98.7%	0.00
8	2.04e+14	3.07e+14	33.5%	0.01	**3.44e+13**	-83.1%	0.00
9	**1.66e+08**	5.39e+08	69.2%	0.00	7.37e+08	77.4%	0.00
10	1.53e+06	2.37e+06	35.4%	0.02	**2.83e+03**	-99.8%	0.00
11	**2.20e+09**	1.44e+10	84.7%	0.00	3.29e+11	99.3%	0.00
12	**1.75e+04**	1.37e+06	98.7%	0.00	1.78e+11	100.0%	0.00
13	**9.87e+08**	1.38e+10	92.8%	0.00	9.45e+10	98.9%	0.00
14	**1.03e+09**	1.18e+11	99.1%	0.00	7.53e+11	99.8%	0.00
15	**1.76e+07**	2.08e+07	15.3%	0.02	2.45e+08	92.8%	0.00
Overall			53.4%			58.3%	
Multi-modal			78.4%			57.6%	

scale for the $y - axis$. It is noticeable that MPS achieves a steady improvement up to the final stages of the search process, avoiding premature convergence.

Table II contains the results of MPS on the LSGO benchmark. The best, median, worst, mean, and standard deviation of the 25 runs are presented for $FEs = 1.2 \times 10^5$, 6×10^5 and 3×10^6. MPS achieves significantly better results than DE and PSO, and these results on the LSGO contest provide useful insights. A key observation is that MPS' strong and continuous exploration hinders the algorithm from performing an efficient exploitation of the final attraction basin. This lack of exploitation is more noticeable in the unimodal functions (F_1, F_4, F_8, F_{11}, F_{13}, F_{14} and F_{15}), where MPS achieves relatively poor results. However, this strong exploration (in particular effective exploration) makes MPS an excellent companion for hybridizing with more exploitative algorithms such as CMA-ES [1], and hybrid techniques of this nature are a promising path towards developing state-of-the-art methods [20].

VI. CONCLUSIONS

Minimum Population Search was specifically designed to perform well on multi-modal functions and to consider from the beginning the issues that may arise when scaling to large scale global optimization. The increasing population size and the orthogonal step aim to maintain a strong exploration as dimensions increase, the use of a minimum step from Thresheld Convergence aims to reduce concurrent exploration and exploitation in the early stages of the search (and thus a biased exploration). New experiments highlight how this design improves exploration (especially in the context of the four sub-categories caused by fitness-based selection).

Results in Fig. 2 show that MPS does not only keep a high rate of exploration in large dimensions but that this exploration actually increases. However, it also shows that

TABLE II
RESULTS OF MPS FOR THE CEC LSGO SPECIAL SESSION.

