Metadata of the chapter that will be visualized in SpringerLink

Book Title	Swarm Intelligence
Series Title	
Chapter Title	Stability-Guided Particle Swarm Optimization
Copyright Year	2022
Copyright HolderName	Springer Nature Switzerland AG

Corresponding Author	Family Name	**Engelbrecht**
	Particle	
	Given Name	**Andries**
	Prefix	
	Suffix	
	Role	
	Division	Department of Industrial Engineering, and Computer Science Division
	Organization	Stellenbosch University
	Address	Stellenbosch, South Africa
	Email	engel@sun.ac.za
	ORCID	http://orcid.org/0000-0002-0242-3539

Abstract	Particle swarm optimization (PSO) performance has been shown to be sensitive to control parameter values. To obtain best possible results, control parameter tuning or self-adaptive PSO implementations are necessary. Theoretical stability analyses have produced stability conditions on the PSO control parameters to guarantee that an equilibrium state is reached. Should control parameter values be chosen to satisfy a stability condition, divergent and cyclic search behaviour is prevented, and particles are guaranteed to stop moving. This paper proposes that control parameter values be randomly sampled to satisfy a given stability condition, removing the need for control parameter tuning. Empirical results show that the resulting stability-guided PSO performs competitively to a PSO with tuned control parameter values.

Stability-Guided Particle Swarm Optimization

Andries Engelbrecht(✉)

Department of Industrial Engineering, and Computer Science Division, Stellenbosch University, Stellenbosch, South Africa
engel@sun.ac.za

Abstract. Particle swarm optimization (PSO) performance has been shown to be sensitive to control parameter values. To obtain best possible results, control parameter tuning or self-adaptive PSO implementations are necessary. Theoretical stability analyses have produced stability conditions on the PSO control parameters to guarantee that an equilibrium state is reached. Should control parameter values be chosen to satisfy a stability condition, divergent and cyclic search behaviour is prevented, and particles are guaranteed to stop moving. This paper proposes that control parameter values be randomly sampled to satisfy a given stability condition, removing the need for control parameter tuning. Empirical results show that the resulting stability-guided PSO performs competitively to a PSO with tuned control parameter values.

1 Introduction

The performance of particle swarm optimization (PSO) [28] algorithms is sensitive to control parameter values [4,6,7,38]. Literature has suggested various control parameter configurations that result in good PSO performance [32]. However, for best performance, the PSO control parameters require tuning for each problem [27,38]. Various tuning approaches are available [3–5,13,29,37]. These approaches can be computationally expensive. Tuning of PSO control parameters prior to solving an optimization problem has recently been shown to not necessarily result in best performance due to the time-dependence of control parameter optimality [20]. An alternative to control parameter tuning is to deterministically adjust or self-adapted control parameter values during the search process. Though, recent studies have shown these dynamic and self-adaptive approaches are mostly inefficient [16,19,21].

Theoretical analyses of PSO algorithms provided a good understanding of PSO behavior [6,8,11,33,34,38], specifically the impact of control parameters [10,20,22–25]. These theoretical studies provided stability conditions derived on the values of the PSO control parameters, giving guarantees under which conditions particle swarms will reach an equilibrium state. An important outcome of these stability conditions is formal guidance on the setting of PSO control parameter values. This paper proposes that control parameter values be

M. Dorigo et al. (Eds.): ANTS 2022, LNCS 13491, pp. 1–10, 2022.
https://doi.org/10.1007/978-3-031-20176-9_33

randomly selected to satisfy a provided stability condition. The resulting PSO algorithms are referred to in this paper as stability-guided PSO algorithms. Results show that these algorithms are very competitive to that of a PSO algorithm with tuned control parameters.

Section 2 discusses the PSO algorithm on which this work is based. Stability conditions are reviewed in Sect. 3. The stability-guided PSO algorithms are summarized in Sect. 4. The empirical process is provided in Sect. 5, and the results are presented and discussed in Sect. 6.

