# A Multi-Population Exploration-only Exploitation-only Hybrid on CEC-2020 Single Objective Bound Constrained Problems

Antonio Bolufé-Röhler
*Mathematical and Computational Sciences*
*University of Prince Edward Island*
Charlottetown, Canada
ORCID 0000-0002-3181-1864

Stephen Chen
*Information Technology*
*York University*
Toronto, Canada
ORCID 0000-0001-8657-6200

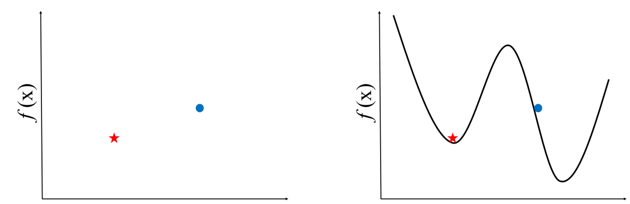
***Abstract*—Many metaheuristics attempt to "transition" a single algorithm from exploration to exploitation. Conversely, previous research has shown that it can be better for the two distinct tasks of exploration and exploitation to instead be performed by two distinct algorithms/mechanisms. This has led to the development of Exploration-only, Exploitation-only Hybrid search techniques. This paper presents a Multi-Population Exploration-only Exploitation-only Hybrid in which exploitation occurs in one population while a global search strategy performs exploration in another population. Unlike a sequential hybrid, this hybridization allows the exploratory technique (in this case Unbiased Exploratory Search) to delay convergence (up to indefinitely) which allows the hybrid system to benefit from a large budget of function evaluations. The new hybrid is evaluated on the CEC2020 test suite in the special session and competition on single objective bound constrained numerical optimization.**

***Index Terms*—hybrid heuristic, exploration, exploitation**

## I. Introduction

Population-based metaheuristics maintain a set of solutions which often serve two purposes: store the best-found solutions, and keep track of promising areas of the search space for further exploration. For example, the elistist selection in a steady-state Genetic Algorithm [1], [2] means that the mating population consists of only the best-found solutions, and the generated offspring solutions will be in the neighbourhood of these parent solutions. In Particle Swarm Optimization (PSO) [3], the personal best (*pbest*) positions store the best-found solutions (for each particle), and these positions also act as attractors which draw the particles to conduct further exploration and exploitation in their nearby neighbourhoods.

The use of one homogeneous population forces solutions in this population to serve these two distinct purposes, and this situation can be highly ineffective. In particular, the first purpose of storing the best-found solutions is easily accomplished by measuring and comparing the current/actual fitness of the solutions. However, fitness-based selection and the primacy of best-found solutions can interfere greatly with the second purpose of identifying and targeting the most promising regions of the search space. An example of this interference is shown in Fig. 1.




(a) elitist selection (b) failed exploration

Fig. 1. An example of how elitist selection can lead to failed exploration in a multi-modal search space. The star represent a reference solution (a solution against which new solutions are compared) and the dot represents a newly sampled solution. If elitist selection (a) is applied then the star will be selected, despite being located on a worse attraction basin (b).

Selection errors can be avoided in memetic algorithms (MA) [4] by having all of the initial solutions become local optima before they are compared with existing reference solutions (which in MA are also all local optima). Locally optimizing all of the initial solutions can be highly inefficient, so the implicit goal of metaheuristics such as PSO is to filter out less promising search solutions and to only optimize/perform exploitation on the most promising exploratory solutions. However, this filtering process can become susceptible to the error of failed exploration shown in Fig. 1.

The accuracy of fitness-based selection for exploratory search solutions is greatly reduced by exploitation/local optimization of the reference solution [5]. For two random solutions (e.g. from the initial population), there can be a greater than 50% chance that the fitter solution will also be from the more promising region of the search space. Conversely, the comparison of a random solution (e.g. an exploratory search solution) with a partially optimized solution (e.g. a reference solution that has moved towards its local optimum under the effects of exploitation) can lead to rates of failed exploration that approach 100%.

Two attempts to reduce the effects of failed exploration include "Thresheld Convergence" (TC) [6] and the two-population mechanism of Leaders and Followers (LaF) [5]. In Thresheld Convergence, convergence is "held" back by a threshold function which specifies a minimum distance (in continuous domain search spaces) between search solutions and reference solutions. This minimum distance helps to prevent early exploitation in attraction basins already represented by the stored population for a metaheuristic. Comparisons between new exploratory search solutions and stored reference solutions which have not been improved by exploitation can more accurate identify the most promising regions of the search space and lead to the improved performance of metaheuristics in multi-modal search spaces [6].

Another way to reduce failed exploration is to store reference solutions that can be local optima, but to avoid the direct comparison of new exploratory search solutions with them. Leaders and Followers uses two populations. The population of leaders stores the best-known solutions which guide the search process, and the population of followers stores the new search solutions. Search solutions are only compared amongst themselves within the population of followers until they reach a similar fitness (e.g. local optima) to the solutions in the population of leaders. This delayed comparison with the highly fit reference solutions leads to a reduction in the rate of failed exploration [5].