		F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
$FEs = 1.2 \times 10^5$	Best	3.46e+10	2.90e+04	2.03e+01	9.49e+11	1.01e+07	5.52e+05	1.35e+10	1.96e+16
	Median	3.68e+10	3.07e+04	2.04e+01	1.12e+12	1.08e+07	5.89e+05	1.81e+10	2.25e+16
	Worst	3.98e+10	3.20e+04	2.05e+01	1.49e+12	1.12e+07	6.13e+05	2.00e+10	2.56e+16
	Mean	3.74e+10	3.06e+04	2.04e+01	1.17e+12	1.07e+07	5.83e+05	1.71e+10	2.29e+16
	Std	2.27e+09	1.09e+03	9.20e-02	2.07e+11	4.21e+05	2.31e+04	2.82e+09	2.63e+15
$FEs = 6 \times 10^5$	Best	1.57e+10	1.51e+04	1.63e+01	4.42e+11	8.06e+06	3.52e+05	2.38e+09	2.23e+15
	Median	1.65e+10	1.58e+04	1.65e+01	4.89e+11	8.66e+06	3.59e+05	3.41e+09	2.61e+15
	Worst	1.73e+10	1.63e+04	1.66e+01	7.15e+11	8.88e+06	3.76e+05	4.30e+09	3.74e+15
	Mean	1.66e+10	1.57e+04	1.65e+01	5.31e+11	8.61e+06	3.63e+05	3.39e+09	2.83e+15
	Std	6.51e+08	4.39e+02	1.09e-01	1.07e+11	3.34e+05	1.08e+04	6.88e+08	6.52e+14
$FEs = 3 \times 10^6$	Best	6.14e+08	3.96e+03	1.79e+00	1.01e+11	7.92e+05	5.93e+03	4.73e+07	1.04e+14
	Median	6.89e+08	4.17e+03	1.98e+00	1.05e+11	1.13e+06	6.05e+03	6.55e+07	1.95e+14
	Worst	7.00e+08	4.56e+03	2.01e+00	1.16e+11	1.55e+06	6.07e+03	1.05e+08	3.61e+14
	Mean	6.68e+08	4.20e+03	1.94e+00	1.07e+11	1.20e+06	6.01e+03	7.19e+07	2.04e+14
	Std	3.95e+07	2.34e+02	9.12e-02	5.95e+09	3.07e+05	7.01e+01	2.39e+07	9.63e+13
		F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}	
$FEs = 1.2 \times 10^5$	Best	8.21e+08	3.02e+07	3.07e+12	9.12e+11	8.25e+11	5.09e+12	1.75e+13	
	Median	8.42e+08	3.18e+07	4.37e+12	9.87e+11	8.42e+11	6.02e+12	2.66e+13	
	Worst	9.11e+08	3.36e+07	5.52e+12	1.09e+12	1.59e+12	8.03e+12	3.19e+13	
	Mean	8.54e+08	3.19e+07	4.20e+12	9.93e+11	1.01e+12	6.23e+12	2.50e+13	
	Std	3.40e+07	1.25e+06	9.23e+11	6.43e+10	3.28e+11	1.16e+12	6.46e+12	
$FEs = 6 \times 10^5$	Best	6.65e+08	1.37e+07	1.23e+11	9.77e+10	8.32e+10	6.13e+11	4.46e+08	
	Median	7.07e+08	1.63e+07	1.72e+11	1.27e+11	1.14e+11	7.72e+11	7.72e+08	
	Worst	7.10e+08	1.77e+07	3.11e+11	1.30e+11	1.35e+11	1.21e+12	1.50e+09	
	Mean	6.99e+08	1.61e+07	2.01e+11	1.20e+11	1.08e+11	8.42e+11	9.27e+08	
	Std	1.92e+07	1.48e+06	7.68e+10	1.34e+10	2.12e+10	2.27e+11	4.14e+08	
$FEs = 3 \times 10^6$	Best	1.37e+08	1.27e+06	1.74e+09	5.86e+03	6.47e+08	8.12e+08	2.12e+07	
	Median	1.69e+08	1.31e+06	2.09e+09	1.08e+04	7.41e+08	1.04e+09	2.72e+07	
	Worst	1.84e+08	1.24e+07	2.97e+09	5.21e+04	1.80e+09	1.20e+09	3.57e+07	
	Mean	1.66e+08	3.53e+06	2.20e+09	1.75e+04	9.87e+08	1.03e+09	2.76e+07	
	Std	1.91e+07	4.98e+06	4.77e+08	1.95e+04	4.88e+08	1.46e+08	6.21e+06	

failed exploration increases with dimensionality, albeit reducing concurrent exploration and exploitation in early stages of the search (Fig. 3). Although Thresheld Convergence limits the aforementioned concurrence, recent work shows that even with the integration of TC, search solutions located deep in their attraction basins may still be produced due to the stochastic nature of the sampling process [12].

To reduce failed exploration the reference solution and the newly sampled solution should have similar depths in their attraction basins. Pursuing this ideal, a new metaheuristic called Leaders and Followers (LaF) has been recently designed to emphasize these fair comparisons. The signature feature of LaF is the use of two populations: a population of leaders which contains solutions which represent the fittest attraction basins and which guide the search, and a population of followers in which fair comparisons can occur [12]. Given the increase of failed exploration in large dimensions, the integration of LaF's selection scheme into other methauristics, such as MPS, seems as a promising research line for LSGO.

Another promising research direction is to combine the strong exploration provided by MPS with more exploitative metaheuristics into a hybrid algorithm. A first step in this direction has already been taken with the design of the MPS-CMAES hybrid which is regarded as a state of the art algorithm in LSGO [20]. However, this result may be further improved by pursuing the design of an exploration-only exploitation-only hybrid. Such a hybrid aims to completely separate exploration from exploitation by developing a "purely exploratory technique". Recent publications show the potential of this approach using LaF and MPS as the exploration techniques [1], [27], but, these methods still show inefficiencies as they still perform too much failed exploration.

Finally, another aspect to research is the development of new and diverse benchmark functions that share the useful property that Rastrigin has. The ability to efficiently compute the shape and size of attraction basins in necessary to measure exploitation and exploration. Thus, such new functions will become a fundamental part in the quest of developing an exploration-only exploitation-only hybrid.

REFERENCES

[1] A. Bolufé-Röhler, S. Fiol-González, and S. Chen, "A minimum population search hybrid for large scale global optimization," in *Evolutionary Computation (CEC), 2015 IEEE Congress on*, 2015, pp. 1958–1965.

