Stall Detection in Particle Swarm Optimization

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

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

***Abstract*—Particle Swarm Optimization can experience stall in multi-modal search spaces. A stalled swarm is unable to converge and unable to find better solutions because all of its exploratory search solutions are rejected. Remedies to address the stall condition can benefit from knowing which particles are performing exploration and which are performing exploitation, so we develop an efficient and accurate real-time, search space independent method to perform this identification.**

***Index Terms*—exploration, selection, convergence, particle swarm optimization**

I. Introduction

Many population-based metaheuristics have convergent search trajectories which lead all of the population members towards a single (ideally global) optimum. The term "premature convergence" is often applied when a metaheuristic fails to reach the global optimum and there is no more improvement in performance. In Particle Swarm Optimization (PSO) [1], true convergence would mean that all of the particles have the same location and zero velocities. An alternative situation in multi-modal search spaces that can cause PSO to be unable to achieve improving performance (and to be unable to converge) is the "stall" condition [2].

The initial observation and definition of the stall condition is predicated on a ring topology which is a part of standard PSO [3]. Within the ring, a certain number of particles can come to have a highly fit *pbest* position in one region of the search space and a highly fit *lbest* position in another region of the search space. The distance between these *pbest* and *lbest* positions can cause an affected particle to have large velocities during its oscillatory trajectory between these two points of attraction [4]. Since slower speed particles are required for effective exploitation [5], it is possible for a high speed particle to never sample a position (sufficiently close to a local optimum) which would be fit enough to survive selection [2], [6].

Stall, much like premature convergence, leads to wasted computational effort. However, unlike a converged swarm, a stalled swarm has its *pbest* positions in many different and highly promising regions of the search space. These multiple local optima could be useful in globally convex search spaces to target the center of the best local optima where the global optimum should reside [7]. A key component to leveraging the local optima that have been found by the swarm is to be able to identify the specific particles which (likely) represent these local optima.

We propose a computationally efficient, search space independent method to identify a stalled swarm. It is shown that stalled swarms have a characteristic pattern of "clusters" of particles in the ring topology. Within each cluster, the particles tend to perform exploitation around the same local optimum. At the edge between neighbouring clusters on the ring topology, there is one particle with its *pbest* position in one cluster and its *lbest* position in the other cluster, and this particle performs a large amount of exploration. A simple velocity measurement allows us to identify these particles, and thus the clusters of exploiting particles that they separate.

An initial set of experiments depends on the specific definitions of exploration and exploitation that were used in the initial identification of the stall condition [2], and this background information is presented in Section II. Extensive coverage of the stall condition with useful new insights is provided in Section III. Section IV covers the development of the search-space independent, computationally efficient method to detect particle clusters. Section V shows how the new method works on several interesting search spaces before the paper finishes with a Discussion.

II. Background

A broad survey of over 100 papers led Crepinšek, Liu, and Mernik to the unexpected conclusion that "The fact that until now exploration and exploitation have only been implicitly defined in EAs comes as a big surprise." [8] Similarly imprecise terms are common in the literature for PSO. For example, "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." [9] It is noted that this diversity-based description of exploration is unable to determine for a specific particle at a given iteration whether it is performing exploration or exploitation.

Explicit definitions for exploration and exploitation are possible with precise definitions of attraction basins. An attraction basin is defined as a region of a search space in which monotonic paths (e.g. of decreasing fitness for a minimization objective) from all of its points will lead to the same local optimum. Based on this definition, a multi-modal

This research has received funding support from the Faculty of Liberal Arts & Professional Studies at York University through an NSERC Bridging Grant.

search space contains a countable number of attraction basins, and a unimodal search space will have exactly one attraction basin. It is also useful to define the fitness of an attraction basin to be the fitness of its local optimum. [6]

This definition for attraction basins can now be leveraged to define "exploration" as a search point (e.g. the current position of a particle) that is in a different attraction basin than its reference solution(s) (e.g. a particle's *pbest* position), and "exploitation" occurs when the search point is in the same attraction basin as (one of) its reference solution(s). During exploration, the comparison of the current position with its *pbest* position can result in four different outcomes, and this shows the importance of selection on the operation and performance of PSO.