2 Particle Swarm Optimization

The first PSO algorithm was proposed by Kennedy and Eberhart [28]. While various variations of the PSO algorithm have been developed, the focus of this paper is on the inertia weight PSO developed by Shi and Eberhart [36]. For the inertia weight PSO, particle positions are updated using

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \tag{1}$$

The velocity is calculated using

$$\mathbf{v}_i(t+1) = w\mathbf{v}_i(t) + c_1\mathbf{r}_{1,i}(t) \odot (\mathbf{y}_i(t) - \mathbf{x}_i(t)) + c_2\mathbf{r}_{2,i}(t) \odot (\hat{\mathbf{y}}_i(t) - \mathbf{x}_i(t)) \tag{2}$$

where $\odot$ is the Hadamard product, $w \in [0,1]$ is the inertia weight, c_1 and c_2 are acceleration coefficients, $\mathbf{r}_{1,i}(t)$ and $\mathbf{r}_{2,i}(t)$ are vectors of random values sampled from an n_x-dimensional uniform distribution over $[0,1]$, n_x is the number of decision variables, $\mathbf{y}_i(t)$ and $\hat{\mathbf{y}}_i(t)$ are the personal and global best positions.

3 Stability Conditions

This paper focuses only on the following two stability conditions: Referred to as SC1, Van den Bergh and Engelbrecht [38] derived under both the deterministic and stagnation assumption [8,9] that

$$c_1 + c_2 < 2(1+w), \quad c_1 > 0, \quad c_2 > 0, \quad 0 < w < 1 \tag{3}$$

The same stability condition was derived by Cleghorn and Engelbrecht [8] under the deterministic and weak chaotic assumption [8,9]. Referred to as SC2, Poli and Broomhead [33,34] derived under the stagnation assumption that

$$c_1 + c2 = \frac{24(1-w^2)}{7-5w}, \quad -1 < w < 1 \tag{4}$$

4 Stability-Guided Particle Swarm Optimization

The stability-guided PSO algorithms presented in this paper are not the first approaches towards random selection of control parameters to satisfy stability conditions. Erwin and Engelbrecht [17] presented a similar approach for a multi-objective PSO algorithm, i.e. the multi-guide PSO (MGPSO) [35]. It was shown that the stability-guided MGPSO performed on par with a tuned MGPSO.

The stability-guided PSO algorithms are variations of the global best inertia weight PSO algorithm, with control parameter values randomly sampled to satisfy a given stability condition. The first variation samples the control parameters to satisfy the stability condition of Eq. (3). The second variation samples the control parameter values to satisfy the stability condition of Eq. (4). Note that though the stability condition by Poli and Broomhead allows for $w \in [-1, 1]$, the sampling algorithm considers only values of $w \in [0, 1]$.

The control parameter sampling is done before the particle velocity update. Each particle samples its own values for the control parameters.

5 Empirical Process

This section outlines the empirical process. Section 5.1 summarizes the benchmark functions used, Sect. 5.2 lists the algorithms used and discusses the tuning process, and Sect. 5.3 provides the performance measures and the statistical tests used.

Table 1. List of benchmark functions (* indicates that the function was generalized for $n_x \geq 1$)

Function	Domain	Reference	Function	Domain	Reference
Ackley 1	[−32, 32]	[26]	Paviani	[2.001, 9.999]	[26]
Alpine 1	[−10, 10]	[26]	Penalty 1	[−50, 50]	[18]
Bohachevsky 1*	[−15, 15]	[26]	Penalty 2	[−50, 50]	[18]
BMF	[−5, 5]		Pinter 2	[−10, 10]	[26]
Brown	[−1, 4]	[26]	Price 2*	[−10, 10]	[26]
CosineMixture	[−1, 1]	[26]	Qings	[−500, 500]	[26]
CrossInTray	[−10, 10]	[26]	Quadric	[−100, 100]	[15]
Discus	[−100, 100]	[30]	Rana	[−500, 500]	[26]
DropWave*	[−5.12, 5.12]	[2]	Rastrigin	[−5.12, 5.12]	[1]
Easom*	[−100, 100]	[26]	Riple 25*	[0, 1]	[26]
Elliptic	[−100, 100]	[15]	Rosenbrock	[−30, 30]	[26]
EggCrate*	[−5, 5]	[26]	Salomon	[−100, 100]	[26]
EggHolder	[−512, 512]	[26]	Schwefel 1	[−100, 100]	[26]
Exponential	[−1, 1]	[26]	Schwefel 2.26	[−500, 500]	[26]
Giunta*	[−1, 1]	[26]	Shubert 4	[−10, 10]	[26]
Levy 3	[−10, 10]	[1]	Step 3	[−100, 100]	[26]
LevyMontalvo	[−5, 5]	[18]	Trigonometric	[−10, 10]	[26]
Mishra 1	[0, 1]	[26]	Vincent	[0.25, 10]	[15]
Mishra 4	[−10, 10]	[26]	Weierstrass	[−0.5, 0.5]	[26]
Mishra 7	[−10, 10]	[26]	XinSheYang 1	[−5, 5]	[26]
NeedleEye	[−10, 10]	[2]	XinSheYang 3	[−20, 20]	[26]
Norwegian	[−5, 5]	[15]	XinSheYang 4	[−10, 10]	[26]