The key features of TC and LaF have been combined into a new metaheuristic called Unbiased Exploratory Search (UES). Specifically, the TC-based movement operator from Minimum Population Search (MPS) [7] is used to create the new search solutions in the population of followers. The issue of when to end this exploratory search (phase) is reduced by using UES in a Multi-Population Exploration-only Exploitation-only Hybrid (MP-EEH). Specifically, leaders can be optimized, and their improved fitness has little effect on the exploratory search being performed in the population of followers. This method of hybridization is particularly useful for non-terminating optimization processes and situations where a large number of function evaluations are possible – such as the CEC-2020 contest on Single Objective Bound Constrained Problems [8].

This paper begins with a background on exploration, MPS, and LaF. The development of UES is then presented in Section III. The Multi-Population Exploration-only Exploitation-only Hybrid is presented next in Section IV. This section also includes some comparative results of MP-EEH against some related algorithms. The results for the contest are presented in Section V, and a Discussion concludes the paper in Section VI.

## II. BACKGROUND

The analysis in this paper depends on precise definitions for exploration and exploitation [9]. We begin by defining a multi-modal search space to consist of attraction basins, each with a single local optimum. An attraction basin around an optimum includes all the points in the search space that can reach (only) that optimum by following a path on which every point has a monotonically decreasing fitness (for minimization problems). We further define that a search solution within the attraction basin of one of its reference solution(s) (e.g. a *pbest* position) represents *exploitation*, and that a search solution in a different attraction basin represents *exploration*.

In a multi-modal search space, the comparison of a search solution with a reference solution from a different attraction basin can lead to four outcomes [9]. These outcomes are based on the fitness of the reference solution, the fitness of the search solution, and the fitness of the local optima of the attraction basins containing these two solutions. For brevity, we will refer to the fitness of the local optimum of an attraction basin as the fitness of that attraction basin.

The cases of “successful exploration” and “successful rejection” are rather obvious, so it is the two error cases that require the most attention. “Deceptive exploration” occurs when a search solution representing a less fit attraction basin is accepted because it is fitter than its reference solution which represents a fitter attraction basin. “Failed exploration” occurs when a search solution which represents a fitter attraction basin is rejected because it is less fit than its reference solution which represents a less fit attraction basin.

Within this context, the goal of Minimum Population Search is to reduce failed exploration by altering the creation of search solutions, and the goal of Leaders and Followers is to reduce failed exploration by altering the comparison of search solutions with reference solutions. These two methods are orthogonal to each other, and they can be easily combined in UES. A brief introduction to MPS and LaF is now provided.

### A. Minimum Population Search

Failed exploration can be limited by reducing the number of solutions sampled in the same attraction basin, i.e. by avoiding the concurrence of exploration and exploitation. Thresheld Convergence does this by modifying the sampling strategy so that new solutions are created at least a ‘threshold’ distance away from the its reference solution(s). Managing this step makes it possible to better control the transition from exploration to exploitation, convergence is thus “held” back until the last stages of the search process [6].

Initially the minimum step is set to a fraction of the main diagonal of the search space ($diagonal$), and it is updated over the execution of a metaheuristic by following the decay rule shown in Equation 1. In this equation, $totalFEs$ is the total number of function evaluations, and $FEs$ is the amount of evaluations performed so far. The parameter $\alpha$ determines the initial threshold and $\gamma$ controls the decay rate.

$$minStep = \alpha * diagonal * \left(\frac{totalFEs - FEs}{totalFEs}\right)^{\gamma} \quad (1)$$

Thresheld Convergence has been successfully added to many popular metaheuristics [6], but its most effective implementation is in MPS which has a search operator specifically designed to use TC. This sampling strategy is the most distinct feature of MPS [7]. A key feature of the search operator is the use of an orthogonal step to guarantee full coverage of a $d$ dimensional search space (when $d \geq n$). This step is particularly important in high dimensional search spaces (e.g. $d = 1.000$) when the population size $n$ is usually set to be smaller than the dimensionality $d$ in order to promote convergence when using a restricted budget of function evaluations [10]. New solutions in MPS are based upon the line segments formed among the population members, and these solutions could otherwise become "trapped" inside the hyperplane defined by the $n$ population members.

### B. Leaders and Followers

A second approach to reduce failed exploration is to avoid the bias towards solutions with high relative fitness (e.g. reference solutions) when they are compared to solutions with low relative fitness (e.g. new sampled solutions). The metaheuristic Leaders and Followers (LaF) [5] was specifically designed with this goal in mind. LaF is not metaphor based because leaders and followers are labelled based on their features within the metaheuristic as opposed to an attempt to mimic some "real world" analog [11].

The leaders are reference solutions which guide the search, and the followers are (new) search solutions. It is noted that new followers (which are likely to have poor relative fitness) should not be immediately compared against their leaders (which can have very high relative fitness). This is the inspiration for the two-population strategy which is the most distinct feature of LaF. New followers are compared against existing followers which are more likely to have similar relative fitness, and this leads to lower rates of failed exploration. It is only when the followers have reached a similar fitness to the leaders (e.g. the median fitness of the followers population has surpassed that of the leaders population) that comparisons between leaders and followers are made. The resulting reduction in failed exploration allows LaF to outperform popular metaheuristics such as PSO and DE despite a simplistic method for solution generation [5].