The first possible outcome is "Successful Exploration": a current position from a fitter attraction basin is compared against its less fit *pbest* position from a less fit attraction basin. This comparison results in acceptance of the current solution as the new *pbest* position. The second possible outcome is "Successful Rejection": a less fit current position from a less fit attraction basin is rejected when it is compared against its fitter *pbest* position that is in a fitter attraction basin. The third possible outcome is "Deceptive Exploration": a current position from a less fit attraction basin is accepted because it is fitter than its *pbest* position that was from a fitter attraction basin. The fourth possible outcome is "Failed Exploration": a current position from a fitter attraction basin is rejected because it is less fit than its *pbest* position which is in a less fit attraction basin. [6]

The ability to determine these categories depends on knowing the boundaries for all of the attraction basins in the search space. This is possible for the Rastrigin function (see Equation 1) which has the useful property that every point with integer values in all dimensions is a local optimum, and all other points in the search space belong to the attraction basin of the local optimum that is determined by rounding each solution term to its nearest integer value [10]. It is accepted that definitions for exploration and exploitation which are limited to specific artificial search spaces are highly limited, so a key objective of the following research is to present tools to identify exploration and exploitation that can be used more broadly.

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

III. Characteristics of Stall in PSO

Our experiments are based on a version of standard Particle Swarm Optimization [3] in which a ring topology is used. The key parameters specified from this standardization are χ = 0.72984, and c_1 = c_2 = 2.05 for the velocity updates given in Equation 2. Additional implementation details are the use of n = 50 particles [3], zero initial velocities [11], 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) [12]. The source code for this implementation is available online [13].

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

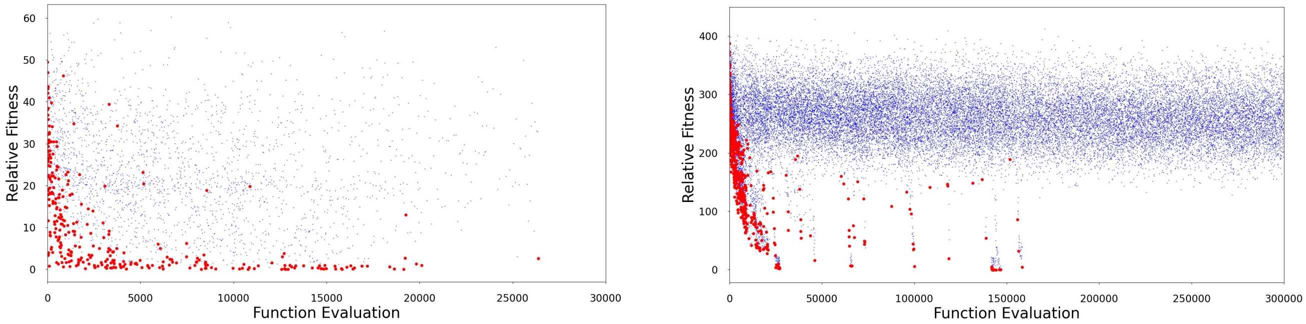
The initial positions of the swarm are uniform random across the full range of the search space. For the Rastrigin function, these initial positions will have a fitness difference with their local optima that is on average half the maximum possible fitness difference within an attraction basin (which is 60 in $d = 3$ dimensions and 600 in $d = 30$ dimensions). A reference solution with a large fitness difference from its local optimum (i.e. a poor "relative fitness") can lead to high rates of acceptance of exploratory search solutions, while a reference solution that is a local optimum can lead to near 100% rates of rejection [10]. In PSO, *pbest* positions start with poor relative fitness and rapidly approach local optima (which have zero relative fitness), and rejection rates increase in a corresponding manner [2], [6].

Fig. 1 shows the effect of relative fitness on selection for PSO on Rastrigin in $d = 3$ and $d = 30$ dimensions. The dots indicate the relative fitness of every exploratory search solution that is in an attraction basin that is fitter than its *pbest* and attraction basin. The blue dots represent rejected solutions (i.e. Failed Exploration), and the red dots represent accepted solutions (i.e. Successful Exploration). It can be seen that the red dots have a large range of relative fitness during the first few iterations, but soon become only the most relatively fit of the dots as PSO progresses.