5.1 Benchmark Functions

Each of the algorithms have been evaluated on 30-dimensional instances of the 44 benchmark functions listed in Table 1. Note that the LevyMontalvo function is a generalization of the Levy 13 function [18]. The BonyadiMichalewicz function (BMF) is defined as $f(\mathbf{x}) = \frac{\prod_{j=1}^{n_x}(x_j+1)}{\prod_{j=1}^{n_x}((x_j+1)^2+1)}$.

5.2 Algorithms

The stability-guided PSO algorithms are compared with two versions of the standard inertia weight PSO algorithm: (1) PSO$_s$, where the control parameters are static and set to $w = 0.7, c_1 = 1.4$ and $c_2 = 1.4$ [12,14]; (2) PSO$_t$, with tuned control parameters using the grid search process outlined in Algorithm 1.1. The found best control parameter configurations are listed in Table 2. These values serve as additional confirmation of the strong dependence of the control parameter values on achieving best results.

Algorithm 1.1. Control Parameter Tuning Process

```
for w = 0 to w = 1 in increments of 0.05 do
    for c1 = 0 to c1 = 2 * w - 2 in increments of 0.2 do
        for c2 = 2 - c1 to c2 = 2 * w - 2 - c1 in increments of 0.2 do
            Execute PSO(w, c1, c2) for 10 independent runs
        end for
    end for
end for
return Control parameter configuration that resulted in best average solution
```

The two stability-guided PSO algorithms are respectively referred to as PSO$_{sc1}$ and PSO$_{sc2}$.

The swarm size for each algorithm was set to 30 particles, and each algorithm was executed on each problem for 1000 iterations.

In order to show proof of concept, it is sufficient to compare the stability-guided PSO algorithms with a tuned PSO. It should be noted that recent analyses of dynamic and self-adaptive approaches to PSO control parameter setting have shown that existing approaches do not perform well [19,21]. An approach to random sampling of the inertia weight was shown to provide the best performance, though not statistically significantly better than the PSO$_s$ approach [19]. All other approaches were shown not to perform significantly better than statically assigned control parameter values as for PSO$_s$.

5.3 Performance Measure and Statistical Tests

The quality of the global best position after 1000 iterations, averaged over 30 independent runs, was used to rank the algorithms per benchmark problem based

