Optimal Parameter Regions and the Time-Dependence of Control Parameter Values for the Particle Swarm Optimization Algorithm

Kyle Robert Harrison[a], Andries P. Engelbrecht[a,*], Beatrice M. Ombuki-Berman[b]

[a] *Department of Computer Science, University of Pretoria, Pretoria, South Africa*
[b] *Department of Computer Science, Brock University, St. Catharines, Canada*

Abstract

The particle swarm optimization (PSO) algorithm is a stochastic search technique based on the social dynamics of a flock of birds. It has been established that the performance of the PSO algorithm is sensitive to the values assigned to its control parameters. Many studies have examined the long-term behaviours of various PSO parameter configurations, but have failed to provide a quantitative analysis across a variety of benchmark problems. Furthermore, two important questions have remained unanswered. Specifically, the effects of the balance between the values of the acceleration coefficients on the optimal parameter regions, and whether the optimal parameters to employ are time-dependent, warrant further investigation. This study addresses both questions by examining the performance of a global-best PSO using 3036 different parameter configurations on a set of 22 benchmark problems. Results indicate that the balance between the acceleration coefficients does impact the regions of parameter space that lead to optimal performance. Additionally, this study provides concrete evidence that, for the examined problem dimensions, larger acceleration coefficients are preferred as the search progresses, thereby indicating that the optimal parameters are, in fact, time-dependent. Finally, this study provides a general recommendation for the selection of PSO control parameter

*Corresponding author
Email address: engel@cs.up.ac.za (Andries P. Engelbrecht)

values.

Keywords: Particle swarm optimization, control parameter values, time-dependence

1. Introduction

An effective search technique must strike a balance between exploring new regions in the search space and exploiting known, promising regions. In the particle swarm optimization (PSO) algorithm, exploration and exploitation can be controlled by the values of the control parameters [1, 2, 3, 4, 5, 6]. Moreover, the performance of the PSO algorithm can be improved by appropriately tuning the control parameters to the current problem [7, 8, 9, 10], given that the particle movement patterns are influenced by the values of the control parameters [6, 8].

While there is no doubt that parameter tuning in the PSO algorithm is important, especially when more complex optimization problems are considered, the task of effectively tuning the parameters is computationally expensive. Parameter tuning is often an arduous manual process whereby a large number of candidate parametrizations must be examined and analysed. While there have been various automated parameter configuration tools proposed (for example, the F-Race algorithm [11]), such tools have two clear drawbacks. Firstly, automated methods simply automate the process of parameter tuning and do not necessarily reduce the time complexity of the parameter search. Although various optimizations can be made, such as removing a particular parameter configuration from consideration if enough evidence is gathered to deduce that the configuration is poorly performing, the original argument still stands in that this does not necessarily reduce the overall complexity of the control parameter tuning problem. Rather, it simply translates a manual process into an automated process. Secondly, there is an implicit, often-overlooked assumption in *a priori* parameter tuning that the optimal parameter configuration does not change over time. With an automated parameter tuning strategy (and, for that matter, manual tuning techniques), the tuned parameters will be statically used

throughout the course of the search. This is likely not an optimal scenario given that there is a well-established ideology that the best parameters change over time. For instance, the time-varying inertia weight PSO by Shi and Eberhart [12, 13] is premised on the idea of reducing the value of the inertia weight (described in Section 2) over time. Leonard and Engelbrecht [5] empirically found that parameters well-suited for exploration were not well-suited for exploitation, and *vice versa*. Moreover, heterogeneous PSO algorithms have evidenced that the most suitable velocity update scheme to employ varies during the search [14, 15, 16, 17].

To alleviate the issue of time-sensitive parameter values, various self-adaptive particle swarm optimization (SAPSO) algorithms that adapt their control parameters over time have been proposed [12, 18, 19, 20, 21, 22, 23]. While many such adaptive schemes have been proposed, their performance has largely been unconvincing [24, 25, 26, 27]. However, the poor performance of SAPSO algorithms can be somewhat explained by the fact that such algorithms are concurrently optimizing two highly inter-dependent continuous search problems. The inter-dependence is twofold for such algorithms. On one hand, the best parameters to employ (i.e., the secondary search problem) are dependent upon the current state of the primary search. On the other hand, the performance of the primary search is heavily dependant upon the identification of good parameters in the secondary search. To further complicate the matter, a poor set of parameters will almost certainly lead to poor primary search performance. Therefore, it is crucial that a SAPSO algorithm perform an effective parameter search. By extension, any reduction in the complexity of the parameter search will almost certainly lead to an improvement in performance for SAPSO algorithms.

There has been a number of studies that have empirically examined the performance of various PSO parameter configurations [6, 8, 9, 10, 28, 29]. However, there is no general consensus as to which parameter configurations lead to the best performance. Most empirical studies examined a limited set of parameter configurations over a small number of benchmark problems and for specific problem dimensionalities, thereby providing recommended parametriza-

tions. Recently, Cleghorn and Engelbrecht [28] examined 1264 parameter configurations over 28 benchmark problems and identified regions of the parameter space that lead the PSO algorithm to perform worse than a random search. Similarly, Harrison et al. [29] empirically investigated 1012 parameter configurations over a set of 22 benchmark functions with the primary objective of determining which areas of parameter space lead to good performance of the PSO algorithm. Both of these studies have led to an enhanced understanding of the general region in parameter space where good parameter configurations lie. However, neither study answered an important question, namely whether the best parameters to employ are in fact time-dependent. Similarly, both studies assumed that the cognitive and social control parameters (described in Section 2) have equal values. Thus, an additional, equally important question remains unanswered: in what regions of parameter space do the best parameters reside when the values of these two parameters are not equal?

This paper provides answers to both questions presented above, namely how the best parameter values change over time and what regions of parameter space lead to the best performance when the cognitive and social control parameters are not equal. To this end, 3036 parameter configurations of the PSO algorithm are examined over a set of 22 benchmark problems. The performance of each parameter configuration is captured at various points throughout the search to give an indication of whether the best performing parameter configurations remain static throughout the entire search. Moreover, the performance is correlated with the best-known convergence criterion [30, 31] to ascertain the effect of particle convergence/stability on the overall performance of the PSO algorithm.

This paper contributes to the understanding of PSO by concluding that the balance between the social and cognitive acceleration coefficients has a significant impact on the areas in parameter space that lead to good performance. Moreover, this study shows that the optimal regions in parameter space shift over time, thus providing direct evidence in support of SAPSO algorithms.

The remainder of this paper is structured as follows. Section 2 presents the

PSO algorithm and discusses the theoretical convergence criterion used in this study. Section 3 discusses the experimental procedures and analysis techniques as well as the results regarding the identification of optimal parameter regions. Finally, concluding remarks and avenues of future work are given in Section 4.