A *pbest* reference solution that has become a local optimum will often require an exploratory search solution to have exceptional relative fitness to survive selection and achieve Successful Exploration. In lower dimensions (e.g. $d = 3$ as shown in Fig. 1(a)), this happens frequently enough for the swarm to achieve convergence – after the last red dot, there are no more dots representing exploratory solutions because all of the particles have reached the same attraction basin. In higher dimensions (e.g. $d = 30$ as shown in Fig. 1(b)), particles which reach one local optimum cannot make the jump into the neighbouring particle's attraction basin, so the swarm stalls – the band of only blue dots after approximately 50% of the allotted function evaluations indicates a stable pattern filled with Failed Exploration.

Fig. 2 shows another way to observe the difference between convergence and stall in PSO. In these plots, the classification of every particle at each iteration of PSO is shown for the allotted function evaluations – this leads to 600 iterations in $d = 3$ dimensions and 6000 iterations in $d = 30$ dimensions. The particles are indexed from 0-49 along the x-axis, and they are labelled for each iteration as being one of exploitation (grey), Successful Exploration (red), Successful Rejection (blue), Deceptive Exploration (yellow), or Failed Exploration (black).

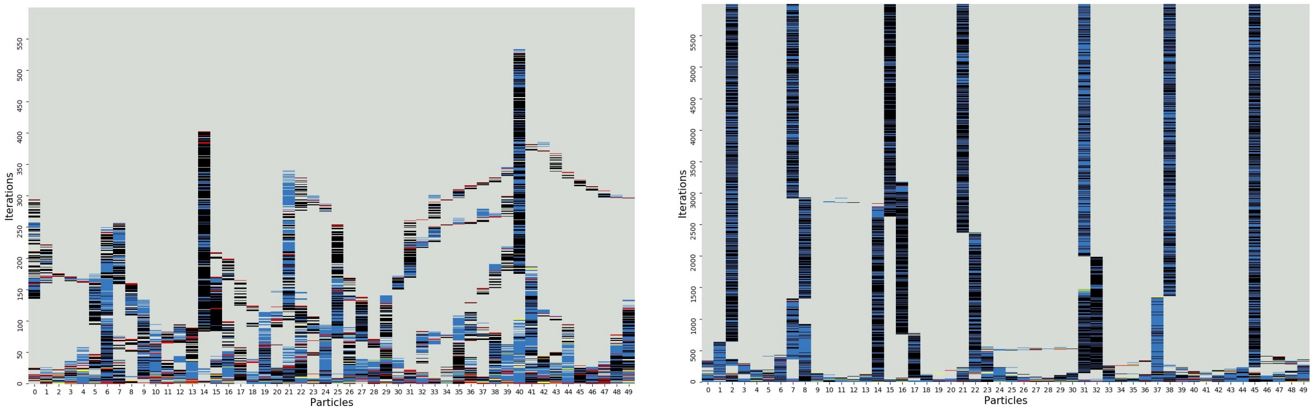
The expectation for PSO with random initial positions is that there is an initial phase of exploration. However, as can be seen



(a) $d = 3$

(b) $d = 30$

Fig. 1. The effect of relative fitness on selection. Successful Exploration (red dots) occurs frequently for solutions with average relative fitness in the beginning, but soon only exploratory solutions with exceptional relative fitness can survive selection. The blue band of dots representing Failed Exploration in plot (b) for $d = 30$ dimensions is a characteristic of a stalled swarm.



(a) $d = 3$

(b) $d = 30$

Fig. 2. PSO converges in $d = 3$ dimensions in (a) and stalls in $d = 30$ dimensions in (b). A converged swarm performs no more exploration as indicated by the full rows of grey. A signature feature of a stalled swarm is the formation of clusters – each (grey) cluster of particles performs exploitation in one attraction basin and the clusters are separated by a particle performing exploration that is rejected by selection (blue and black).

