# Particle Swarm Optimization with *pbest* Perturbations

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

Imran Abdulselam
*Information Technology*
*York University*
Toronto, Canada
ORCID 0000-0001-8457-5546

Naeemeh Yadollahpour
*Information Technology*
*York University*
Toronto, Canada
ORCID 0000-0002-2022-6619

Yasser Gonzalez-Fernandez
*Information Technology*
*York University*
Toronto, Canada
ORCID 0000-0002-5685-3876

***Abstract*—Restarts are a popular remedy to address (premature) convergence in metaheuristics. In Particle Swarm Optimization, it has been observed that swarms often "stall" as opposed to "converge". A stall occurs when all of the forward progress that could occur is instead rejected as failed exploration. Since the swarm is in a good region of the search space with the potential to make more progress, a (random) restart could be counter productive. We instead introduce a method to address the stall mechanism. The introduction of perturbations to the *pbest* positions leads to significant improvements in the performance of standard Particle Swarm Optimization.**

***Index Terms*—exploration, selection, convergence**

## I. Introduction

A distinguishing feature between metaheuristics for optimization and random search is the use of reference solutions to guide the search process. Reference solutions represent a form of accumulated knowledge about the search process, and they help the metaheuristic to target future search solutions towards areas that are perceived to be the most promising regions of the search space. Progress to the search process thus depends on the updating of these reference solutions; otherwise, the same areas of the search space will be targeted over and over again.

Particle Swarm Optimization (PSO) [1] is the founding exemplar of swarm intelligence techniques for continuous domain search spaces. Its reference solutions are the personal best (*pbest*) positions of each particle. Current solutions generate new search solutions under the influence of attraction vectors towards a set of these reference solutions. In particular, particles in PSO have oscillatory search trajectories around their reference solutions [2]. The movement/updating of reference solutions is thus critical to the ability of PSO to guide its particles towards better regions of the search space.

A personal best position represents a form of elitism. The fitness of a *pbest* position improves monotonically, so the selection mechanism of PSO leads to the rejection of all current/search positions which are less fit. In a multi-modal search space, the comparison of a current position with a *pbest* position from a different attraction basin can lead to four outcomes [3]. These outcomes are based on the fitness of the *pbest* position/reference solution, the fitness of the current position/search solution, and the fitness of the local optima of the attraction basins around these two positions/solutions. For brevity, we will refer to the fitness of the local optimum of an attraction basin as the fitness of that attraction basin.

Case one is "successful exploration": the current position which represents a fitter attraction basin is compared against its less fit *pbest* position which represents a less fit attraction basin. Based on its superior fitness, the current position is accepted to become the new *pbest* position. Case two is "successful rejection": a less fit current position representing a less fit attraction basin is rejected after it is compared against its fitter *pbest* position which represents a fitter attraction basin. Case three is "deceptive exploration": a current position representing a less fit attraction basin is accepted because it is fitter than its *pbest* position which represents a fitter attraction basin. Case four is "failed exploration": a current position which represents a fitter attraction basin is rejected because it is less fit than its *pbest* position which represents a less fit attraction basin.

The demonstration of failed exploration in [3] leads to our differentiation of "convergence" and "stall" in PSO. At convergence, all *pbest* positions are located in the same attraction basin, and the search trajectories of the particles no longer generate any (exploratory) search points in other attraction basins. In a stall situation, the *pbest* positions still represent many different attraction basins and the particles which have attractors in different attraction basins can still explore new attraction basins as part of their search trajectories. However, all of these search solutions are rejected, including many under the category of failed exploration.

It is shown that the typical operation of PSO involves an initial phase where large amounts of successful exploration can occur. After this phase, PSO often enters a stall pattern where large amounts of failed exploration occur. We introduce a new mechanism to PSO which allows replication of the initial phase with large amounts of successful exploration. We then show that this modification can lead to large improvements in the performance of PSO on multi-modal functions.

The new mechanism of *pbest* perturbations is based on recent insights on the effects of selection on exploration

This research has received funding support from the Faculty of Liberal Arts & Professional Studies at York University through a Dean's Award for Research Excellence and an NSERC Bridging Grant.

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

which are reviewed in the Background Section. We then formally introduce the concept of "stall" and contrast that with "convergence" is Section III. Section IV provides a baseline of restarts against which perturbations can be compared with in Section V. Section VI develops line searches which are necessary to apply perturbations to the benchmark functions in Section VII. The paper then closes with a Discussion.

## II. Background

The analysis in this paper depends on precise definitions for exploration and exploitation [3]. 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) or increasing fitness (for maximization problems). We further define that a current position within the attraction basin of one of its attractors (e.g. *pbest* and *lbest*) represents *exploitation*, and that a current position in a different attraction basin represents *exploration*.