2. Particle swarm optimization

The PSO algorithm consists of a collection of agents, referred to as particles, where each particle represents a candidate solution to the current optimization problem. Each particle retains three pieces of information, namely its current position, velocity, and its personal best found position within the search space. Movement of particles is governed by an iterative calculation and addition of a velocity vector to the position vector. The calculation of each particle's velocity is based on its attraction towards two promising locations in the search space, namely the best position found by the particle and the best position found by any particle within the particle's neighbourhood. The neighbourhood of a particle refers to the other particles within the swarm from which it may take influence. The original PSO algorithm employed one of two neighbourhood strategies, either a star topology where the neighbourhood is the entire swarm, or a ring topology where the neighbourhood consists of the immediate neighbours when the particles are arranged in a ring [32]. The star and ring neighbourhood topologies, commonly referred to as global-best and local-best, respectively, have been found to exhibit no statistically significant difference in performance when aggregated over a wide variety of benchmark functions [33]. However, the topology must be considered as a parameter to be tuned given that the best topology to employ is dependent upon both the optimization problem and computational budget [33, 34]. Furthermore, it was shown by Harrison et al. [29] that the topology employed has a noticeable influence on the regions in parameter space that lead to good performance.

For the purposes of this study, a global-best topology is employed. The velocity is then calculated for particle i according to the inertia weight model

of [12] as

$$\begin{aligned} v_{ij}(t+1) = \omega v_{ij}(t) &+ c_1 r_{1ij}(t)(y_{ij}(t) - x_{ij}(t)) \\ &+ c_2 r_{2ij}(t)(\hat{y}_j(t) - x_{ij}(t)) \end{aligned} \tag{1}$$

where $v_{ij}(t)$ and $x_{ij}(t)$ are the velocity and position in dimension j at time t, respectively. The inertia weight is given by ω while c_1 and c_2 represent the cognitive and social acceleration coefficients, respectively. The stochastic component of the algorithm is provided by the random constants, $r_{1ij}(t), r_{2ij}(t) \sim U(0,1)$. Finally, $y_{ij}(t)$ and $\hat{y}_j(t)$ denote the personal and neighbourhood best positions in dimension j, respectively. Particle positions are then updated according to

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t+1). \tag{2}$$

2.1. Convergence of particle swarm optimization

There has been a significant amount of research effort devoted to the theoretical study of PSO convergence [2, 3, 30, 31, 35, 36, 37, 38]. Specifically, researchers have been interested in the regions of parameter space that lead to convergent behaviour for the PSO algorithm. Given the stochastic nature of the PSO algorithm, various assumptions have been made to assist in the derivation of the convergent region. One such assumption is known as the stagnation assumption, which makes the simplifying assumption that the personal and neighbourhood best positions are fixed and do not move throughout the search. However, the various assumptions employed by these theoretical studies have proven to be limiting, and therefore many of these purely theoretical studies do not capture the true convergent region [39]. Empirical simulations (without any simplifying assumptions) by Cleghorn and Engelbrecht [39] have shown that, of the various convergence criteria, the criterion

$$c_1 + c_2 < \frac{24(1-\omega^2)}{7-5\omega}, \tag{3}$$

as proposed by Poli and Broomhead [30] and Poli [31], is the most accurate in practice. The region defined by Equation (3) is illustrated in Figure 1, where

parameter values that lie within the parabolic region are theoretically convergent. Note that convergence, in this context, does not necessary imply that the swarm has converged to a single point. Rather, convergence is defined to be order-2 stability such that the limit of the variance of particle movements is zero [31]. While the region defined by Equation (3) was derived using the stagnation assumption, Bonyadi and Michalewicz [40] have shown that an equivalent criterion can be derived without the stagnation assumption.

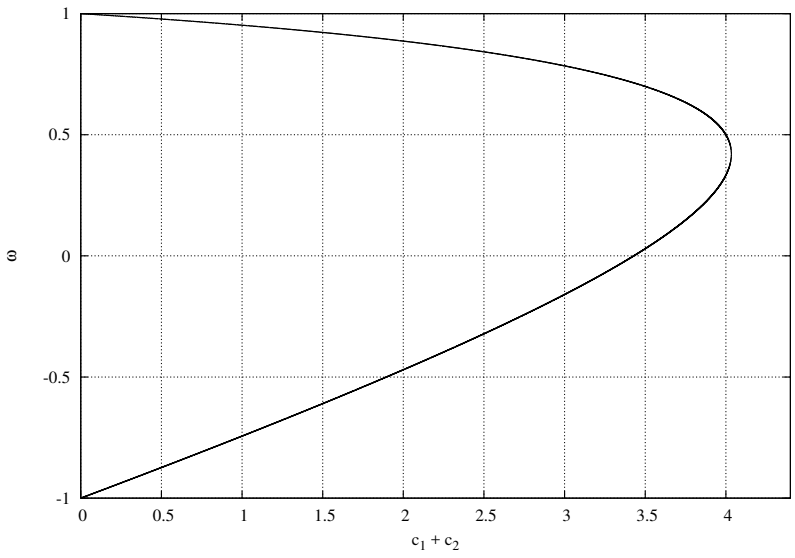


Figure 1: Visualization of Poli's theoretically defined region for convergent behaviour of PSO parameters.

3. Optimal parameter regions

To identify the regions of parameter space that lead to good performance, a total of 3036 parameter configurations were examined on a set of 22 benchmark problems, as summarized in Table 1. All of the benchmark functions were evaluated in 30 dimensions. Further descriptions of the benchmark functions can be found in Appendix A. Each experiment made use of synchronous updates [41] and ran for 5000 iterations with a swarm size of 30. Experiments were repeated

30 times using the global best (star) topology. To prevent invalid attractors, personal best positions were only updated if the new position was both feasible and had a better objective function value than the previous personal best position. Particles were initialized uniformly within the feasible region, with initial velocities of zero [42].

Experiments were grouped into three categories based on the ratio between the cognitive and social control parameter values to ascertain the effects of imbalanced acceleration coefficients. The experiments were labelled 'equal', where $c_1 = c_2$, 'cognitive', where $c_1 = 3c_2$, and 'social', where $c_2 = 3c_1$. For each experiment type, the parameter configurations were constructed as sampled points, (C, w), taken every 0.1 units within the ranges

$$C \in [0.1, 4.4] \text{ and } \omega \in [-1.1, 1.1].$$

Note that while negative inertia weights are not traditionally employed in the PSO algorithm, recent results have shown that negative inertia weights can lead to stable behaviour and thus are not necessarily unreasonable [40, 43]. The values assigned to the control parameters c_1 and c_2 were then calculated based on the type of experiment. The control parameter values were taken as $c_1 = c_2 = \frac{C}{2}$ for experiments labelled 'equal', $c_1 = \frac{3C}{4}$ and $c_2 = \frac{C}{4}$ for experiments labelled 'cognitive', and $c_1 = \frac{C}{4}$ and $c_2 = \frac{3C}{4}$ for experiments labelled 'social'. For the cognitive and social experiments, a multiplier value of three was chosen to provide an environment where either the cognitive or social acceleration coefficient would dominate the other without resorting to a cognitive-only or social-only model. Each experiment group thus consisted of 1012 parameter configurations, leading to a combined total of 3036 parameter configurations examined in this study.