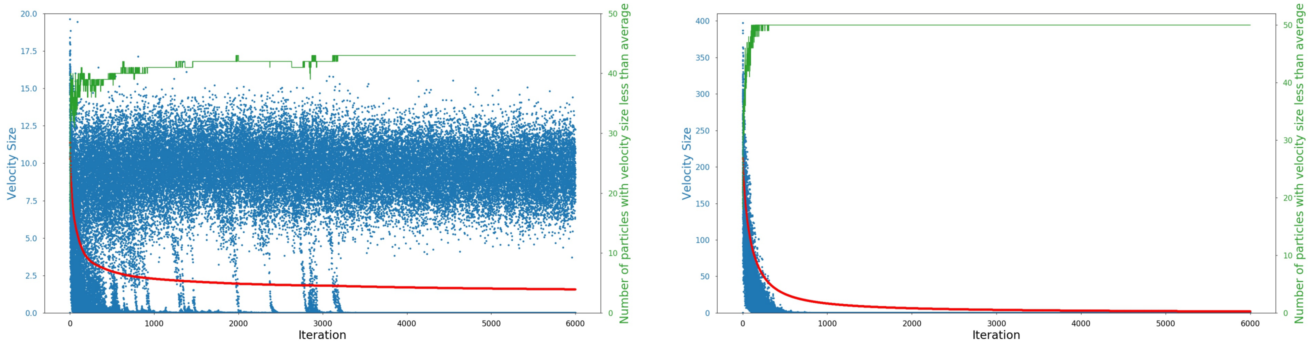
in Fig. 2, this phase is very short as clusters (of exploitation shown in grey) can appear after as few as 5% of the allowed iterations. In $d = 3$ dimensions, "staircase" patterns are visible in which clusters can grow bigger by consuming neighbouring particles. In $d = 30$ dimensions, the clusters can become very stable. These stable clusters show how a stalled swarm can have particles in many diverse regions of the search space and still be unable to benefit from the exploration that is performed.

The dark vertical lines in Fig. 2 are almost entirely due to the blue and black colour labels representing rejection. The existence of exploration is simple evidence that the swarm has not converged, but other interesting insights are also possible. For example, when particle 37 stops exploration (and begins exploitation), there is a coincidental transition from exploitation to exploration in particle 38. This observation supports our claim that clusters of exploiting particles are separated by an exploring particle that has its *pbest* position in one cluster/attraction basin and its *lbest* position in the other. These small changes at the boundaries of the clusters also show that the core of the clusters can become highly stable.

IV. Simple Cluster Identification in PSO

The stall pattern shown in Fig. 2 has some particles in clusters that are performing exploitation and some particles between the clusters that are performing exploration. The particles that perform only exploitation clearly have velocities which never lead them to escape the confines of their current attraction basin – i.e. they are small in magnitude. The particles which perform exploration instead require velocities with sufficiently large magnitudes to allow them to visit different attraction basins. We believe we can thus identify exploring and exploiting particles based on their speed.

We measure the speed of every particle for our implementation of standard PSO when applied to the Rastrigin function in $d = 30$ dimensions. Fig. 3(a) shows the speed of each particle (blue dots), the cumulative average (mean) speed of all particles (red dots), and the total number of particles in each iteration with a speed below the cumulative average (green line). The cumulative average speed is used because the average speed cannot differentiate between stall and convergence. For example, Fig. 3(b) shows the same data for Sphere in $d = 30$ dimensions, and all of the particles rapidly slow below the cumulative average speed while the



(a) Rastrigin Function

(b) Sphere Function

Fig. 3. Blue dots represent particle speed ($n = 50$ per iteration), red dots indicate cumulative average speed (one per iteration), and the green line shows the number of particles in each iteration that have a slower speed than the cumulative average.

average speed at a given iteration would obviously require some particles to have greater speeds. A converged swarm still has one cluster, but it is otherwise noted that the number of particles with above average speeds will match the number of clusters in the swarm. This leads to the following speed-based method for cluster identification.

We first label the particles with below average speed to be exploitative and particles with above average speed to be explorative. The higher speed explorative particles break the ring topology into clusters of particles which each perform exploitation around the same optimum. The identification of explorative and exploitative particles and the clusters of exploitative particles can both be determined in linear time (with respect to the number of particles n). The accuracy of the new speed-based method is tested on Rastrigin which has known and easily measurable attraction basins [10], and the results are shown in a Table I. We show the accuracy at several specific iterations (e.g. there is 100% accuracy at iteration 600 which is 10% of the allotted function evaluations), and also the overall accuracy averaged for all iterations from 600 to 6000.

The data presented in Fig. 3 provides another observable difference between convergence and stall. The convergence condition shown in Fig. 3(b) for Sphere has all of the particle speeds drop below the cumulative average after about 10% of the iterations. The zero velocities indicate that all of the particles have reached the same location, and these are the two conditions which define convergence. The stall condition shown in Fig. 3(a) for Rastrigin also enters a stable pattern after about 50% of the iterations. There are 43 particles with near zero velocity that are performing exploitation and seven particles with large velocities that are performing exploration. The lack of particle speeds between these two distinct sets (as seen sporadically between iterations 1000 and 3000) implies that there is no movement in *pbest* positions that could cause previously (near) stationary particles to begin moving.