Table 2. Tuned control parameter values

Function	w, c_1, c_2	Function	w, c_1, c_2	Function	w, c_1, c_2
Ackley 1	0.6, 2.0, 1.8	Levy 3	0.8, 1.8, 1.0	Rastrigin	0.85, 1.4, 0.4
Alpine 1	0.65, 1.6, 1.8	LevyMontalvo	0.8, 2.0, 0.8	Ripple 25	0.8, 0.8, 1.8
Bohachevsky 1	0.75, 1.2, 2.0	Mishra 1	0.75, 2.0, 1.0	Rosenbrock	0.85, 1.2, 1.0
BMF	0.55, 1.8, 1.8	Mishra 4	0.85, 1.2, 0.8	Salomon	0.75, 1.6, 1.4
Brown	0.6, 1.8, 1.6	Mishra 7	0.8, 0.2, 1.6	Schwefel 1	0.5, 2.0, 1.8
CosineMixture	0.8, 0.2, 1.8	NeedleEye	1.0, 0.2, 0.2	Schwefel 2.26	0.8, 0.2, 2.0
CrossInTray	0.75, 2.0, 0.4	Norwegian	0.65, 0.6, 2.0	Shubert 4	0.8, 0.2, 2.0
Discus	0.75, 2.0, 1.0	Paviani	0.6, 1.4, 1.8	Step 3	0.15, 2.0, 2.0
DropWave	0.8, 2.0, 0.6	Penalty 1	0.4, 2.0, 2.0	Trigonometric	0.8, 1.8, 0.6
Easom	0.8, 1.0, 0.4	Penalty 2	0.7, 1.8, 1.5	Vincent	0.55, 1.8, 1.8
Elliptic	0.5, 1.6, 2.0	Pinter 2	0.8, 2.0, 0.8	Weierstrass	0.9, 2.0, 1.0
EggCrate	0.85, 1.8, 0.2	Price 2	0.65, 1.2, 1.8	XinSheYang 1	0.85, 1.6, 0.6
EggHolder	0.8, 0.2, 2.0	Qings	0.5, 1.6, 2.0	XinSheYang 3	0.75, 0.2, 2.0
Exponential	0.45, 1.8, 2.0	Quadric	0.6, 1.4, 1.8	XinSheYang 4	0.6, 1.0, 2.0
Giunta	0.8, 2.0, 0.8	Rana	0.8, 0.2, 2.0		

on a wins-losses approach. For each pair of algorithms, a Mann-Whitney U test (at confidence level of 0.05) was applied to determine if there is a statistical significant difference in performance. If so, the winning algorithm is scored a win and the loosing algorithm a loss. The ranking is done on the differences between the wins and losses, with a lower rank indicating better performance. To determine the extend to which one algorithm is better than another, the ratio A_1/A_2 is reported for each function, where A_1 and A_2 refer to two different algorithms. A ratio close to one indicates similar performance. A ratio greater than one indicates the extend to which algorithm A_1 is worse than algorithm A_2. A ratio less than one indicates the extent to which algorithm A_1 is better than algorithm A_2.

6 Results

Table 3 summarizes the ranks per function as well as the average rank over all of the functions. The first observation from Table 3 is that the static approach, PSO$_s$, ranked the worst, with only one function for which it ranked best (i.e. NeedleEye), though together with the other algorithms. PSO$_t$ ranked on average the best over all of the problems. The ranks for PSO$_{sc1}$ and PSO$_{sc2}$ are very close, and close to that of the tuned PSO$_t$. For nine of the problems (i.e. BonyadiMichalewicz, Exponential, NeedleEye, Price 2, Qings, Rosenbrock, Salomon, Step 3, and Vincent) there is no significant difference between the stability-guided PSOs and the tuned PSO. For the rest of the problems, both stability-guided PSOs ranked best for two problems (i.e. CosineMixture and XinSheYang 4); PSO$_t$ and PSO$_{sc1}$ ranked both best for one problem (i.e. Mishra 7); PSO$_t$ and PSO$_{sc2}$ ranked both the best for one problem (i.e. Schwefel 2.26).