In contrast, the critical concepts of exploration and exploitation often lack clear definitions. A broad survey of over 100 papers led Crepinšek, Liu, and Mernik to the unexpected conclusion that "exploration and exploitation have only been implicitly defined in EAs." [4] Compared to our new definitions, the imprecise term often used in the literature would mean "the potential for exploration". A specific example is "Diversity is related to the notions of exploration and exploitation: the more diverse a swarm is, the more its particles are dispersed over the search space, and the more the swarm is exploring." [5]

Diversity based definitions for exploration are problematic on two fronts. First, diversity is often a measure of the inputs used to create exploratory search solutions –- not a measure of the produced solutions. For example, even in highly diverse swarms, PSO has a non-zero chance to produce an "exploratory" solution that is identical to its *pbest* position/reference solution. As such, large amounts of exploitation can occur under the guise of exploration, and this undesired exploitation can lead to premature convergence and the poor performance of PSO. The second issue is that there is no guarantee that diversity in exploratory search solutions can survive selection. In fact, previous research indicates that diversity is routinely eliminated by selection [3].

Our definition for exploration, and our ability to measure its occurrence, depends on the ability to determine the attraction basin that is associated with every solution in the search space. For general search spaces (e.g. real-world problems), this can often be infeasible. These experiments are thus limited to artificial search spaces such as the Rastrigin function shown in Equation 1. As first described in [6], this function has a regular fitness landscape in which every point with integer values in all dimensions is a local optimum, and all other points belong 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 identify the attraction basin for a solution and to easily calculate the fitness of its local optimum.

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

The Rastrigin function has regularly located attraction basins of similar size and shape. These attraction basins have a sinusoidal shape, and the overall search space has perfect global convexity. In general, these search space features lead to easier optimization problems. Despite the limitations of this experimental design, we believe that the identified effects will also apply to more complicated problems and search spaces. In particular, experiments across benchmark function sets are used to determine if the insights gained from the detailed analysis on Rastrigin affect the results for other types of functions.

Our experiments use a version of standard particle swarm optimization [7] with a ring topology. 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 2. Additional implementation details are the use of $p$ = 50 particles [7], zero initial velocities [8], 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) [9]. The source code for this implementation is available online [10].

$$\begin{aligned} v_{i+1,d} = \chi\{&v_{i,d} + c_1\epsilon_1(pbest_{i,d} - x_{i,d}) \\ &+ c_2\epsilon_2(lbest_{i,d} - x_{i,d})\} \end{aligned} \tag{2}$$

Many modified versions of PSO exist [11]. Some of these versions modify weights (e.g. [2], [12]), communication topologies (e.g. [13], [14]), velocity update rules (e.g. [15], [16]), or population size (e.g. [17], [18]). However, the broad survey in [11] indicates no significant research activity to address the effects of failed exploration. The results of this paper should thus be equally relevant to these types of modified versions of PSO which have similar selection strategies to the standard version of PSO (i.e. updating *pbest* only when fitter current positions are found). Specifically, any swarm intelligence method which can experience stall (as a result of its elitist selection) may be able to benefit from the methods and insights we now present.

## III. Stall vs. Convergence

The general assumption in PSO is that if the fitness of the best overall particle has not improved for a large number of iterations, then the swarm has converged. However, from a linguistic perspective, the word "converge" in English implies that a converged swarm should have all of the particles reach the same location of the search space. The current experiments using the Rastrigin function clearly show that this is not the typical case for the operation of PSO (in higher dimensions).

These experiments apply standard PSO to the Rastrigin function in $d = 3$ and $d = 30$ dimensions. A total of 10,000$d$

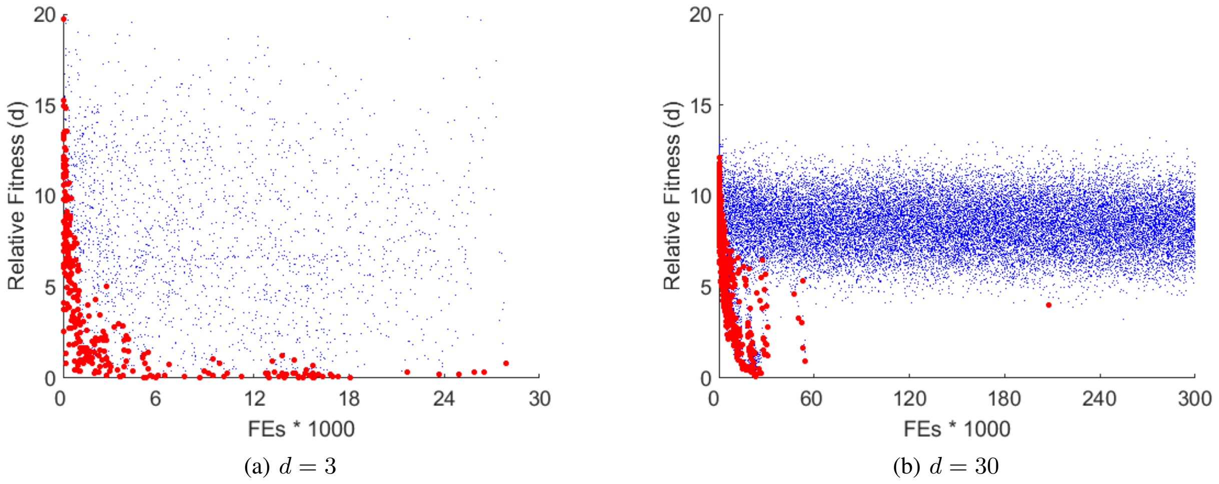


Fig. 1. Convergence vs. stall in PSO. In $d = 3$ dimensions (a), the swarm has converged after the last red circle representing successful exploration. In $d = 30$ dimensions (b), large amounts of (failed) exploration continue to occur in more promising regions of the search space as indicated by the blue dots. However, swarm progress has largely stalled as indicated by the lack of red circles which represent successful exploration.