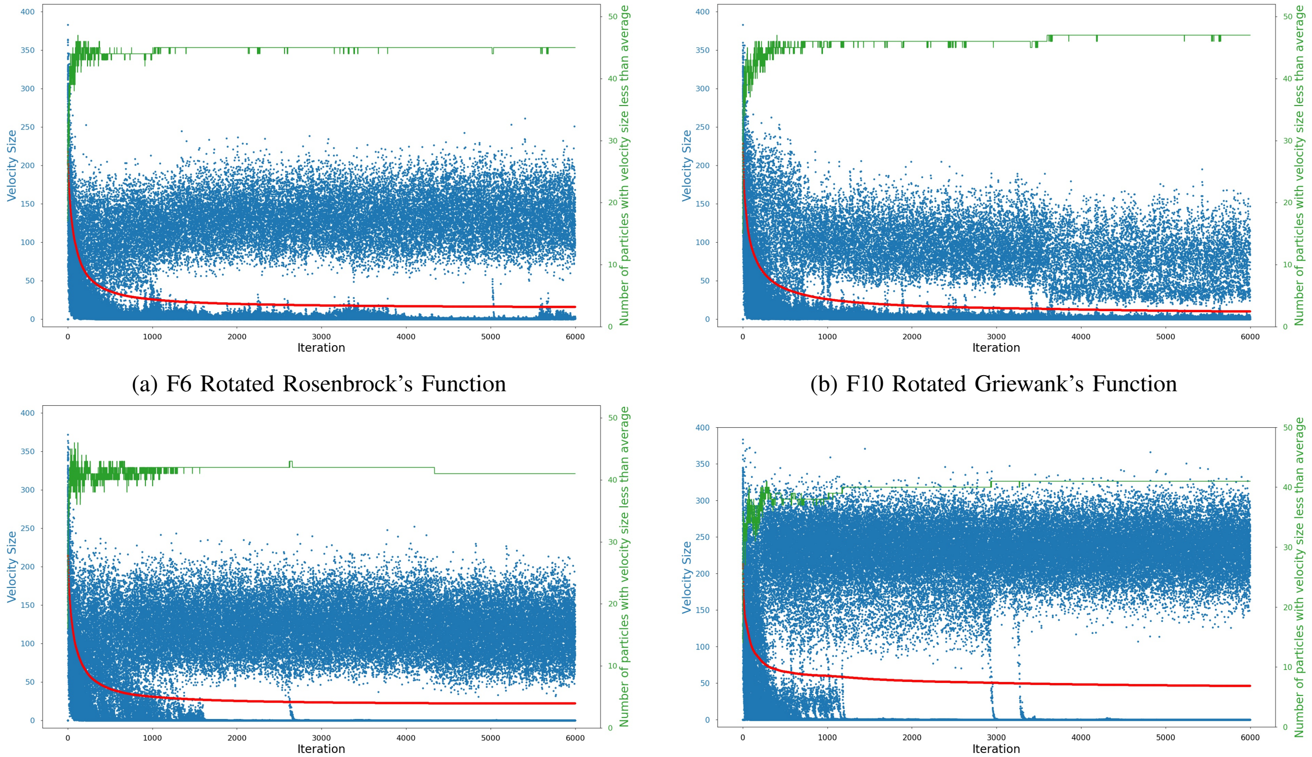
TABLE I
ACCURACY OF SPEED-BASED PARTICLE IDENTIFICATION FOR EXPLORATION AND EXPLOITATION

Iterations	Accuracy
600	100 %
1200	100 %
1800	100 %
2400	100 %
3000	100 %
3600	100 %
4200	100 %
4800	100 %
5400	100 %
6000	100 %
Overall	99.7%

V. Other Search Spaces

The speed-based method for stall detection has been applied to the CEC2013 benchmark functions [14]. Several notable samples are shown in Fig. 4 and Fig. 5. First, in Fig. 4, stall patterns that are highly similar to the one shown for Rastrigin in Fig. 3(a) are presented. We believe that this demonstrates that the speed-based method has the ability to identify explorative and exploitative particles in PSO when meaningful clusters have formed.