Table 3. Ranks based on solution quality and performance ratios

Function	PSO$_s$	PSO$_t$	PSO$_{sc1}$	PSO$_{sc2}$	$\frac{PSO_{sc1}}{PSO_t}$	$\frac{PSO_{sc2}}{PSO_t}$	$\frac{PSO_{sc2}}{PSO_{sc1}}$
Ackley 1	3	1	2	2	4.87E+00	4.78E+00	9.80E−01
Alpine 1	4	1	2	3	5.98E−01	4.89E+01	8.18E+01
Bohachevsky 1	3	1	2	2	2.32E+00	2.59E+00	1.12E+00
BonyadiMichalewicz	2	1	1	1	1.00E+00	1.00E+00	1.00E+00
Brown	4	3	2	1	1.10E−48	3.83E−48	3.47E+00
CosineMixture	3	2	1	1	1.04E+00	1.04E+00	1.00E+00
CrossInTray	4	1	3	2	7.60E−01	8.81E−01	1.16E+00
Discus	4	1	3	2	9.27E−48	7.41E−49	8.00E−02
DropWave	4	1	3	2	6.71E−01	7.61E−01	1.13E+00
Easom	3	1	3	2	1.02E+00	1.01E+00	9.97E−01
Elliptic	4	1	3	2	3.97E−27	1.02E−23	2.57E+03
EggCrate	4	1	3	2	4.97E+00	4.11E+00	8.26E−01
EggHolder	3	1	2	2	9.05E−01	9.51E−01	1.05E+00
Exponential	2	1	1	1	1.00E+00	1.00E+00	1.00E+00
Giunta	3	1	2	2	9.88E−01	9.92E−01	1.00E+00
Levy 3	3	1	2	2	1.50E+00	2.50E+00	1.67E+00
LevyMontalvo	3	1	2	2	1.29E+00	8.23E−01	6.36E−01
Mishra 1	4	3	1	2	9.80E−01	9.80E−01	1.00E+00
Mishra 4	3	1	2	2	9.18E−01	8.97E−01	9.77E−01
Mishra 7	3	1	1	2	1.00E+00	1.00E+00	1.00E+00
NeedleEye	1	1	1	1	1.00E+00	1.00E+00	1.00E+00
Norwegian	2	3	1	4	1.04E+00	8.56E−01	8.22E−01
Paviani	4	3	1	2	1.22E+00	1.22E+00	1.00E+00
Penalty 1	4	1	3	2	1.08E+00	1.35E+00	1.25E+00
Penalty 2	3	1	2	2	4.06E+00	7.67E+01	1.89E+01
Pinter 2	3	1	2	2	2.03E+00	1.97E+00	9.68E−01
Price 2	1	1	1	1	1.00E+00	1.00E+00	1.00E+00
Qings	2	1	1	1	2.02E−01	1.04E+03	5.11E+03
Quadric	2	1	3	4	7.36E+05	9.37E+06	1.27E+01
Rana	4	1	3	2	9.04E−01	9.48E−01	1.05E+00
Rastrigin	4	1	3	2	1.54E+00	1.15E+00	7.46E−01
Ripple 25	4	2	3	1	9.82E−01	1.00E+00	1.02E+00
Rosenbrock	2	1	1	1	9.16E−01	6.40E−01	6.99E−01
Salomon	2	1	1	1	1.08E+00	1.05E+00	9.66E−01
Schwefel 1	4	1	3	2	3.03E+54	1.42E+59	4.68E+04
Schwefel 2.26	3	1	2	1	1.20E+00	1.09E+00	9.11E−01
Shubert 4	4	2	3	1	1.05E+00	1.00E+00	9.58E−01
Step 3	2	1	1	1	2.53E+00	1.88E+00	7.45E−01
Trigonometric	4	1	3	2	1.00E+00	1.00E+00	1.00E+00
Vincent	2	1	1	1	1.00E+00	1.00E+00	1.00E+00
Weierstrass	3	4	1	2	7.21E−01	7.31E−01	1.01E+00
XinSheYang 1	3	1	2	2	8.19E+00	6.38E+00	7.78E−01
XinSheYang 3	3	2	1	2	4.16E−16	7.94E+22	1.91E+38
XinSheYang 4	3	2	1	1	2.65E−14	1.55E−13	5.87E+00
Average	3.02	1.37	1.96	1.83			
Deviation	0.91	0.74	0.84	0.77			

For the 22 problems where the tuned PSO ranked better than the stability-guided PSO algorithms, the question is whether the stability-guided PSO algorithms showed totally unacceptable performance or not. To answer this question, refer to the performance ratios provided in Table 3. It is only for two problems (i.e. Quadric and Schwefel 1) that the performance of the stability-guided PSO algorithms were order of magnitude worse than that of the tuned PSO. For the rest of these problems, the performance of the algorithms are in the same order of magnitude.

Where PSO_{sc1} is better than PSO_t, it is to a great extend for five problems (i.e. Brown, Discus, Elliptic, XinSheYang 3, and XinSheYang 4). The same applies for PSO_{sc2}, except for XinSheYang 3 for which PSO_{sc2} performed significantly worse than PSO_t. Where PSO_{sc2} is worse than PSO_{sc1}, it is notably so for seven problems (i.e. Alpine 1, Elliptic, Penalty 2, Qings, Quadric, Schwefel 1, and XinSheYang 3). PSO_{sc1} is significantly worse than PSO_{sc2} for only one problem (i.e. Discus).