[2] M. N. Omidvar, X. Li, and X. Yao, "Cooperative co-evolution with delta grouping for large scale non-separable function optimization," in *Evolutionary Computation (CEC), 2010 IEEE Congress on*. IEEE, 2010, pp. 1–8.

[3] R. Storn, "On the usage of differential evolution for function optimization," in *Fuzzy Information Processing Society, 1996. Biennial Conference of the North American*. IEEE, 1996, pp. 519–523.

[4] S. Chen, J. Montgomery, and A. Bolufé-Röhler, "Measuring the curse of dimensionality and its effects on particle swarm optimization and differential evolution," *Appl Intell*, vol. 42, no. 3, pp. 514–526, 2015.

[5] S. Chen, J. Montgomery, A. Bolufé-Röhler, and Y. Gonzalez-Fernandez, "A review of thresheld convergence," *GECONTEC: Revista Internacional de Gestión del Conocimiento y la Tecnología*, 2015.

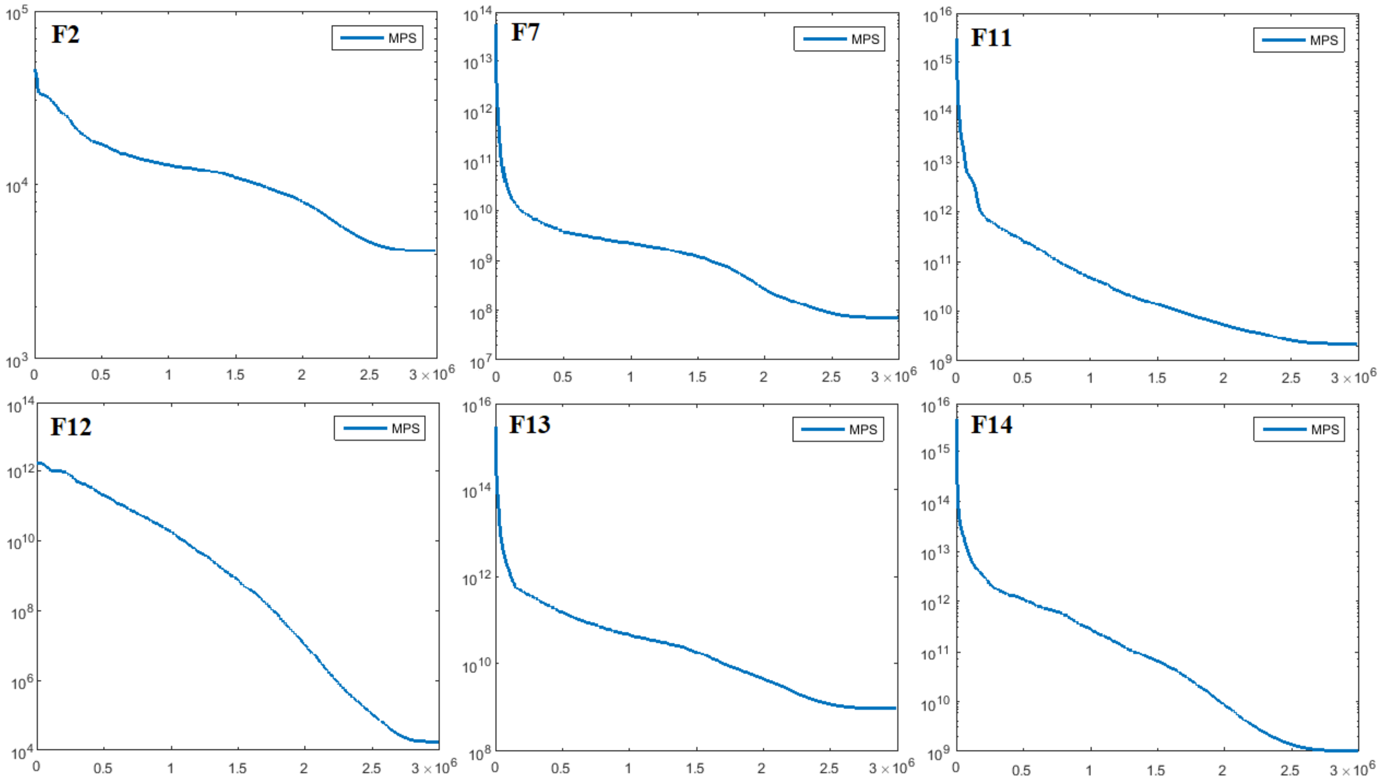

Fig. 4. Average convergence graphs of 25 runs for functions F_2, F_7, F_{11}, F_{12}, F_{13} and F_{14}.