3.1. Statistical analysis

Results were analysed using the following statistical analysis procedure. For each benchmark problem, a Kruskal-Wallis test was performed to first determine if any significant differences existed among the fitness values obtained by

Table 1: Characteristics of the benchmark functions. 'Equation' specifies the equation number of the function (see Appendix A). 'Modality' specifies the modality ('U' for unimodal, 'M' for multimodal). 'Separability' denotes the separability ('S' for separable, 'NS' for non-separable).

Function	Name	Equation	Modality	Separability
f_1	Absolute Value	(A.1)	U	S
f_2	Ackley	(A.2)	M	NS
f_3	Alpine	(A.3)	M	S
f_4	Egg Holder	(A.4)	M	NS
f_5	Elliptic	(A.5)	U	S
f_6	Griewank	(A.6)	M	NS
f_7	HyperEllipsoid	(A.7)	U	S
f_8	Michalewicz	(A.8)	M	S
f_9	Norwegian	(A.9)	M	NS
f_{10}	Quadric	(A.10)	U	NS
f_{11}	Quartic	(A.11)	U	S
f_{12}	Rastrigin	(A.12)	M	S
f_{13}	Rosenbrock	(A.13)	M	NS
f_{14}	Salomon	(A.14)	M	NS
f_{15}	Schaffer 6	(A.15)	M	NS
f_{16}	Schwefel 1.2	(A.16)	U	NS
f_{17}	Schwefel 2.21	(A.17)	U	S
f_{18}	Schwefel 2.22	(A.18)	U	S
f_{19}	Shubert	(A.19)	M	NS
f_{20}	Spherical	(A.20)	U	S
f_{21}	Step	(A.21)	M	S
f_{22}	Vincent	(A.22)	M	S

using each of the parameter configurations. If the Kruskal-Wallis test indicated that a significant difference existed, pairwise Mann-Whitney U tests were then performed to identify the individual differences. When the Mann-Whitney U test indicated that a difference in performance existed, the median fitness values were used to assign wins and losses; the parameter configuration that lead to better performance was awarded a win, while the inferior configuration was awarded a loss. Finally, the parameter configurations were ranked based on the difference between the number of wins and losses. In this context, a lower rank corresponds to superior performance. Both the Kruskal-Wallis and Mann-Whitney U tests were performed at a confidence level of 0.05.

3.2. Results and discussion

This section presents the results of the experiments described in Section 3. Firstly, the overall performance of the parameter configurations is examined, followed by an examination of the time-dependence.

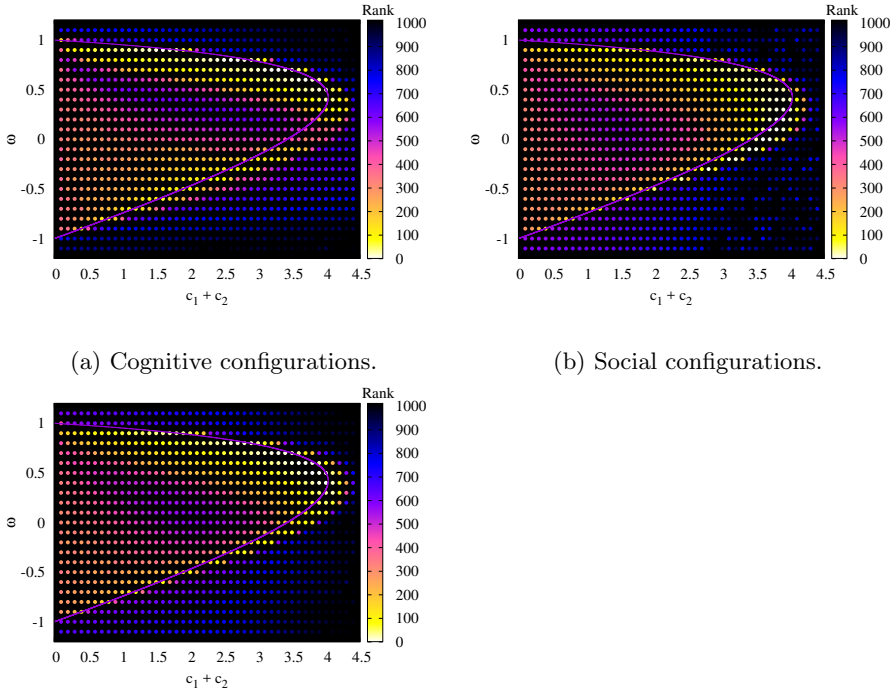
3.2.1. Overall Performance

Figure 2 depicts the overall rank of each parameter configuration for each experiment after 5000 iterations. While each of the experiments depicts a tendency for parameters near the boundary of the convergent region to perform best, there are observable differences regarding where the best parameters lie. Such observations are reinforced by Figure 3, which visualizes the 100 best parameter configurations by overall rank after 5000 iterations for each experiment type. For the cognitive experiments, a majority of the best parameter configurations are clustered along the upper boundary. However, there is also a notable cluster of points along the lower boundary. For the social experiments, the best parameters are clustered very strongly around the apex, with a slight preference for positive values of ω. For the equal experiments, the best parameters are mostly clustered around the apex, with a slightly more pronounced preference for positive values of ω than the social configurations.

Examining Figure 2, it is also apparent that the regions leading to the worst

performance are significantly different among the various configurations. For each of the experiment types, it is clear that there is a tendency for parameters that lie outside the convergent region to perform worse than those within the region. However, the exact areas that lead to the absolute worst performance are notably different. For the cognitive experiments, the worst parameters are mostly dependent upon the value of ω. That is, configurations where $|\omega| \approx 1$ (albeit more prominently with $\omega \approx -1$) tend to perform the worst, largely irrespective of the values of c_1 and c_2. For the social configurations, the worst performance is scattered throughout the extreme ends of the examined parameter space. Examining the equal configurations, the worst performance is again clustered around the extreme ends of the examined parameter space. However, it is noted that, for parameter configurations that lie outside the convergent region, there is a definite correlation between distance to the convergent region and performance. Specifically, it is clear that performance degradation is proportional to the distance from the convergent region. It is also noted that, for each experiment type, there is a cluster within the convergent region that leads to relatively poor performance.