## III. Unbiased Exploratory Search

A key weakness of TC in single-population metaheuristics (e.g. PSO, DE, MPS, etc) is that stored reference solutions invariably become closer to their local optima even without any exploitation. Unbiased Exploratory Search (UES) addresses this issue by using both TC during the generation of new search solutions and the two-population scheme of LaF to avoid comparing reference solutions (leaders) against newly sampled solutions (followers).

The algorithm starts by randomly initializing the populations of leaders and followers. At each iteration new trial solutions are generated using information from both populations. Between the set of new solutions and current followers, truncation selection is performed to produce the next population of followers. At the end of each iteration the median fitness of the population of followers is compared against the median fitness of the leaders. If the median of the followers is better, then a restart is performed. In a restart the two populations are merged and the best solutions become the new leaders, and the followers are randomly initialized.

New search solutions in the followers population are created by using a method similar to the one developed for MPS. Starting with a minimum step size established by the threshold function in Equation 1, a maximum search step is set as $maxStep = 2{*}minStep$. New solutions are generated starting from each leader with an initial movement that is at least the $minStep$ distance away from the leader but no further away than $maxStep$. In this way, the leaders population will guide the search by determining the regions around which exploration/exploitation is performed.

Solution generation consists of two steps. First a step is taken away from the leader towards the centroid of the population of followers; this allows the followers to direct the exploration process. Second, an orthogonal step is taken away from the (vector of the) first step. This is done by randomly generating a vector and then making it orthogonal to the difference vector between the leader and the centroid. Equation 2 shows this two-step process for generating the new trial solutions ($trial$), where $x$ is the parent (the leader), $x_c$ is the centroid (of the followers) and $orth$ corresponds to the orthogonal vector. The $f$ factor is drawn with a uniform distribution from $[-maxStep; maxStep]$.

$$trial = x + f \cdot \frac{x - x_c}{\|x - x_c\|} + fo \cdot \frac{orth}{\|orth\|} \tag{2}$$

To ensure that the distance from the new trial solution $trial$ to its parent solution $x$ stays within the acceptable $[minStep, maxStep]$ range, the $fo$ factor is selected with a uniform distribution from the $[minOrth, maxOrth]$ interval (note: the $x - x_c$ and $orth$ vectors are normalized before scaling). The $minOrth$ and $maxOrth$ values are calculated by Equations 3 and 4, respectively. Once the new solutions are created, clamping is performed if necessary. Truncation selection means the best $n$ solutions among the trial solutions and the followers will form the next generation of followers. The $\alpha$ and $\gamma$ parameters for Thresheld Convergence and the maximum number of function evaluations are parameters of the algorithm presented in Algorithm 1.

$$minOrth = \sqrt{max(minStep^2 - f^2; 0)} \tag{3}$$

$$maxOrth = \sqrt{max(maxStep^2 - f^2; 0)} \tag{4}$$

## IV. A Multi-Population Exploration-only Exploitation-only Hybrid

The exploration-only exploitation-only model for the optimization of multi-modal problems involves two distinct tasks: identifying promising attraction basins and finding the local optima in these basins. Assigning these tasks to specialized algorithms makes sense since some metaheuristics (e.g. population-based) are more efficient in terms of exploration but not well suited to fine-tune the search [12]. The first task should be performed by a purely exploratory algorithm while the second task should be assigned to a purely exploitative (local search) method. Recent results have shown the effectiveness of this approach in continuous [13] and discrete optimization problems [14].

These hybrids follow a high-level relay hybridization strategy [12], i.e. they rely on a strong exploration method to

**Algorithm 1** UES $(\alpha, \gamma, popSize, maxFEs)$

$leaders \leftarrow randomPopulation(popSize)$
$followers \leftarrow randomPopulation(popSize)$
**while** $FEs \leq maxFEs$ **do**
  $minStep \leftarrow \alpha \cdot d \cdot \left(\frac{maxFEs-FEs}{maxFEs}\right)^{\gamma}$
  $maxStep \leftarrow 2 \cdot minStep$
  $x_c \leftarrow centroid(followers)$
  **for** $i = 1 : n$ **do**
    $x_i \leftarrow leaders_i$
    $f_i \leftarrow unifRandom(-maxStep, maxStep)$
    $fo_i \leftarrow unifRandom(minOrth, maxOrth)$
    $orth_i \leftarrow orthVector(x_i - x_c)$
    $trial_i \leftarrow x_i + f_i \cdot \frac{x_i - x_c}{\|x_i - x_c\|} + fo_i \cdot \frac{orth_i}{\|orth_i\|}$
  **end for**
  $followers \leftarrow bestSolutions(followers, trial)$
  **if** $mean(followers) < mean(leaders)$ **then**
    $leaders \leftarrow selectBest(followers, leaders)$
    $followers \leftarrow randomPopulation()$
  **end if**
**end while**
**return** $\vec{x}_k \in leaders \cup followers$ with minimum $y_k$

detect the best attraction basins before using a local search strategy to achieve a fast convergence to the corresponding (local) optima. However, when a large budget of function evaluations is available, a low-level teamwork hybrid becomes a more effective hybridization strategy. As described in [12], these hybrids use a local search metaheuristic that is embedded into an (exploratory) population-based metaheuristic. Individual solutions are optimized locally as the population-based metaheuristic continues to optimize globally.