TABLE I
PERFORMANCE OF PSO

| Dimensions | mean | std dev |
|---|---|---|
| 3 | 0.0 | 0.0 |
| 30 | 66.6 | 14.0 |

function evaluations are allowed. The typical performance metric of measuring the fitness of the best overall particle leads to the results shown in Table 1. Although not shown at this time (see Figure 2(b) instead), a plot of the best overall fitness with time/iteration shows convergence curves that approach and then reach the final overall fitness in approximately 15% of the allocated function evaluations. It would then be typical to state that performance of PSO in $d = 30$ dimensions has suffered from "premature convergence".

The plots in Fig. 1 are for an insightful trial from the 30 independent trials used to produce the results in Table I, and they demonstrate a key difference between "convergence" and "stall". The plots show the relative fitness of exploratory search positions from fitter attraction basins. Relative fitness refers to the difference between the fitness of the current position and the fitness of its attraction basin (which is also the nearest local optimum in Rastrigin). Exploratory search solutions are with respect to the *pbest* and *lbest* attraction basins which would represent exploitation. The small blue dots represent failed exploration (i.e. rejection), and the large red dots represent successful exploration.

In Fig. 1(a) for $d = 3$ dimensions, PSO converges. The first (i.e. *gbest*) particle reaches the global optimum after about 15% of the allocated function evaluations, and all of the other particles reach that attraction basin by the end of the run. After the last large red dot for successful exploration on the far right, there are no more dots at all. As the run progresses, the density of red and blue dots decreases as fewer and fewer particles produce exploratory search solutions (in fitter attraction basins). The lack of any dots at the end of the run is a sign of convergence – all of the trajectories of the particles stay within the attraction basins of their respective *pbest* positions.

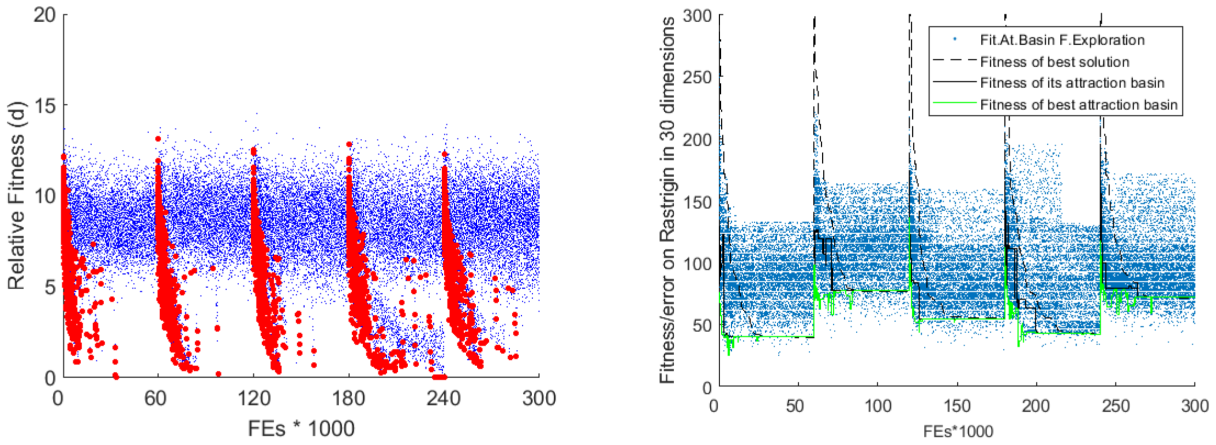
In Fig. 1(b) for $d = 30$ dimensions, PSO stalls. The first (i.e. *gbest*) particle often reaches the final fitness level after about 15% of the allocated function evaluations, but then the swarm stalls. There are still large amounts of exploration in fitter attraction basins that could lead to better overall solutions being found, but almost all of it is rejected in the form of failed exploration.

In both sub-plots for Fig. 1, it is worth noting that the relative fitness of red dots covers a large range during the first few iterations. An initial position with poor relative fitness can be improved upon by an exploratory search solution in a fitter attraction basin with similar relative fitness. However, extensive studies in [3], [6] show that *pbest* positions rapidly approach their local optima with a relative fitness of zero. This improving relative fitness of the *pbest* positions forces the relative fitness of the exploratory search solution to similarly improve in order to achieve successful exploration.

A swarm that has converged (e.g. Fig. 1(a)) requires techniques to increase exploration since no additional exploratory search positions are being generated. However, a swarm that has stalled (e.g. Fig. 1(b)) is still producing many useful exploratory search positions. The problem is that (almost) all of these useful exploratory search positions are being rejected as failed exploration. Our strategy is to reduce the rates of failed exploration through the use of *pbest* perturbations. These perturbations are distinctly different from restarts which will be analyzed in the next section.