Table 2 provides a summary of the fitness attained by the best performing parameter configuration for each benchmark problem. In the event of a tie, the parameter configuration that lead to the lowest median fitness was selected. In the cases of f_{21} and f_{22}, no definitive best configuration could be chosen given that there were multiple parameter configurations that always lead to the optimal fitness being attained. For f_{21}, there were 69 configurations that always lead to the optimal fitness while there were three such parameter configurations for f_{22}. From Table 2, it is clear that the best parameter values to employ are problem specific. Furthermore, there is no clear configuration type that outperformed the others. However, when the best configurations are correlated with modality, a few observations can be made. For seven of the functions, six of which were unimodal, an equal configuration lead to the best performance. For five of the functions, all of which were multimodal, a cognitive configuration lead to the best performance. For eight of the functions, five of which were mul-



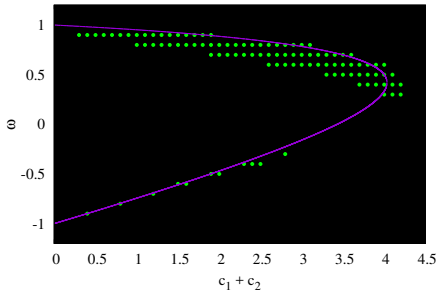
(a) Cognitive configurations.

(b) Social configurations.

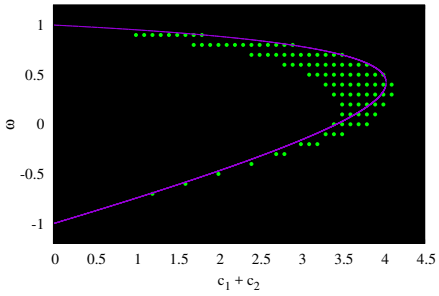
(c) Equal configurations.

Figure 2: Overall rank after 5000 iterations based on experiment type.

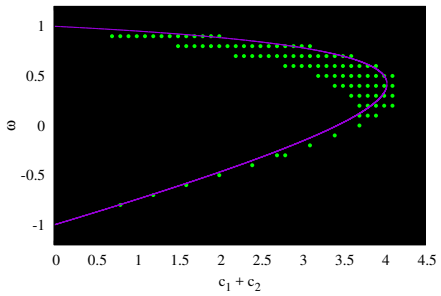
timodal, a social configuration lead to the best performance. In other words, the best performance was attained for six out of nine unimodal problems when $c_1 = c_2$ and in no instance did a cognitive configuration lead to the best performance on a unimodal problem. In contrast, parameter configurations that had $c_1 = c_2$ attained the best performance on only one multimodal problem whereas the cognitive and social configurations each lead to the best performance on five multimodal problems. A further observation regarding Table 2 was made when the best parameters were correlated with the convergence criterion given in Equation (3). For five of the benchmark problems, all of which were multimodal, the best parameter configuration violated the convergence criterion. However, it should be noted that violating the convergence criterion does not necessarily imply divergence, but rather that convergence cannot be



(a) Cognitive configurations.



(b) Social configurations.



(c) Equal configurations.

Figure 3: 100 best parameter configurations after 5000 iterations by overall rank.

guaranteed.

Table 3 presents the 10 best parameter configurations determined by aggregate rank across all benchmark problems. Of the best 10 parameter configurations, four have $c_1 = c_2$ while six are social; none of the best 10 parameter configurations were cognitive. In fact, the best cognitive configuration had a rank of 19. Moreover, of the best 100 parameter configurations, 54 were categorized as social, 36 were categorized as equal, while only nine were cognitive configurations. This result is not surprising given that the strength of the PSO algorithm lies in its social aspect. Three of the best 10 configurations employed a negative inertia weight, indicating a preference to switch directions rather than resist directional changes. Despite the convergent region defined by Equation (3) containing negative inertia weight values, they are rarely, if ever, used

Table 2: Summary of the fitness attained by the best parameter configuration for each benchmark function. A * indicates that multiple parameter configurations were found that always lead to the optimal fitness.

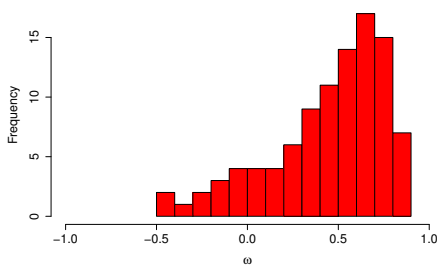
Function	ω	c_1	c_2	Median	Average	Standard Deviation
f_1	0.4	2.000	2.000	3.17E-048	2.94E-042	1.24E-0.041
f_2	0.7	2.550	0.850	6.66E-015	5.01E-002	2.74E-001
f_3	0.7	2.625	0.875	1.07E-014	1.69E-014	3.15E-014
f_4	-0.1	0.875	2.625	-1.57E+004	-1.59E+004	1.25E+003
f_5	0.4	1.950	1.950	2.76E-100	9.64E-094	3.81E-093
f_6	0.5	0.975	2.925	3.70E-003	1.01E-002	1.81E-002
f_7	0.4	1.950	1.950	9.96E-105	1.13E-099	5.99E-099
f_8	-0.5	1.500	0.500	-2.70E+001	-2.69E+001	8.34E-001
f_9	0.6	1.850	1.850	-7.85E-001	-7.88E-001	6.88E-003
f_{10}	0.5	1.750	1.750	6.69E-014	2.12E-013	5.46E-013
f_{11}	0.5	1.750	1.750	1.11E-179	7.52E-170	0.00E+000
f_{12}	-0.1	0.900	2.700	1.39E+001	1.51E+001	7.93E+000
f_{13}	0.8	0.500	1.500	2.47E+000	3.47E+000	3.55E+000
f_{14}	0.8	2.025	0.675	3.00E-001	3.20E-001	5.51E-002
f_{15}	0.1	0.950	2.850	4.51E+000	4.56E+000	6.36E-001
f_{16}	0.2	0.900	2.700	2.60E-014	3.10E-013	8.45E-013
f_{17}	0.0	0.850	2.550	1.02E-007	1.41E-007	1.57E-007
f_{18}	0.4	2.000	2.000	8.21E-048	9.40E-036	5.15E-035
f_{19}	0.8	2.025	0.675	-1.89E+034	-2.02E+034	9.27E+033
f_{20}	0.4	1.950	1.950	1.31E-106	1.08E-100	5.82E-100
f_{21}	*	*	*	0.00E+000	0.00E+000	0.00E+000
f_{22}	*	*	*	-3.00E+001	-3.00E+001	0.00E+000

Table 3: The 10 best parameter configurations by overall rank across all benchmark problems.