7 Conclusions

This paper proposed that values for the three particle swarm optimization (PSO) control parameters be sampled randomly such that a given theoretically derived stability condition is satisfied. Because the stability conditions guarantee that an equilibrium state will be reached, such random sampling is then also ensures that an equilibrium state will be reached. The resulting stability-guide PSO algorithms are then offered as alternatives to having to tune control parameters prior to application of the PSO, and to currently available inefficient self-adaptive PSO algorithms. The empirical analysis of the performance of the stability-guided PSO algorithms has shown that these algorithms perform very competitively in comparison to a well-tuned PSO algorithm. It is only for two problems out of the studied 44 problems that the tuned PSO outperformed the stability-guided PSO algorithms with orders of magnitude.

This paper analyzed the performance of the stability-guided PSO algorithms only on 30-dimensional instances of the benchmark problems. Future studies will evaluate performance on larger-scale problems. Recent research has shown a preference for control parameter values that facilitate exploitative search behavior when PSO is applied to solve large-scale optimization problems [31]. Future work will determine the regions of the stability region that facilitates exploitative behavior, and will develop stability-guide PSO algorithms that bias sampling of control parameter values towards values that facilitate exploitation. The current approaches sample control parameter values per particle. Future work will explore the potential benefit of sampling control parameter values per dimension. Lastly, for the Poli and Broomhead stability conditions, values for the inertia weight were restricted to be in [0, 1], despite the condition allowing values in [−1, 1]. Future work will evaluate the impact if random sampling allows negative inertia weight values.

References

1. Adorio, E.: MVF – Multivariate Test Functions Library in C for Unconstrained Global Optimization. Technical report. University of the Philippines Diliman (2005)
2. Al-Roomi, A.: Unconstrained Single-Objective Benchmark Functions Repository (2015). https://www.al-roomi.org/benchmarks/unconstrained
3. Balaprakash, P., Birattari, M., Stützle, T.: Improvement strategies for the F-Race algorithm: sampling design and iterative refinement. In: Bartz-Beielstein, T., et al. (eds.) HM 2007. LNCS, vol. 4771, pp. 108–122. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-75514-2_9
4. Beielstein, T., Parsopoulos, K.E., Vrahatis, M.N.: Tuning PSO parameters through sensitivity analysis. Universitätsbibliothek Dortmund (2002)
5. Birattari, M., Stëtzle, T., Paquete, L., Varrentrapp, K.: Racing algorithm for configuring metaheuristics. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 11–18 (2002)
6. Bonyadi, M.R., Michalewicz, Z.: Impacts of coefficients on movement patterns in the particle swarm optimization algorithm. IEEE Trans. Evol. Comput. **21**(3), 378–390 (2016)
7. Bratton, D., Kennedy, J.: Defining a standard for particle swarm optimization. In: 2007 IEEE Swarm Intelligence Symposium, pp. 120–127. IEEE (2007)
8. Cleghorn, C., Engelbrecht, A.: A generalized theoretical deterministic particle swarm model. Swarm Intell. **8**(1), 35–59 (2014)
9. Cleghorn, C., Engelbrecht, A.: Particle swarm convergence: an empirical investigation. In: Proceedings of the IEEE Congress on Evolutionary Computation (2014)
10. Cleghorn, C., Engelbrecht, A.: Particle swarm optimizer: the impact of unstable particles on performance. In: Proceedings of the IEEE Swarm Intelligence Symposium (2016)
11. Cleghorn, C., Engelbrecht, A.: Particle swarm stability a theoretical extension using the non-stagnate distribution assumption. Swarm Intell. **12**(1), 1–22 (2018)
12. Clerc, M., Kennedy, J.: The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Trans. Evol. Comput. **6**(1), 58–73 (2002)
13. Dobslaw, F.: A parameter tuning framework for metaheuristics based on design of experiments and artificial neural networks. Int. J. Aerosp. Mech. Eng. **64**, 213–216 (2010)
14. Eberhart, R., Shi, Y.: Comparing inertia weights and constriction factors in particle swarm optimization. In: Proceedings of the IEEE Congress on Evolutionary Computation (2000)
15. Engelbrecht, A.: Particle swarm optimization with crossover: a review and empirical analysis. Artif. Intell. Rev. **45**(2), 131–165 (2016)
16. Engelbrecht, A.: Inertia weight control strategies: particle roaming behavior. In: International Conference on Soft Computing and Machine Intelligence (2017)
17. Erwin, K., Engelbrecht, A.: A tuning free approach to multi-guide particle swarm optimization. In: Proceedings of the IEEE Swarm Intelligence Symposium (2021)
18. Gavana, A.: Global Optimisation Benchmarks. http://infinity77.net/global_optimization/index.html. Accessed 31 Mar 2022