A potential risk of a low-level teamwork hybrid is that the locally optimized reference solutions may lead to an increase in failed exploration, hindering the effectiveness of further exploration. This is the motivation behind the Multi-Population approach to an Exploration-only Exploitation-only Hybrid. Local optimization occurs only in the population of leaders which has no effect on the process of selection in the population of followers (and thus cannot affect its rates of failed exploration).

The pseudocode presents a high level description of what we call a Multi-Population Exploration-only Exploitaton-only Hybrid (MP-EEH). This framework doesn't explicitly refer to the exploration and exploitation methods used. In theory, any existing metaheuristic could be used for these two processes, and we in fact use CMA-ES as the method for exploitation due to its strong overall performance on unimodal optimization problems [15]. For the exploration method, we note that most existing metaheuristics are designed to converge which is a redundant feature in a MP-EEH. Our MP-EEH thus uses Unbiased Exploratory Search which is specifically designed for use in EEH systems.

The Multi-Population configuration of an EEH has several advantages over the more common sequential configuration. The first advantage is that there is no transition point between exploration and exploitation. This reduces the need for parameter tuning which can be time consuming, ineffective, and/or problem specific. Another advantage is that local optimization can be the best way to move quickly towards the most promising regions of a search space (e.g. those with large attraction basins and/or broad plateau-like features). A key advantage for the current contest is that a MP-EEH can easily benefit from a large (up to infinite) budget of function evaluations. Without convergence, both processes can operate in a practical "steady-state" with a result produced at any time by taking the best solution from the population of leaders.

The specific coordination of the two populations in the the current MP-EEH is that the population of leaders is updated and the population of followers is restarted once the followers have reached a similar fitness to the leaders (e.g. median fitness of the followers population has surpassed that of the leaders population). The hybrid component of exploitation also happens during each of these merge points, and a random number of leaders are selected to have CMA-ES applied to them. The number of solutions to be locally optimized ($localSol$) and the budget of function evaluations allotted for this purpose ($localFEs$) are parameters of the MP-EEH.

**Algorithm 2** MP-EEH $(localSol, localFEs, maxFEs)$

$leaders \leftarrow randomPopulation(popSize)$
$followers \leftarrow randomPopulation(popSize)$
**while** $FEs \leq maxFEs$ **do**
  $trials \leftarrow createTrialSolutions()$
  $followers \leftarrow bestSolutions(followers, trial)$
  **if** $mean(followers) < mean(leaders)$ **then**
    $leaders \leftarrow selectBest(followers, leaders)$
    **for** $i = 1 : localSol$ **do**
      $leader_i \leftarrow selectLeaderRandomly()$
      $leader_i \leftarrow locallyOptimize(leader_i, localFEs)$
      $followers \leftarrow randomPopulation()$
    **end for**
  **end if**
**end while**
**return** best found solution

### A. Parameters

The proposed MP-EEH framework contains three sets of parameters corresponding to UES, CMA-ES and the hybridization strategy. For CMA-ES the standard set of parameters suggested in [15] were used; except for the $\sigma$ value which is selected much smaller to achieve a local search behavior.

The UES and MP-EEH components are recent algorithms for which no recommended parameter suggestions exist. A coarse parameter tuning was performed on the benchmark functions using a grid search algorithm. Table I shows the three set of parameters that were tuned for the competition, a brief description, and the best found values.

### B. Exploration Analysis

A common weakness in many metaheuristics is the lack of verification that a solution intended to be exploratory is

TABLE I
PARAMETERS FOR THE MP-EEH METAHEURISTIC.

| Parameters | | |
|---|---|---|
| Name | Description | Value |
| | MP-EEH | |
| $maxFEs$ | Number of function evaluations for the entire optimization | 1.0e+7 $(d = 20)$ |
| $localFEs$ | Number of function evaluations allotted to each local optimization process | $3000 + maxFEs/500$ |
| $optSol$ | Number of solutions to be locally optimized on each restart | $popSize/50$ |
| | UES | |
| $popSize$ | The size of the populations of Leaders and the population of Followers | $100 + MaxFEs/50000$ |
| $\alpha$ | determines the initial threshold respect to the search space diagonal | 0.05 |
| $\gamma$ | controls the decay of the threshold | 2.5 |
| | CMA-ES | |
| $\sigma$ | coordinate wise standard deviation (step size) | $(upBound - lowBound)/30$ |