## IV. THE EFFECTS OF RESTARTS

The simplest restart strategy is to perform $n$ sets of PSO for 10,000$d/n$ function evaluations each. Although more ad-



(a) effects of selection

(b) swarm performance

Fig. 2. The effects of selection are similar for each restart (a). The performance of each sub-run is also similar in that the best particle (dashed line) quickly reach the fitness of its local optimum (solid black line), and then there is little improvement thereafter (b). Essentially, each sub-run has stalled, and this is indicated by both the lack or red circles in the relative fitness plot (a) and by the lack of improvement in the performance plot (b). Each sub-run is independent, and this can lead to large variations in performance, especially in a negative direction.

TABLE II
PSO WITH RESTARTS

| Dimensions | mean | std dev |
|---|---|---|
| 3 (best of five) | 0.0 | 0.0 |
| 30 (first sub-run) | 69.8 | 14.8 |
| 30 (best of five) | 51.5 | 10.7 |

vanced restart strategies exist (e.g. [19]–[21]), the simplicity of this restart strategy helps our experimental analysis to highlight several general characteristics of all restart strategies. The time to converge/stall is approximately 15% of the allotted function evaluations. Setting $n = 5$ allows each sub-run 20% of the originally allocated function evaluations which is usually sufficient for the best *pbest* (i.e. *gbest*) position to reach a local optimum.

The results in Table II show that restarts have a minuscule effect for $d = 3$ dimensions – all sub-runs in all 30 trials reach the attraction basin of the global optimum, but some *gbest* positions are a negligible amount away from a 0 fitness and not all particles have converged to the basin with the global optimum. The effects of restarts for $d = 30$ dimensions is highlighted by recording results after the first sub-run and for the best overall of the $n = 5$ sub-runs. As can be expected, the result of the first sub-run, which is equivalent to PSO with fewer allotted function evaluations, is worse. However, PSO has largely stalled, and the improvements achieved by restarts show that it is more effective to completely restart (even randomly) than to hope for fortuitous occurences of successful exploration.

Additional insights into the effects of restarts are available through our observations of successful exploration and failed exploration. Fig. 2 shows two perspectives. In Fig. 2(a), the relative fitness of exploratory search solutions which represent successful exploration (large red dots) and failed exploration (small blue dots) are shown. It can be seen that each restart leads to a large burst of successful exploration, and that each of the $n = 5$ sub-runs are largely similar.

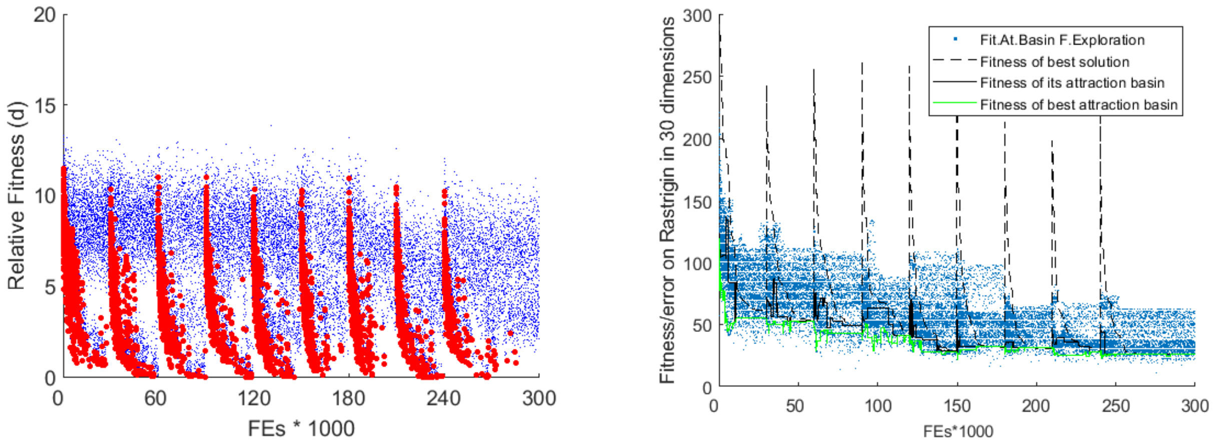
Fig. 2(b) shows the actual/absolute fitness for each sub-run. The dashed line is the actual fitness of the best overall (*gbest*) position, and the solid line is the fitness of its attraction basin. Another solid line below (in green) shows the fitness of the best attraction basin for any *pbest* position. The light blue dots show the fitness for the attraction basins that are rejected under the category of failed exploration.

One observation to highlight is that each of the restarts/sub-runs operates at a different actual fitness (even though the relative fitness is quite similar). Through this variation, the best of the five sub-runs leads to a better overall result than a single sub-run and a single standard execution of PSO (see Table I and Table II). Conversely, all of the sub-runs have blue dots that have better fitness (i.e. are below) the bottom solid line. These dots indicate that useful exploration is still occurring, and that the restarts can also relocate the swarm into a less promising region of the search space.

These two observations lead us towards the goal of achieving increased (bursts of) successful exploration with less disruption than (random) restarts. It can be seen in Fig. 2(b) that the dotted line has converged with the first solid line, and this indicates that the *pbest* position has approached its local optimum. The studies in [3], [6] show that local optimization of (*pbest*) reference solutions leads to drastically increased rates of failed exploration. What is necessary is not a restart of the swarm, but a reversing of the local optimization of the *pbest* position.