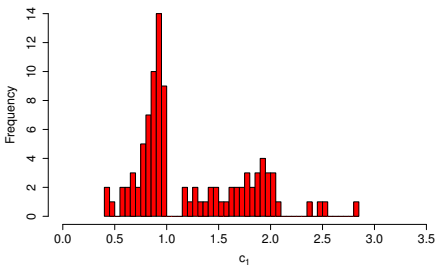
Overall Rank	ω	c_1	c_2	Average Rank	Rank SD
1	0.1	0.950	2.850	67.955	43.951
2	-0.1	0.875	2.625	64.864	54.961
3	0.0	0.900	2.700	66.727	49.891
4	0.0	0.925	2.775	67.364	45.390
5	0.6	1.800	1.800	68.591	58.467
6	0.5	1.900	1.900	67.773	58.557
7	0.7	1.650	1.650	67.500	47.914
8	-0.2	0.800	2.400	76.227	61.082
9	-0.3	0.700	2.100	75.227	50.191
10	0.6	1.850	1.850	67.864	55.555

in practice. However, these results suggest that negative inertia weights may not necessarily be detrimental to the search process. A further two of the best 10 configurations employed no inertia at all (i.e., $\omega = 0$). Given that half of the best 10 parameter configurations employed non-positive inertia weight values, it can be concluded that it is not always beneficial for particles to remain on their current trajectory. However, it should be noted that all five of the parameter configurations that had non-positive inertia weights also were categorized as social configurations while four of the five configurations with positive inertia weight values had equal social and cognitive coefficients. This suggests that having a high social influence may, in part, make having a positive inertia weight value unnecessary. Nonetheless, it is generally not recommended to employ a negative inertia weight, especially for cognitive parameter configurations (see Section 3.2.2 for a further discussion of negative inertia weights).

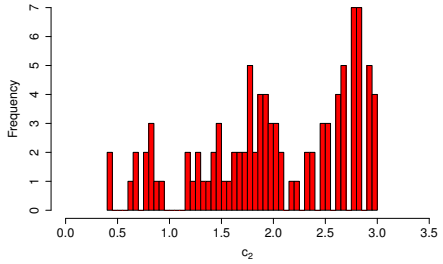
Figure 4 shows the distribution of the values for each control parameter over the 100 best-performing parameter configurations. From Figure 4a, it is evident that the most frequent well-performing inertia weight is 0.7. There is a very skewed distribution, showing a steady increase in frequency for inertia weight



(a) Inertia weight.



(b) Cognitive acceleration coefficient.



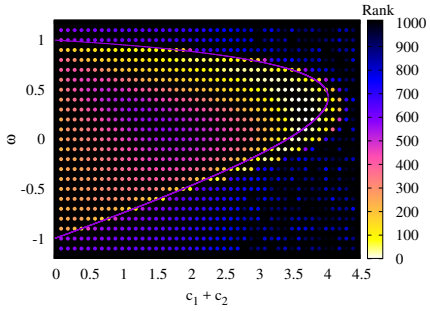
(c) Social acceleration coefficient.

Figure 4: Distribution of the values of the best 100 parameter configurations after 5000 iterations.

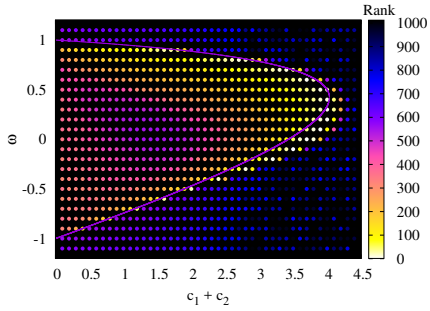
values between -0.5 and 0.7, after which the frequency declines. For the distribution of the cognitive acceleration coefficient, depicted in Figure 4b, a value of approximately 0.900 frequently leads to good performance. Furthermore, there is a noticeable preference for cognitive acceleration coefficients between 0.7 and 1.0. Regarding the distribution of the social acceleration coefficient, larger values are preferred, in general. The highest frequency was observed with social coefficients of approximately 2.75. There is also a notable cluster of good performance when the value of the social acceleration coefficient is between 1.6 and 2.0. These results further demonstrate that, in general, it is preferable to have a larger value for the social coefficient than the cognitive coefficient despite these two parameter values being traditionally taken as equal in the literature.

Figure 5 presents the overall rank of each parameter configuration based on modality. It is evident from Figures 5a, 5b, 5e, and 5f that the effects of modality are minimal when equal and social parameter configurations are employed. However, it is notable that for both social and equal parameter configurations, the region that leads to the best performance shifts toward the boundaries of the convergent region when faced with multimodal problems. That is, larger acceleration coefficients in the best parameter configurations tend to lead to better performance on multimodal problems. Figures 5c and 5d indicate that modality has a noticeable effect on the relative performance of cognitive parameter coefficients, specifically the regions that lead to poor performance. While the region leading to the best performance, namely the top boundary of the convergent region, does not change significantly, the central region where poor performance was observed (i.e., the darker region) is much larger when faced with unimodal problems. When faced with multimodal problems, the performance of cognitive configurations within the convergent region tend to improve as the acceleration coefficients increase. However, when faced with unimodal problems, the cognitive coefficients have two notable regions leading to good performance, namely the bottom left and top right regions of the convergent region, indicating that either a negative inertia weight and small acceleration coefficients or a large inertia weight and large acceleration coefficients lead to the best performance.

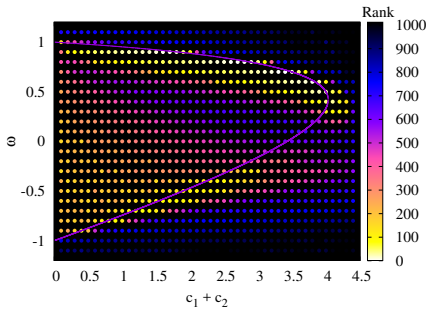
Figure 6 shows the distribution of the values for each control parameter over the 100 best-performing parameter configurations based on modality. When considering the inertia weight, shown in Figures 6a and 6b, it is evident that unimodal problems have a much smaller range in which the inertia weight can lie while leading to good performance. Notably, multimodal problems have a greater tolerance for negative inertia weights as evidenced by the longer left tail. Furthermore, Figure 6a depicts a higher peak with a larger density surrounding it, suggesting that the deviation of the best inertia weight is much smaller for unimodal problems. When considering the cognitive acceleration coefficient in Figures 6c and 6d, there is a notable tendency for values between 0.5 and 1



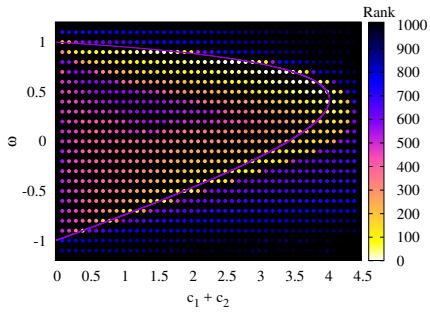
(a) Social configurations, unimodal problems.