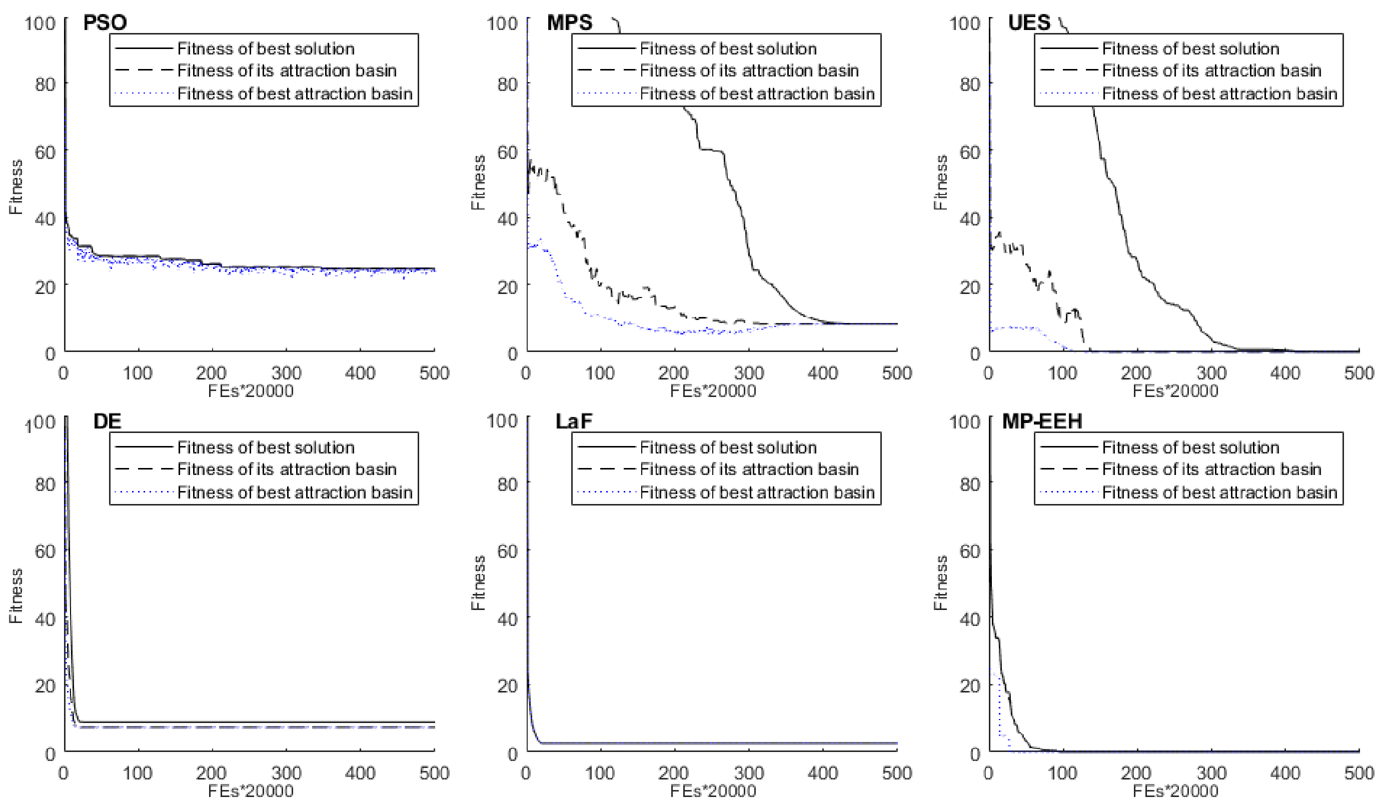



Fig. 2. Convergence curve in Rastrigin function for 20 dimensions and $1.00E + 07$ function evaluations. Average over 30 runs.

indeed in a different attraction basin. Without this verification, search solutions created with mechanisms that have the "potential for exploration" could instead lead to the creation of search solutions that our definitions would classify as "exploitation". The negative effects of exploitation in known attraction basins are presented through a detailed study on the Rastrigin function in [5]. One possible effect is "premature convergence" as reference solutions with high relative fitness cause increasing rates of failed exploration. Metaheuristics which suffer from premature convergence are unable to benefit from large budgets of function evaluations.

$$f(x) = 10n + \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i)\right) \tag{5}$$

This section presents an experiment intended to measure exploration and the convergence curves of several metaheuristics. The Rastrigin function shown in Equation 5 is used for these experiments because it 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. For brevity, we will use “the fitness of an attraction basin” to mean the fitness of the (local) optimum of an attraction basin.

This experiment uses a standard implementation of MPS with $\alpha = 0.3$, $\gamma = 3$ and population size of $p = dim$ as described in [16]. In the case of LaF the same implementation is used as described in [5], with a population size of $p = 50$. For PSO a standard version [17] with a ring topology is used. Additional implementation details are the use of $p = 50$ particles [17], zero initial velocities [18] 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) [19]. Differential Evolution is an implementation of DE/rand/1/bin with typical parameters of population size $p = 50$, crossover $Cr = 0.9$, and scale factor $F = 0.8$ [20], [21]. For UES and MP-EEH the parameters described in the previous section are used.

The experiment involves 30 independent trials in $d = 20$ dimensions using a fixed limit of $10,000,000$ total function evaluations. Fig. 2 shows averages for the fitness of the best overall solution, the fitness of its attraction basin, and the fittest attraction basin represented by any reference solution. Six different plots are presented for DE, PSO, MPS, LaF, UES and MP-EEH. Reference solutions are the current population in DE and MPS, the population of $pbest$ positions in PSO, and the population of $leaders$ in LaF, UES, and MP-EEH.

A feature to observe is the premature convergence of PSO, DE and LaF, which don’t improve much further after reaching around $10\%$ of the budget of function evaluations. In the case of PSO it can be noticed that better attraction basins are continuously found but rejected, leading to little improvement. DE and LaF converge even earlier than PSO but achieve better results. LaF performance is only surpassed by UES and MP-EEH, which reflects the effectiveness of its exploration thanks to the two-population selection scheme.