## V. *pbest* PERTURBATIONS

The local optima in the Rastrigin function are located at the points in the search space with integer values in all dimensions. The size of the attraction basin is thus ±0.5 in each dimension



(a) effects of selection

(b) swarm performance

Fig. 3. Perturbations have similar effects on selection as restarts (a). These bursts of successful exploration are also visible in the performance of the swarm (b) – especially in the movements of the solid lines. There is also a steadier trajectory of continuous improvement with perturbations compared to the more random changes caused by restarts.

TABLE III
PSO WITH *pbest* PERTURBATIONS

| **Dimensions** | **mean** | **std dev** |
|---|---|---|
| 3 | 0.0 | 0.0 |
| 30 | 29.0 | 4.9 |

around each local optima. The experiments in [6] show that a random solution in the $\mathbf{x} = \mathbf{-3}$ basin can achieve successful exploration over 75% of the time whereas a solution that has moved even half of this distance towards its local optimum will achieve less than 1% successful exploration. To improve the rates of successful exploration without losing forward progress of the swarm, we "perturb" the *pbest* (and current position) of each particle to become a random point within its current attraction basin. Assuming that the *pbest* position has reached its local optimum, the perturbation is to add a random uniform value in the range of [-0.5, 0.5] to each term of the solution.

The perturbations occur within a single trial of PSO, so the swarm does not have to converge after the interim perturbations. Thus, rather than using the restart level of 20% of the function evaluations which allows the fitness of the best overall position to reach a local optimum (see Fig. 2(b)), a perturbation level of 10% of the function evaluations is chosen which is close to the end of the initial wave of successful exploration (see Fig. 2(a)). Perturbations thus occur at 10, 20, 30, 40, 50, 60, 70, and 80 percent of the allotted function evaluations (which gives the final perturbation a good chance to reach its local optimum).

The addition of perturbations lead to much larger improvements than the addition of restarts – see Table III. The actual effects of perturbations are better observed in Fig. 3. First, Fig. 3(a) shows that the desired replication of bursts of successful exploration has been achieved. Fig. 3(b) then shows that these bursts occur within the context of a progressing, single search trajectory. In particular, the "backwards" step after the first restart in Fig. 2(b) is replaced by steadier progress as seen in Fig. 3(b).

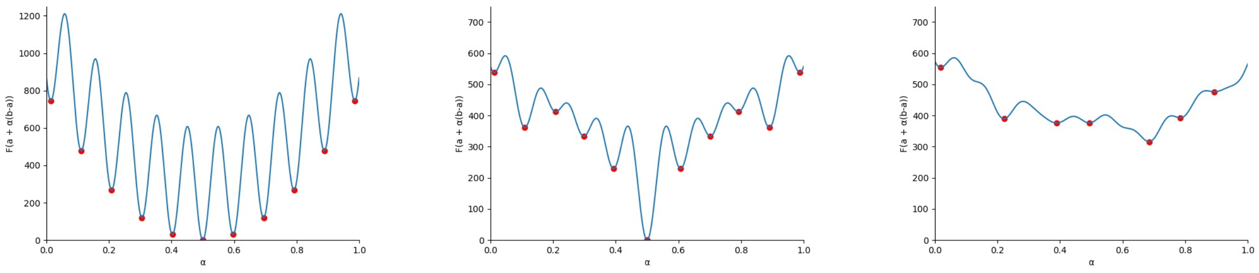
The introduction of perturbations increases both the probability of successful exploration and deceptive exploration. The effects of deceptive exploration can be observed by the upward movements of the solid lines in Fig. 3(b) which indicate that *pbest* positions have moved into worse attraction basins. Further, the swarm still stalls short of the global optimum (in $d = 30$ dimensions). Nonetheless, these promising initial results warrant a further analysis on the effects of perturbations across a broader range of functions.

## VI. Measuring the Size of Attraction Basins

The perturbation size used in the previous section was informed by the knowledge of the size of the attraction basins caused by the Rastrigin generating function. The extension of perturbations to general functions thus requires a method to measure the size of attraction basins in general search spaces. This method is limited by two considerations. First, a comprehensive analysis of the search space is akin to exhaustive search which is a computationally infeasible task. Second, the goal of the analysis is to produce results which can be used in real-time to guide the search activity of PSO, so it must use a restricted number of function evaluations.

The proposed analysis method is based on "line searches". These searches satisfy the above conditions since they use $1000d$ function evaluations (or 10% of the allotted amount). In particular, we perform $d$ line searches (one in each dimension of a $d$-dimensional search space), and each search uses 1000 function evaluations. For each line, one term traverses the full range of the search space, and all other terms traverse between a random pair of values (see Algorithm 1).

A line search along the axis of a separable function will naturally reveal the size and shape of attraction basins for



(a) main diagonal

(b) random diagonal

(c) random

Fig. 4. Examples of line searches for Rastrigin. Although random lines (c) often do not identify all optima (see (a) and (b) for examples), useful estimates on the size of attraction basins are still possible.