[6] A. Bolufé-Röhler and S. Chen, "Minimum population search - a scalable metaheuristic for multi-modal problems," *Revista Investigacin Operacional*, vol. 36, no. 1, pp. 85–95, 2015.

[7] D. Bratton and J. Kennedy, "Defining a standard for particle swarm optimization," in *Swarm Intelligence Symposium, 2007. SIS 2007. IEEE.* IEEE, 2007, pp. 120–127.

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

[9] M. R. Bonyadi and Z. Michalewicz, "Particle swarm optimization for single objective continuous space problems: a review," 2017.

[10] S. Das, S. S. Mullick, and P. N. Suganthan, "Recent advances in differential evolution–an updated survey," *Swarm and Evolutionary Computation*, vol. 27, pp. 1–30, 2016.

[11] 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.

[12] Y. Gonzalez-Fernandez and S. Chen, "Leaders and followers — a new metaheuristic to avoid the bias of accumulated information," in *Evolutionary Computation (CEC), 2015 IEEE Congress on*. IEEE, 2015, pp. 776–783.

[13] D. Tamayo-Vera, S. Chen, A. Bolufé-Röhler, J. Montgomery, and T. Hendtlass, "Improved exploration and exploitation in particle swarm optimization," in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer, 2018, pp. 421–433.

[14] S. Chen, A. Bolufé-Röhler, J. Montgomery, and T. Hendtlass, "An analysis on the effect of selection on exploration in particle swarm optimization and differential evolution," in *Evolutionary Computation (CEC), 2019 IEEE Congress on*. IEEE, 2019.

[15] A. Bolufé-Röhler, S. Estévez-Velarde, A. Piad-Morffis, S. Chen, and J. Montgomery, "Differential evolution with thresheld convergence," in *2013 IEEE Congress on Evolutionary Computation*, pp. 40–47.

[16] D. Tamayo-Vera, A. Bolufé-Röhler, and S. Chen, "Estimation multivariate normal algorithm with thresheld convergence," in *Evolutionary Computation (CEC), 2016 IEEE Congress on*, 2016, pp. 3425–3432.

[17] A. Piad-Morffis, S. Estévez-Velarde, A. Bolufé-Röhler, J. Montgomery, and S. Chen, "Evolution strategies with thresheld convergence," in *Evolutionary Computation (CEC), 2015 IEEE Congress on*. IEEE, 2015, pp. 2097–2104.

[18] A. Bolufé-Röhler and S. Chen, "Minimum population search - lessons from building a heuristic technique with two population members," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*. IEEE, 2013, pp. 2061–2068.

[19] ——, "Extending minimum population search towards large scale global optimization," in *Evolutionary Computation (CEC), 2014 IEEE Congress on*. IEEE, 2014, pp. 845–852.

[20] E. Glorieux, B. Svensson, F. Danielsson, and B. Lennartson, "Constructive cooperative coevolution for large-scale global optimisation," *Journal of Heuristics*, vol. 23, no. 6, pp. 449–469, 2017.

[21] D. Bratton and J. Kennedy, "Defining a standard for particle swarm optimization," in *Swarm Intelligence Symposium, 2007. SIS 2007. IEEE.* IEEE, 2007, pp. 120–127.

[22] A. Engelbrecht, "Particle swarm optimization: Velocity initialization," in *Evolutionary Computation (CEC), 2012 IEEE Congress on*, pp. 1–8.

[23] S. Helwig, J. Branke, and S. Mostaghim, "Experimental analysis of bound handling techniques in particle swarm optimization," *IEEE Transactions on Evolutionary computation*, vol. 17, no. 2, pp. 259–271, 2013.

[24] J. Montgomery and S. Chen, "An analysis of the operation of differential evolution at high and low crossover rates," in *Evolutionary Computation (CEC), 2010 IEEE Congress on*. IEEE, 2010, pp. 1–8.

[25] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[26] X. Li, K. Tang, M. N. Omidvar, Z. Yang, K. Qin, and H. China, "Benchmark functions for the cec 2013 special session and competition on large-scale global optimization," *gene*, vol. 7, no. 33, p. 8, 2013.

[27] A. Bolufé-Röhler, D. Tamayo-Vera, and S. Chen, "An laf-cmaes hybrid for optimization in multi-modal search spaces," in *Evolutionary Computation (CEC), 2017 IEEE Congress on*. IEEE, 2017, pp. 757–764.