On the other hand, MPS and UES show a controlled convergence that stretches through most of the optimization process. This is a direct consequence of using Thresheld Convergence in the sampling strategy, which controls the tranistion between exploration and exploitation based on the available FEs. However, in the case of MPS, good attraction basins are lost due to failed exploration resulting from comparisons between reference solutions with high relative fitness against new (random) exploratory solutions. The convergence plot of UES reflects the combination of LaF’s and MPS’s exploratory approaches: the use of TC avoids premature convergence while the two-population selection scheme reduces the number of good attraction basins that are lost due to failed exploration. As a result, UES converges to the global optimum in each of the 30 runs.

The last plot shows the convergence curve of MP-EEH. The hybrid is governed by UES as its global search strategy, therefore TC avoids a premature convergence of the global search. However, the exploitaion with CMA-ES of some the leaders quickly leads these reference solutions towards the local optima of the best search regions. As a consequence, MP-EEH arrives much faster at the global optimum in Rastrigin.

### C. Comparing MP-EEH to related algorithms

The following experiment compares the performance of MP-EEH against its sub-component algorithms, i.e. UES and CMA-ES. The purpose of this comparison is to confirm that the hybridization strategy leads to meaningful improvements over the independent algorithms. Results are also compared against a high-level relay hybrid of UES and CMA-ES designed for large scale global optimization – which is essentially a sequential Exploration-only, Exploitation-only Hybrid (EEH).

The experiment involves 30 independent trials over the 10 contest functions in $d = 20$ dimensions with a limit of $10,000,000$ function evaluations. The MP-EEH and UES use the same parameters as described in previous sections, CMA-ES uses the parameters suggested in [15] and the EEH algorithm is the same as in [13]. Table II reports the mean errors achieved by each algorithm and the relative performances $100(a - b)/\max(a, b)$ achieved by MP-EEH versus the other algorithms. These values indicate by what amount (percent) MP-EEH (b) outperforms UES/CMA-ES/EEH (a) —- positive values indicate that MP-EEH outperforms the respective algorithm.

It can be noticed that MP-EEH achieves the best results (bolded) in 8 out of the 10 functions with an overall improvement of $67.31\%$ and $45.07\%$ over CMA-ES and UES, respectively. The Multi-Population EEH also outperforms the sequential EEH by $37.25\%$. We believe this result is primarily due to the ability of the Multi-Population hybrid to benefit more fully from the large number of function evaluations allowed for these contest function.

## V. Evaluating the UES-CMAES Multi-Population Exploration-only Exploitation-only Hybrid on the CEC 2020 Benchmark Set

The complexity of the proposed hybrid is shown in Table III. The simulation is executed on a personal computer with an Intel CPU (3.80GHz) and 8GB RAM, under Matlab 2019a programming environment. $T_0$ denotes the execution time of the following program:

$x = 0.55$;
**for** $i$ = 1:1,000,000
  $x$=$x$+$x$; $x$=$x$/2; $x$=$x$ *$x$; $x$=sqrt($x$);
  $x$=log($x$); $x$=exp($x$); $x$=$x$/($x$+2);
**end**

$T_1$ denotes the execution time of Function 7 for 200,000 evaluations of a certain dimension. $T_2$ is the running time of the proposed algorithm on Function 7 for 200,000 evaluations. $T_2$ is calculated five times, and $\hat{T}2$ is the mean value. Finally, the algorithm complexity is estimated by $(\hat{T}2 - T1)/T0$.

The results of MP-EEH are shown in Tables IV-VII. With the rules of CEC 2020 benchmark competition, the search

TABLE II
COMPARISON OF OF MP-EEH AGAINST UES, CMA-ES AND EEH .

| | MP-EHH | CMA-ES | | UES | | EEH | |
|---|---|---|---|---|---|---|---|
| No. | Mean | Mean | %-diff | Mean | %-diff | Mean | %-diff |
| 1 | **0.00E+00** | 0.00E+00 | 0.00% | 4.81E+01 | 100.0% | 0.00E+00 | 0.00% |
| 2 | **1.70E+02** | 2.16E+03 | 92.12% | 7.23E+02 | 76.46% | 5.87E+02 | 71.04% |
| 3 | 2.33E+01 | 5.42E+01 | 57.06% | **7.91E+00** | -65.99% | 2.04E+01 | -12.09% |
| 4 | **4.25E-01** | 2.31E+00 | 81.58% | 1.66E+00 | 74.44% | 1.27E+00 | 66.46% |
| 5 | **2.36E+02** | 1.22E+03 | 80.62% | 8.79E+02 | 73.11% | 6.40E+02 | 63.07% |
| 6 | 4.27E+01 | 4.90E+02 | 91.30% | **2.74E+01** | -35.72% | 5.74E+01 | 25.59% |
| 7 | **7.77E+01** | 7.17E+02 | 89.17% | 5.26E+02 | 85.21% | 1.92E+02 | 59.49% |
| 8 | **8.00E+01** | 2.48E+03 | 96.78% | 2.25E+02 | 64.46% | 1.00E+02 | 20.00% |
| 9 | **9.67E+01** | 4.31E+02 | 77.59% | 4.01E+02 | 75.87% | 4.00E+02 | 75.84% |
| 10 | **4.01E+02** | 4.30E+02 | 6.90% | 4.12E+02 | 2.81% | 4.14E+02 | 3.15% |
| Overall | | | 67.31% | | 45.07% | | 37.25% |

space is set to $[-100, 100]$ for each variable. If the solution error is smaller than $10e{-}08$, the error is set to $0$. The maximum number of function evaluations $50,000$, $100,000$, $3,000,000$ and $10,000,000$ respectively for dimensions 5, 10, 15 and 20. The number of runs for each function is 30, and the best, worst, median, mean, and standard deviation of the solution error are recorded.