**Algorithm 1** General line searches (for Rastrigin)

```
for $i = 1 : d$ do
    //set $\vec{a}$
    $x_i = -5.12$
    for $j = 1 : d$ do
        if $i \neq j$ then
            $x_j = [-5.12, 5.12]$
        end if
    end for
    //set $\vec{b}$
    $x_i = 5.12$
    for $j = 1 : d$ do
        if $i \neq j$ then
            $x_j = [-5.12, 5.12]$
        end if
    end for
    for j = 0:999 do
        $\vec{x}_j = \vec{a} + j/999 \cdot (\vec{b} - \vec{a})$
        record $f(\vec{x}_j)$
    end for
end for
```

that search space. For generating functions such as Rastrigin (see Equation 1), ideal profiles are often shown as in Fig 4(a) which is a line search of the main diagonal (e.g. $\vec{a} =$ **-5.12** and $\vec{b} =$ **5.12**). Test functions are often rotated and transposed, and similar effects can be observed with a rotated and transposed line search. Fig 4(b) shows a random line through the origin (i.e. $\vec{b} = -\vec{a}$, and Fig 4(c) shows a random line (e.g. as generated by Algorithm 1).

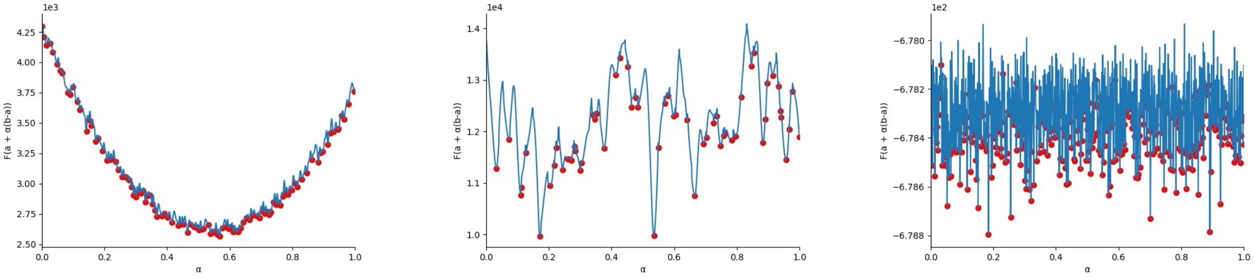
Each of the sub-plots in Fig. 4 show all 1000 points for a line search. The larger red dots indicate the observed local minima – any point which is less than both of its neighbouring two points. The main diagonal in Fig. 4(a) shows the symmetry and perfect global convexity of the Rastrigin function. In random diagonals, the correct number of local optima is usually observed, but the perfect convexity of the search space is often distorted – see the selected example in Fig. 4(b). A random line search often sees neither. The (typical) example shown in Fig. 4(c) has identified only 7 of the 11 expected attraction basins, and the (regular) shape of the search space is almost completely obscured.

Our primary objective with line searches is to determine the size of the attraction basins in the search space. For this task, the minima identified by a random line search are still useful. In particular, the smallest distance between minima in Fig. 4(c) seems to be quite close to the actual distance. This measurement has been applied to 30 independent trials each in $d = 30$ dimensions (for 900 total measurements). The known distance between local optima along an axial direction in Rastrigin is 1, and the measured distance is 0.884 with a standard deviation of 0.192. This measurement appears accurate enough for use on a broader set of benchmark functions as presented next.

## VII. Benchmark Results

We now apply PSO with *pbest* perturbations to the CEC2013 Real-Parameter Optimization Benchmark Functions [22]. The updated algorithm proceeds as follows. The first 10% of function evaluations are now used to perform the line searches. To further leverage these function evaluations, the initial swarm is seeded with the best solution from each of these $d = 30$ line searches and 20 additional uniform random solutions (as opposed to 50 uniform random solutions). During the operation of PSO, perturbations occur after 20, 30, 40, 50, 60, 70, and 80% of the allotted function evaluations. Each perturbation adds a uniform random value of $\pm 0.5$ times the measured distance between local optima in each dimension. Further, to reduce the backward fluctuations after perturbations, the *gbest* position from before the perturbations is copied to replace the worst *pbest* position produced after the perturbations.

One interim result worthy of further analysis is the performance of line searches on the benchmark functions. Focusing on the main multi-modal functions (f6-f20), there are three major patterns for the line searches as shown in Fig. 5. The first sub-plot in Fig. 5(a) shows a globally convex search space where the identified local optima reveal the key structure of the search space. The second sub-plot in Fig. 5(b) shows a deceptive search space where the identified local optima appear to show some structure within the sub-regions of the search space, but there is otherwise no information gleaned about the size and shape of these sub-regions. The third sub-plot in Fig. 5(c) shows a noisy search space where the



(a) f18

(b) f15

(c) f8

Fig. 5. Examples of line searches on the CEC2013 benchmark functions. In addition to the spacing between the local optima, insights into the structure of the search space can be gleaned. Among the multi-modal functions, three major profiles emerge: globally convex (a), deceptive (b), and noisy (c).

identified local optima indicate that there is no structure to the search space.

Table IV shows a comparison between PSO with *pbest* perturbations and PSO with random restarts. In addition to the mean error and standard deviation, the percent difference in the performance of the algorithms is also reported: $(a–b)/max(a, b)$ where $a$ is the performance of PSO with random restarts, and $b$ is the performance of PSO with *pbest* perturbations. Positive values indicate by what amount (percent) perturbations outperform restarts – negative values indicate the opposite. A standard ***t***-test is also performed to measure statistical significance. Results with statistically significant improvements ($p - value < 0.05$) of perturbations over the baseline of restarts are highlighted in bold.