However, in certain "noisy" search spaces with deep and narrow attraction basins, clusters do not form. The speed-based method can show an identifiable alternate pattern for these search spaces. Two key characteristics are the lack of a gap of slow speeds and high speeds (visible but difficult to measure) and fewer than 35 (out of $n = 50$) particles with below average speed (measurable). We hypothesize that the lack of clusters leads to mostly high-speed particles which perform very little exploitation. A potential measurement to support this hypothesis is that PSO performs an average of 11.0 *pbest* updates per iteration on F11 and only 0.2 *pbest* updates per iteration on F16.



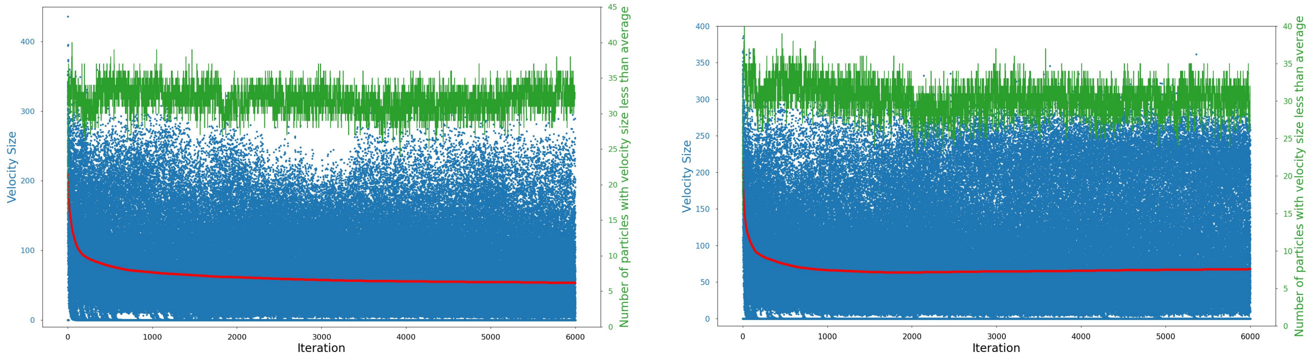
(a) F6 Rotated Rosenbrock's Function

(b) F10 Rotated Griewank's Function

(c) F11 Rastrigin's Function

(d) F14 Schwefel's Function

Fig. 4. Speed-based cluster identification on sample CEC2013 functions. Blue dots represent particle speed ($n = 50$ per iteration), red dots indicate cumulative average speed (one per iteration), and the green line shows the number of particles in each iteration that have a slower speed than the cumulative average. The clear gaps around the red dots indicate the formation of large and stable clusters.



(a) F8 Rotated Ackley's Function

(b) F16 Rotated Katsuura Function

Fig. 5. Speed-based cluster identification on sample CEC2013 functions. Blue dots represent particle speed ($n = 50$ per iteration), red dots indicate cumulative average speed (one per iteration), and the green line shows the number of particles in each iteration that have a slower speed than the cumulative average. The lack of clear gaps around the red dots and the small number of particles with below average speed indicate that cluster formation has not occurred.

VI. Discussion

The stall condition provides useful insight into a potential mode of failure for PSO in multi-modal search spaces [2]. However, the initial identification of stall depended on precise definitions for attraction basins, exploration, and exploitation [6] that made it impossible to observe outside of simple, artificial search spaces such as Rastrigin. It was specifically unclear if the stall condition was present in PSO for other search spaces. The new speed-based methods to observe stall make it clear that stall occurs in PSO across a broad range of search spaces and that its existence can be easily detected.

The identification of a stall in PSO can lead to modifications to address it (e.g. [2], and the identification of which particles are performing exploration and exploitation in stall can lead to more targeted (e.g. heterogeneous) modifications. It is further noted that different types of stall patterns can be identified for different types of search spaces. This insight can lead to modifications that are again more targeted to address specific search space properties. The development of these PSO modifications to address stall is a promising area for future 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] S. Chen, I. Abdulselam, N. Yadollahpour, and Y. Gonzalez-Fernandez, “Particle swarm optimization with pbest perturbations,” in *2020 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2020, pp. 1–8.

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

[4] 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.

[5] 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.

[6] 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.

[7] K. D. Boese, “Models for iterative global optimization.” 1997.

[8] 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.

[9] 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.

[10] 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.

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

[12] 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.

[13] “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

[14] 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.