Results show the algorithm has no difficulty finding the global optimum in the unimodal Function 1, in any dimension. In $5D$ the algorithm manages to find the global optimum at least once in 6 out of the 10 functions. But as expected, performance deteriorates with increasing dimensions. In $20D$ a solution with an error lower than $1.00$ is found in $4$ functions. The proposed algorithm also obtains a median or mean error value less than 100 in 7 of the functions. We believe these results are quite strong, and we look forward to comparing these results against other state-of-the-art methods entered into the competition.

TABLE III
COMPUTATIONAL COMPLEXITY

| | $T0$ | $T1$ | $\hat{T2}$ | $(\hat{T2} - T1)/T0$ |
|---|---|---|---|---|
| D=5 | 0.031 | 0.391 | 11.065 | 344.322 |
| D=10 | 0.031 | 0.391 | 9.837 | 304.709 |
| D=15 | 0.031 | 0.453 | 9.875 | 303.935 |

## VI. DISCUSSION

The motivation behind an exploration-only exploitation-only hybrid comes from acknowledging that exploration and exploitation are different search objectives which can benefit from different search strategies. Further, the concurrence of both processes can interfere with the ability to perform exploration. It has been shown that a bias arises to favour (locally optimized) reference solutions when they are compared against (non-locally optimized) search solutions. These biased comparisons can be limited by separating the locally optimized solutions from those that are not (as in LaF [5]) or locally optimizing every solution prior to the comparison as a means to eliminate selection error (as in memetic algorithms [4]).

From a design perspective the Multi-Population EEH provides a combination of both approaches. Reference solutions are kept separated as in LaF, and they can also be locally optimized. Further, this local optimization is done explicitly by using a dedicated local search algorithm as opposed to accidentally by the same mechanism that was designed to perform

TABLE IV
RESULTS FOR 5D

| Func. | Best | Worst | Median | Mean | Std.Dev. |
|---|---|---|---|---|---|
| 1 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| 2 | 1.25E-01 | 1.32E+02 | 1.34E+01 | 2.46E+01 | 3.92E+01 |
| 3 | 0.00E+00 | 6.95E+00 | 3.48E+00 | 3.43E+00 | 2.13E+00 |
| 4 | 0.00E+00 | 2.25E-01 | 1.03E-01 | 1.02E-01 | 8.06E-02 |
| 5 | 0.00E+00 | 4.25E+01 | 7.09E+00 | 1.03E+01 | 9.90E+00 |
| 6 | 6.64E-03 | 5.11E-01 | 1.55E-01 | 2.38E-01 | 1.77E-01 |
| 7 | 1.93E-04 | 1.19E+02 | 9.00E-03 | 2.01E+01 | 4.48E+01 |
| 8 | 0.00E+00 | 2.67E-08 | 1.65E-08 | 1.68E-08 | 5.30E-09 |
| 9 | 1.00E+02 | 2.00E+02 | 1.00E+02 | 1.03E+02 | 1.83E+01 |
| 10 | 2.00E+02 | 3.47E+02 | 3.47E+02 | 3.10E+02 | 5.42E+01 |

TABLE V
RESULTS FOR 10D

| Func. | Best | Worst | Median | Mean | Std.Dev. |
|---|---|---|---|---|---|
| 1 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| 2 | 6.25E-02 | 1.45E+02 | 1.59E+01 | 3.48E+01 | 4.61E+01 |
| 3 | 2.77E+00 | 1.34E+01 | 1.23E+01 | 1.06E+01 | 3.62E+00 |
| 4 | 0.00E+00 | 2.86E-01 | 6.93E-02 | 8.67E-02 | 7.24E-02 |
| 5 | 2.41E+00 | 8.70E+01 | 2.64E+01 | 3.07E+01 | 2.09E+01 |
| 6 | 3.67E-01 | 1.22E+01 | 9.38E-01 | 1.23E+00 | 2.08E+00 |
| 7 | 4.54E-01 | 1.05E+01 | 1.11E+00 | 1.57E+00 | 1.86E+00 |
| 8 | 3.55E-08 | 1.00E+02 | 4.57E-08 | 4.67E+01 | 5.07E+01 |
| 9 | 1.00E+02 | 2.00E+02 | 1.00E+02 | 1.03E+02 | 1.83E+01 |
| 10 | 1.00E+02 | 3.98E+02 | 3.98E+02 | 3.38E+02 | 1.06E+02 |