The overall results are quite good. Perturbations are better than restarts on unimodal functions (1-5) because they are less disruptive to the swarm's convergent trajectory to the global optimum. Perturbations still perform worse than standard PSO (without restarts), and this makes sense under No Free Lunch [23] since perturbations have been added to achieve improvements on multi-modal functions. On the targeted multi-modal functions (6-20), perturbations achieve just over 25% average improvement compared to restarts. It is further noted that for 10 of the 15 functions in this set (in bold), the improvement is quite significant as the improvement is greater than 10% and the p-value is less than 0.05. Lastly, the composite functions (21-28) show no discernible pattern.

An observation worth highlighting is the relationship between the profile of the line search and the performance of PSO with *pbest* perturbations. The globally convex functions (e.g. f18 as shown in Fig. 5(a)) tend to have the best performance. The deceptive functions with large (partially convex) sub-structures (e.g. f15 as shown in Fig. 5(b)) lead to smaller but still meaningful performance improvements. The noisy functions (e.g. f8 as shown in Fig. 5(c)) can lead to noisy results. In general, perturbations have been designed to help PSO escape from the current local optima. Implicitly, PSO will then be able to search nearby attraction basins to find their (local) optima, and this will be most effective if this type of localized basin hopping can lead to the global optimum as it could in a globally convex search space.

TABLE IV
COMPARISON OF RESTARTS AND PERTURBATIONS IN PSO

| | 5 restarts | | Perturbations | | | |
|---|---|---|---|---|---|---|
| No. | Mean | Std Dev | Mean | Std Dev | %–diff | ***t***-test |
| 1 | 1.34e–8 | 8.69e–9 | 0.00e+0 | 0.00e+0 | **100.0%** | **0.00** |
| 2 | 5.96e+6 | 2.69e+6 | 3.35e+6 | 1.56e+6 | **43.8%** | **0.00** |
| 3 | 1.94e+8 | 1.07e+8 | 4.09e+7 | 3.06e+7 | **78.9%** | **0.00** |
| 4 | 4.08e+4 | 1.01e+4 | 2.66e+4 | 8.13e+3 | **35.0%** | **0.00** |
| 5 | 3.81e–5 | 1.48e–5 | 0.00e+0 | 0.00e+0 | **100.0%** | **0.00** |
| 1–5 | | | | | 71.5% | |
| 6 | 1.89e+1 | 6.05e+0 | 1.60e+1 | 2.17e+0 | **15.6%** | **0.02** |
| 7 | 5.28e+1 | 1.00e+1 | 6.53e+1 | 2.07e+1 | –19.1% | 0.00 |
| 8 | 2.10e+1 | 5.31e–2 | 2.09e+1 | 5.51e–2 | 0.5% | 0.00 |
| 9 | 2.79e+1 | 2.69e+0 | 2.64e+1 | 2.57e+0 | 5.3% | 0.07 |
| 10 | 2.57e+0 | 1.30e+0 | 2.68e–1 | 2.07e–1 | **89.6%** | **0.00** |
| 11 | 5.82e+1 | 1.24e+1 | 5.01e+1 | 1.34e+1 | **13.9%** | **0.04** |
| 12 | 9.66e+1 | 2.07e+1 | 7.20e+1 | 1.44e+1 | **25.5%** | **0.00** |
| 13 | 1.49e+2 | 2.36e+1 | 1.27e+2 | 2.49e+1 | **15.1%** | **0.00** |
| 14 | 2.59e+3 | 3.64e+2 | 2.34e+3 | 4.35e+2 | 9.7% | 0.01 |
| 15 | 4.98e+3 | 6.98e+2 | 3.61e+3 | 4.26e+2 | **27.5%** | **0.00** |
| 16 | 2.05e+0 | 2.59e–1 | 5.42e–1 | 1.71e–1 | **73.5%** | **0.00** |
| 17 | 1.09e+2 | 1.36e+1 | 8.86e+1 | 1.43e+1 | **18.4%** | **0.00** |
| 18 | 2.21e+2 | 2.52e+1 | 9.35e+1 | 1.22e+1 | **57.7%** | **0.00** |
| 19 | 7.23e+0 | 1.59e+0 | 4.01e+0 | 8.76e–1 | **44.5%** | **0.00** |
| 20 | 1.22e+1 | 5.11e–1 | 1.13e+1 | 3.55e–1 | 7.4% | 0.00 |
| 6–20 | | | | | 25.7% | |
| 21 | 2.20e+2 | 4.07e+1 | 2.46e+2 | 7.53e+1 | –10.7% | 0.13 |
| 22 | 2.82e+3 | 4.52e+2 | 2.80e+3 | 3.71e+2 | 0.5% | 0.89 |
| 23 | 5.03e+3 | 6.37e+2 | 4.00e+3 | 5.20e+2 | **20.5%** | **0.00** |
| 24 | 2.73e+2 | 7.62e+0 | 2.73e+2 | 7.75e+0 | 0.1% | 0.92 |
| 25 | 2.85e+2 | 7.37e+0 | 2.92e+2 | 6.52e+0 | –2.3% | 0.00 |
| 26 | 2.00e+2 | 1.54e–1 | 2.11e+2 | 4.00e+1 | –4.9% | 0.17 |
| 27 | 1.02e+3 | 6.59e+1 | 1.01e+3 | 7.23e+1 | 0.6% | 0.73 |
| 28 | 2.93e+2 | 3.58e+1 | 3.00e+2 | 3.65e–7 | –2.2% | 0.33 |
| 21–28 | | | | | 0.2% | |

## VIII. Discussion

It is important to note the difference between a swarm that has "converged" versus a swarm that has "stalled". In a converged swarm that has had all of its particles reach the same region of the search space, particle movement/speed and diversity can both approach zero. The inability of a converged swarm to perform any exploration at all makes a restart (like) disruption necessary to obtain any value from the remaining function evaluations. Conversely, a stalled swarm merely needs to have its particles (or just their *pbest* positions) escape from their current local optima.