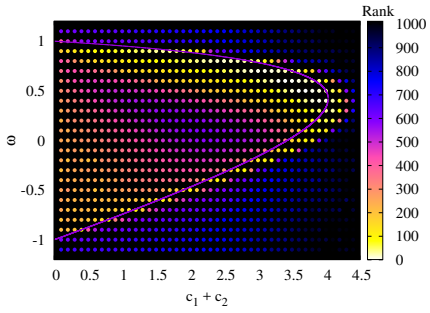
(b) Social configurations, multimodal problems.



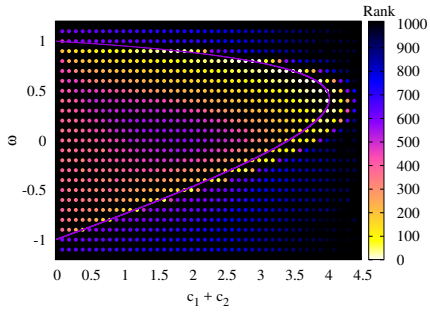
(c) Cognitive configurations, unimodal problems.



(d) Cognitive configurations, multimodal problems.



(e) Equal configurations, unimodal problems.



(f) Equal configurations, multimodal problems.

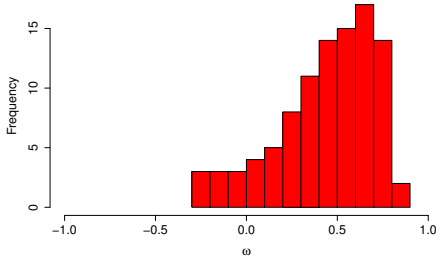
Figure 5: Overall rank after 5000 iterations based on modality.

Table 4: Best parameter configuration by environment type.

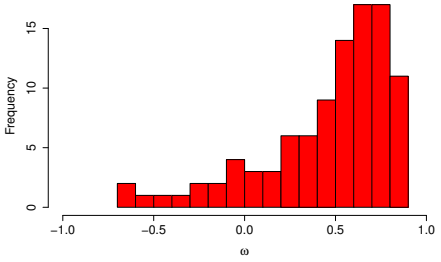
Environment	ω	c_1	c_2	Average Rank	Rank SD
Overall	0.1	0.950	2.850	67.955	43.951
Unimodal	0.5	1.850	1.850	20.333	18.635
Multimodal	-0.1	0.875	2.625	43.846	39.497

to perform well. When faced with multimodal problems, larger values of the cognitive coefficient are acceptable than when faced with unimodal problems, as evidenced by the longer right tail in Figure 6d. Examining the distribution of the social acceleration coefficient, shown in Figures 6e and 6f, it is evident that larger values (i.e., between 2.5 and 3) have a tendency to perform better on unimodal problems, while multimodal problems have a weaker dependence on the value of the social coefficient. However, the best performance on multimodal problems is still obtained within the same region of $[2.5, 3]$. These results suggest that it is preferable to have larger values for the social acceleration coefficient than the cognitive acceleration coefficient, regardless of the modality of the problem.

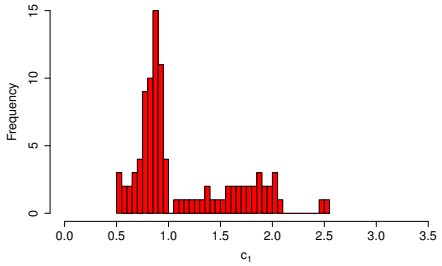
Table 4 presents the best parameter configurations for each environment type, as determined by aggregate rank. For unimodal problems, the best parameter configurations were within the convergent region while the overall best parameter configuration for multimodal problems violated the criterion provided in Equation (3). However, it is noteworthy that the average rank was lower for the best parameter configurations in unimodal environments, i.e., where the best parameter configurations were theoretically convergent. Specifically, the average rank and standard deviation were much lower for unimodal environments, indicating that the observed best parameter configuration performed well consistently across the unimodal problems. Reinforcing what was observed in Table 2, the best configuration for unimodal environments had $c_1 = c_2$, while the best configuration for multimodal environments had a social configuration.



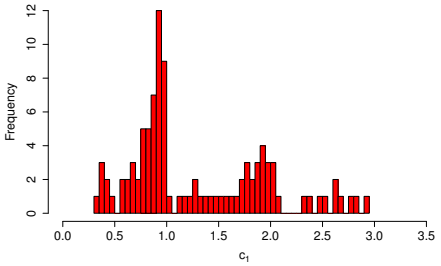
(a) Inertia weight, unimodal problems.



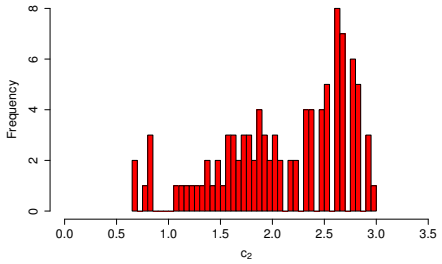
(b) Inertia weight, multimodal problems.



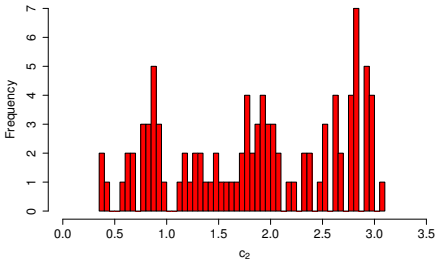
(c) Cognitive acceleration coefficient, unimodal problems.



(d) Cognitive acceleration coefficient, multimodal problems.



(e) Social acceleration coefficient, unimodal problems.



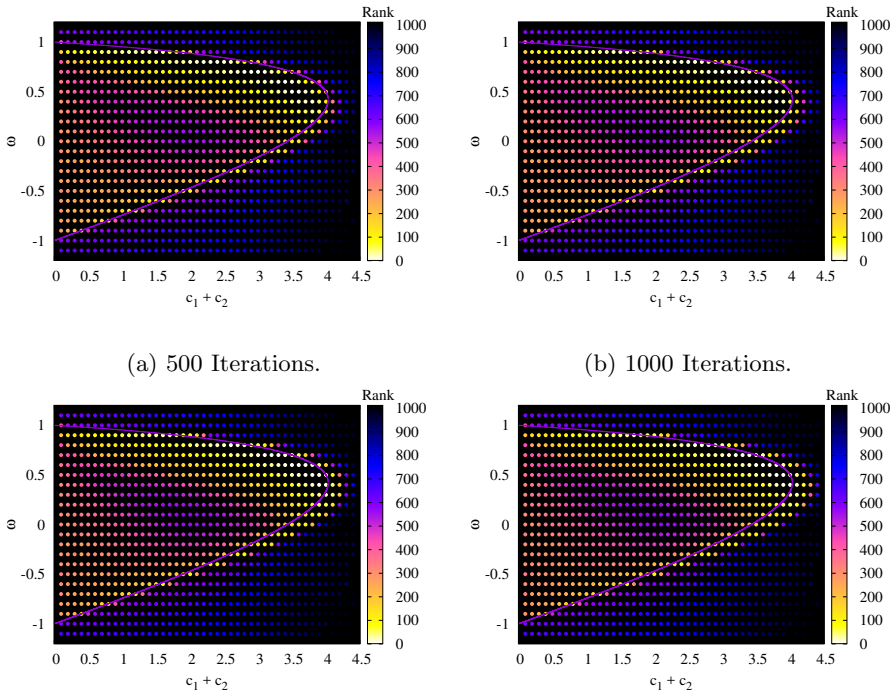
(f) Social acceleration coefficient, multimodal problems.