TABLE VI
RESULTS FOR 15D

| Func. | Best | Worst | Median | Mean | Std.Dev. |
|---|---|---|---|---|---|
| 1 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| 2 | 4.64E+00 | 3.71E+02 | 1.29E+02 | 1.15E+02 | 8.46E+01 |
| 3 | 5.97E+00 | 2.15E+01 | 1.96E+01 | 1.71E+01 | 5.03E+00 |
| 4 | 8.89E-02 | 5.18E-01 | 2.62E-01 | 2.89E-01 | 1.31E-01 |
| 5 | 1.44E+01 | 1.96E+02 | 8.72E+01 | 9.16E+01 | 4.80E+01 |
| 6 | 9.90E-01 | 3.12E+01 | 1.88E+00 | 4.07E+00 | 5.90E+00 |
| 7 | 1.03E+00 | 1.35E+02 | 4.45E+00 | 2.15E+01 | 3.89E+01 |
| 8 | 6.48E-08 | 1.00E+02 | 1.00E+02 | 6.00E+01 | 4.98E+01 |
| 9 | 1.00E+02 | 1.00E+02 | 1.00E+02 | 1.00E+02 | 2.19E-08 |
| 10 | 2.00E+02 | 4.00E+02 | 4.00E+02 | 3.90E+02 | 4.03E+01 |

TABLE VII
RESULTS FOR 20D

| Func. | Best | Worst | Median | Mean | Std.Dev. |
|---|---|---|---|---|---|
| 1 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| 2 | 1.02E+01 | 3.62E+02 | 1.40E+02 | 1.70E+02 | 9.42E+01 |
| 3 | 9.56E+00 | 2.98E+01 | 2.63E+01 | 2.33E+01 | 6.13E+00 |
| 4 | 1.19E-01 | 6.96E-01 | 4.35E-01 | 4.25E-01 | 1.41E-01 |
| 5 | 7.69E+01 | 4.00E+02 | 2.35E+02 | 2.36E+02 | 7.80E+01 |
| 6 | 1.95E+00 | 1.22E+02 | 1.33E+01 | 4.27E+01 | 5.23E+01 |
| 7 | 2.42E+00 | 2.28E+02 | 6.42E+01 | 7.77E+01 | 6.52E+01 |
| 8 | 1.24E-07 | 1.00E+02 | 1.00E+02 | 8.00E+01 | 4.07E+01 |
| 9 | 1.01E-06 | 1.00E+02 | 1.00E+02 | 9.67E+01 | 1.83E+01 |
| 10 | 3.99E+02 | 4.10E+02 | 3.99E+02 | 4.01E+02 | 3.78E+00 |

exploration. An advantage of this framework is that reference solutions can move more quickly towards the best regions without having to rely on successive restarts to reach local optima. This contrasts with many metaheuristics that aim to simultaneously perform both exploration and exploitation, so local optimization generally requires the overall convergence of the algorithm.

Convergence is a natural enemy of exploration as it localizes the search to a very narrow region. A version of MP-EEH with unlimited (*a priori*) function evaluations is thus a promising future line of research. In such a hybrid, UES could be executed with parameters designed to never converge. The advantage of this type of optimization system is the ability to tackle extremely difficult problems where improved solutions can be used at any time. The extraction of a solution does not interfere with the on-going optimization (e.g. by requiring convergence), and the existence of optimized solutions can guide future exploration (e.g. in globally convex search spaces) without undue biasing of the search behaviour (e.g. through increased failed exploration or restriction to the neighbourhood around the known local optima). As a comparison, it is noted that Tabu Search can operate in an "infinite mode", but it is less able to use a set of known local optima to improve search directions [22].

## ACKNOWLEDGMENT

This material is based upon work supported by Google Cloud.

## REFERENCES

[1] J. H. Holland *et al.*, "Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence," 1975.

[2] L. D. Whitley *et al.*, "The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best." in *ICGA*, vol. 89, 1989, pp. 116–123.

[3] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4. IEEE, 1995, pp. 1942–1948.

[4] P. Moscato *et al.*, "On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms," *Caltech concurrent computation program, C3P Report*, vol. 826, p. 1989, 1989.

[5] 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.

[6] 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.

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

[8] Y. CT, P. KV, S. PN, L. JJ, A. MZ, Q. BY, A. NH, and B. P, "Problem definitions and evaluation criteria for the cec 2020 special session and competition on single objective bound constrained numerical optimization," *Computational Intelligence Laboratory, Zhengzhou University. Zhengzhou, China: Tech. Rep*, 2019.

[9] 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.

[10] 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.

[11] K. Sörensen, "Metaheuristics—the metaphor exposed," *International Transactions in Operational Research*, vol. 22, no. 1, pp. 3–18, 2015.

[12] E.-G. Talbi, *Metaheuristics: from design to implementation*. John Wiley & Sons, 2009, vol. 74.

[13] A. Bolufé-Röhler and D. Tamayo-Vera, "Towards an exploration-only exploitation-only hybrid for large scale global optimization," in *2020 IEEE Congress on Evolutionary Computation (CEC), submitted*. IEEE, 2020.

[14] ——, "Machine learning based metaheuristic hybrids for s-box optimization," *J Ambient Intell Human Comput*, 2020.

[15] N. Hansen, S. D. Müller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es)," *Evolutionary computation*, vol. 11, no. 1, pp. 1–18, 2003.

[16] 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.

[17] 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.

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

[19] 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.

[20] 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.

[21] 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.

[22] F. Glover, "Tabu search—part i," *ORSA Journal on computing*, vol. 1, no. 3, pp. 190–206, 1989.