19. Harrison, K., Engelbrecht, A., Ombuki-Berman, B.: Inertia control strategies for particle swarm optimization: too much momentum, not enough analysis. Swarm Intell. **10**(4), 267–305 (2016)
20. Harrison, K., Engelbrecht, A., Ombuki-Berman, B.: Optimal parameter regions and the time-dependence of control parameter values for the particle swarm optimization algorithm. Swarm Evol. Comput. **41**, 20–35 (2018)
21. Harrison, K., Engelbrecht, A., Ombuki-Berman, B.: Self-adaptive particle swarm optimization: a review and analysis of convergence. Swarm Intell. **12**, 187–226 (2018)
22. Harrison, K., Ombuki-Berman, B., Engelbrecht, A.: Optimal parameter regions for particle swarm optimization algorithms. In: Proceedings of the IEEE Congress on Evolutionary Computation (2017)
23. Harrison, K.R., Ombuki-Berman, B.M., Engelbrecht, A.P.: An analysis of control parameter importance in the particle swarm optimization algorithm. In: Tan, Y., Shi, Y., Niu, B. (eds.) ICSI 2019. LNCS, vol. 11655, pp. 93–105. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26369-0_9
24. Harrison, K., Ombuki-Berman, B., Engelbrecht, A.: The parameter configuration landscape: a case study on particle swarm optimization. In: Proceedings of the IEEE Congress on Evolutionary Computation (2019)
25. Jain, N., Nangia, U., Jain, J.: Impact of particle swarm optimization parameters on its convergence. In: Proceedings of the 2nd IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems, pp. 921–926 (2018)
26. Jamil, M., Yang, X.S.: A literature survey of benchmark functions for global optimization problems. Int. J. Math. Model. Numer. Optim. **4**(2), 150–194 (2013)
27. Jiang, M., Luo, Y., Yang, S.: Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm. Inf. Process. Lett. **102**(1), 8–16 (2007)
28. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN 1995-International Conference on Neural Networks, vol. 4, pp. 1942–1948. IEEE (1995)
29. Klazar, R., Engelbrecht, A.: Parameter optimization by means of statistical quality guides in F-Race. In: Proceedings of the IEEE Congress on Evolutionary Computation (2014)
30. Liang, J., Qu, B., Suganthan, P.: Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. Technical report. Tech. Rep. 201311. Zhengzhou University and Nanyang Technological University (2013)
31. Oldewage, E., Engelbrecht, A., Cleghorn, C.: Movement patterns of a particle swarm in high dimensions. Inf. Sci. **512**, 1043–1062 (2020)
32. Pedersen, M.: Good parameters for particle swarm optimization. Technical report. HL1001. Hvass Laboratories (2010)
33. Poli, R.: Mean and variance of the sampling distribution of particle swarm optimizers during stagnation. IEEE Trans. Evol. Comput. **14**(4), 712–721 (2009)
34. Poli, R., Broomhead, D.: Exact analysis of the sampling distribution for the canonical particle swarm optimiser and its convergence during stagnation. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 134–141 (2007)
35. Scheepers, C., Engelbrecht, A.P., Cleghorn, C.W.: Multi-guide particle swarm optimization for multi-objective optimization: empirical and stability analysis. Swarm Intell. **13**(3–4), 245–276 (2019)

36. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: Proceedings of the IEEE International Conference on Evolutionary Computation Proceedings, pp. 69–73 (1998)
37. Smith, S., Eiben, A.: Comparing parameter tuning methods for evolutionary algorithms. In: Proceedings of the IEEE Congress on Evolutionary Computation (2009)
38. Van den Bergh, F., Engelbrecht, A.: A study of particle swarm optimization particle trajectories. Inf. Sci. **176**(8), 937–971 (2006)