Figure 6: Distribution of the values of the best 100 parameter configurations after 5000 iterations based on modality.

3.2.2. Time dependence

To assess the dependence of the optimal parameter region on time, Figures 7 to 9 present the overall rank of each parameter configuration at various iterations. A key observation is that the region containing the best performing parameter configurations shifts over time. Specifically, as time passes, there is a noticeable preference for larger values of $c_1 + c_2$. Similarly, there is an improvement in the relative performance of parameter sets that lie just outside the convergent region but near the apex. This is evidenced by the emergence of a greater number of light-coloured points outside the convergent region, but still near the apex, as the number of iterations increases. This clearly indicates that larger social and cognitive acceleration coefficients are preferred later in the search process. However, this also suggests that the dependence on the convergence criterion is weaker as the search progresses.

To further assess the relative performance of parameters over time, Figure 10 depicts the best 100 parameter configurations (based on aggregate rank) at various iterations for each of the configuration types. The most important observation is that the best parameter configurations are noticeably different at each of the examined time intervals. Specifically, there is a tendency for parameters to shift towards the right over time, implying that larger values of the social and cognitive parameters are preferred later in the search process. In contrast, the results suggest that the inertia weight value is less dependent on time in that the optimal parameter region shifts much more horizontally than vertically. This result provides direct evidence against dynamically reducing the inertia weight over time and further confirms the findings of Harrison et al. [25], where it was found that linearly decreasing inertia weight strategies perform worse than a constant inertia weight. An additional, noteworthy observation is that there is an inherent relationship between performance, time, and adherence to the convergence criterion. Adherence to the convergence criterion is less important as time passes; parameter configurations that violate the convergence criterion, but still perform relatively well, are more frequent as



(a) 500 Iterations.

(b) 1000 Iterations.

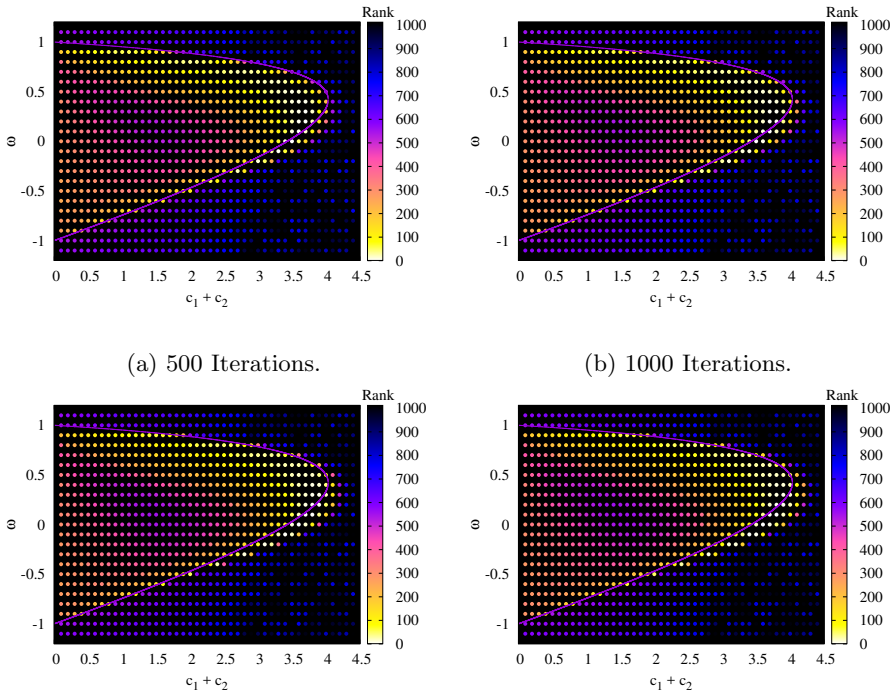
(c) 2500 Iterations.

(d) 5000 Iterations.

Figure 7: Overall rank at various iterations with $c_1 = c_2$.

the search progresses. This is likely a result of the larger variances in particle positions resulting from parameter configurations that are near, or even outside, the convergent region [6]. A large variance corresponds to larger particle step sizes, which means that complete stagnation is less likely. Therefore, parameter configurations with larger variances may be preferred later in the search due to their prevention (or delaying) of stagnation. The observed time-dependence of control parameter values further emphasizes the importance of developing efficient SAPSO algorithms.

Another observation from Figure 10 reinforces that the region containing the best parameters is noticeably different based on the balance between the social and cognitive parameters. For parameter configurations that are considered social, the optimal region forms a cluster that closely resembles the apex



(a) 500 Iterations.

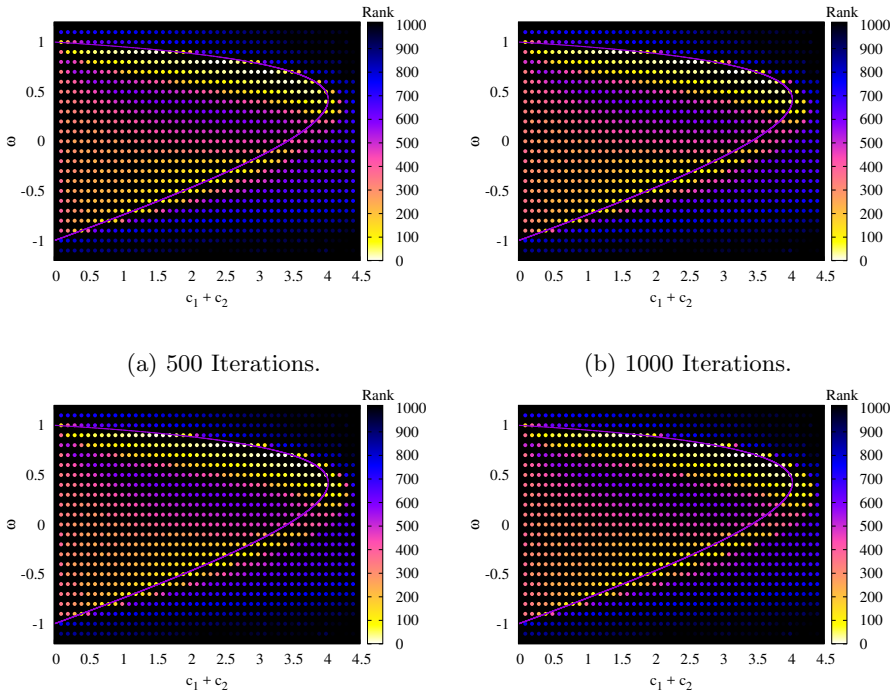
(b) 1000 Iterations.