One popular method to help particles to escape from local optima is to change the topology of the swarm (e.g. [21], [24]). A new swarm topology can allow particles to acquire new neighbours/attractors from different regions of the search space. The resulting trajectories can allow exploration of new and different attraction basins. However, implicit in the ability of topological changes (only) to alter particle trajectories is the realization that the swarm has stalled, not converged. Changing the topology of particles which have *pbest* positions near their local optima does nothing to address failed exploration which is presented as a key factor in causing a stalled swarm.

Methods to increase exploration have of course been successful in improving the performance of PSO [11]. It is noted, however, that increasing exploration and increasing the survival rates of highly promising exploratory search solutions are largely orthogonal activities. The perturbation methods introduced in this paper could be added to any PSO modification that alters particle trajectories (e.g. by changing swarm topologies). The altered search trajectories can find solutions from new and more promising regions of the search space, and the perturbed *pbest* positions can increase the chances that these solutions can survive selection and lead to successful exploration.

A separate contribution of this paper is the introduction of line searches as a method to analyze the search space. Further analysis tools are in development which can help reveal the global structure of the search space as opposed to just the local structure (e.g. ruggedness [25]). It is noted that PSO is a convergent algorithm, so it should not be expected to perform well on noisy functions (in which convergence to one good solution/local optimum does not help to find another/the next good solution). Matching a suitable metaheuristic (and/or one of its variations/modifications) to measured search space characteristics is a promising direction for further research.

## References

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

[2] F. Van den Bergh and A. P. Engelbrecht, "A study of particle swarm optimization particle trajectories," *Information sciences*, vol. 176, no. 8, pp. 937–971, 2006.

[3] 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, pp. 3038–3045.

[4] 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.

[5] P. Bosman and A. P. Engelbrecht, "Diversity rate of change measurement for particle swarm optimisers," in *International Conference on Swarm Intelligence*. Springer, 2014, pp. 86–97.

[6] 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.

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

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

[9] 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.

[10] "https://www.researchgate.net/publication/259643342_source_code_for_an_implementation_of_standard_particle_swarm_optimization_–_revised?ev=prf_pub," June 2017. [Online]. Available: code

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

[12] ——, "Analysis of stability, local convergence, and transformation sensitivity of a variant of the particle swarm optimization algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 3, pp. 370–385, 2016.

[13] J. Kennedy, "Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance," in *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, vol. 3. IEEE, 1999, pp. 1931–1938.

[14] C. Zhang and Z. Yi, "Scale-free fully informed particle swarm optimization algorithm," *Information Sciences*, vol. 181, no. 20, pp. 4550–4568, 2011.

[15] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE transactions on evolutionary computation*, vol. 10, no. 3, pp. 281–295, 2006.

[16] H. Wang, H. Sun, C. Li, S. Rahnamayan, and J.-S. Pan, "Diversity enhanced particle swarm optimization with neighborhood search," *Information Sciences*, vol. 223, pp. 119–135, 2013.

[17] D. Chen and C. Zhao, "Particle swarm optimization with adaptive population size and its application," *Applied Soft Computing*, vol. 9, no. 1, pp. 39–48, 2009.

[18] M. A. M. De Oca, T. Stutzle, K. Van den Enden, and M. Dorigo, "Incremental social learning in particle swarms," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 2, pp. 368–384, 2011.

[19] T. Hendtlass, "Wosp: a multi-optima particle swarm algorithm," in *2005 IEEE Congress on Evolutionary Computation*, vol. 1. IEEE, 2005, pp. 727–734.

[20] S. Chen, "Locust swarms-a new multi-optima search technique," in *2009 IEEE Congress on Evolutionary Computation*. IEEE, 2009, pp. 1745–1752.

[21] J.-J. Liang and P. N. Suganthan, "Dynamic multi-swarm particle swarm optimizer," in *Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005*. IEEE, 2005, pp. 124–129.

[22] J. Liang, B. Qu, P. Suganthan, and A. G. Hernández-Díaz, "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, pp. 3–18, 2013.

[23] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE transactions on evolutionary computation*, vol. 1, no. 1, pp. 67–82, 1997.

[24] Y.-X. Wang and Q.-L. Xiang, "Particle swarms with dynamic ring topology," in *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*. IEEE, 2008, pp. 419–423.

[25] K. M. Malan and A. P. Engelbrecht, "Fitness landscape analysis for metaheuristic performance prediction," in *Recent advances in the theory and application of fitness landscapes*. Springer, 2014, pp. 103–132.