(c) 2500 Iterations.

(d) 5000 Iterations.

Figure 8: Overall rank at various iterations with a social configuration.

of the convergent region. However, when cognitive parameter configurations are employed, the best parameter configurations form two distinct clusters. Regarding the balanced parameter sets, the region containing the best configuration is somewhat a mix of the two regions described above. For all three parameter schemes, there is a visible preference for positive values of the inertia weight. This observation is least prominent for the social configuration, likely as a result of the increased influence of the global best position. To illustrate this point, consider a cognitive parameter configuration where the movement direction of a particle is most prominently influenced by its own personal best position. Thus, it is reasonable to assume a high degree of correlation between the direction of subsequent particle movements. That is, if a particle is moving in one direction and finds a new personal best solution, it is likely to continue in the same



(a) 500 Iterations.

(b) 1000 Iterations.

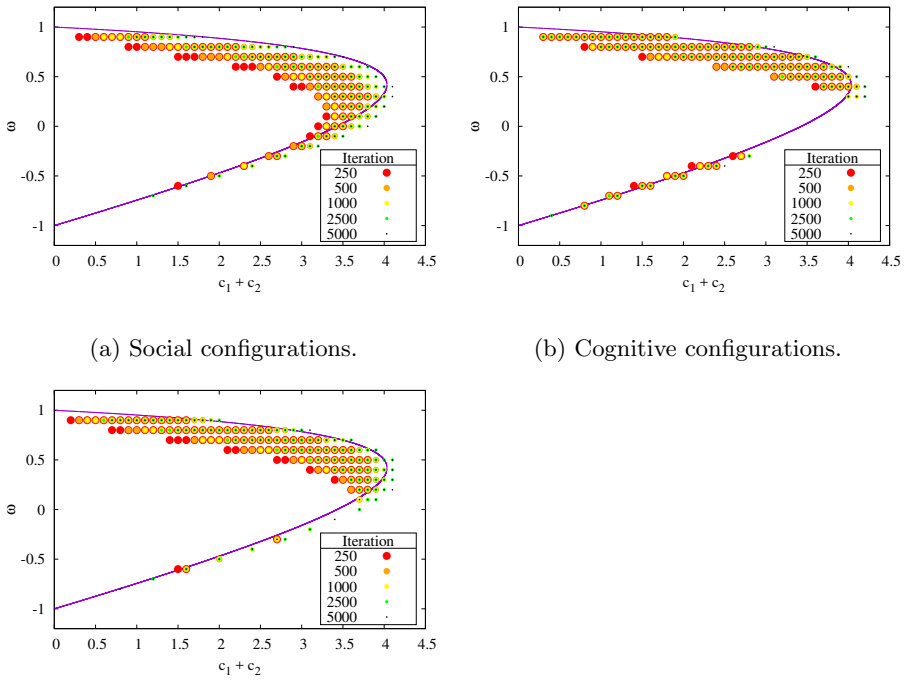
(c) 2500 Iterations.

(d) 5000 Iterations.

Figure 9: Overall rank at various iterations with cognitive configurations.

direction. Therefore, a negative inertia weight would be rather detrimental to a cognitive parameter configuration. Conversely, a social solution takes a high degree of influence from the remainder of the swarm and thus is subject to have less correlation between the direction of subsequent movements.

Figures 11 to 13 show the distribution of the 100 control parameter values that lead to the best performance at various time intervals. Considering the distribution of the inertia weight values, shown in Figure 11, there is little change in the distribution after 1000 iterations, suggesting that the best inertia weight values to employ are within the range $[0.4, 0.8]$ but do not, in general, change as the search progresses. This provides further evidence to suggest that decreasing inertia weight strategies are suboptimal. For each of the iterations examined, the most frequent well-performing cognitive control parameter values, shown



(a) Social configurations.

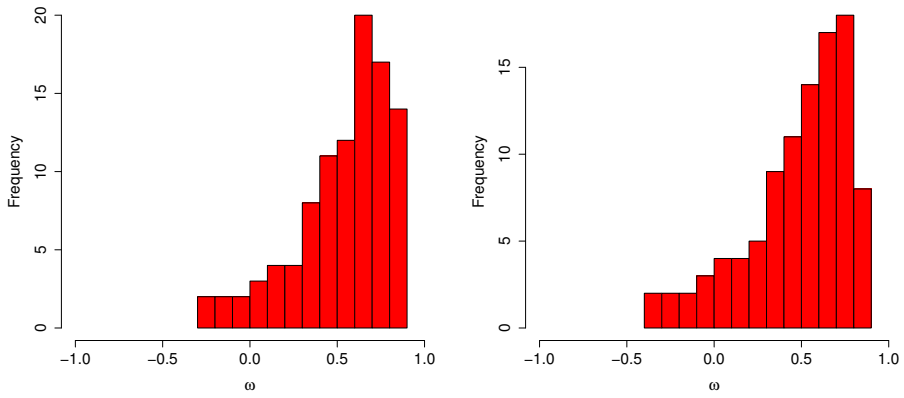
(b) Cognitive configurations.

(c) Equal configurations.

Figure 10: 100 best parameter configurations at various iterations.

in Figure 12, are within the range of $[0.5, 1.0]$. However, the value of the cognitive control parameter that most frequently leads to the best performance shows a slight increase as the search progresses. After 500 iterations, cognitive acceleration values near 0.8 occur most frequently in the 100 best parameter configurations, while the most frequent well-performing cognitive acceleration coefficient increases to approximately 0.9 after 5000 iterations. The distribution of the best-performing social acceleration coefficients, shown in Figure 13, depicts a similar trend to the cognitive acceleration coefficient. After 500 iterations, values of the social acceleration coefficient that are approximately 2.6 lead to good performance most frequently, while a social acceleration value of approximately 2.8 tends to most frequently lead to good performance after 5000 iterations. In general, the best performance is observed when a social acceler-

ation coefficient within the range of $[2.5, 3.0]$ is employed, further suggesting that parameter configurations favouring the social influence tend to perform better than those that favour a cognitive influence. Based on these results, a general guideline for selecting PSO control parameter values, which should lead to reasonable performance regardless of the number of iterations, is to set $\omega \in [0.4, 0.8], c_1 \in [0.5, 1.0]$, and $c_2 \in [2.5, 3.0]$.



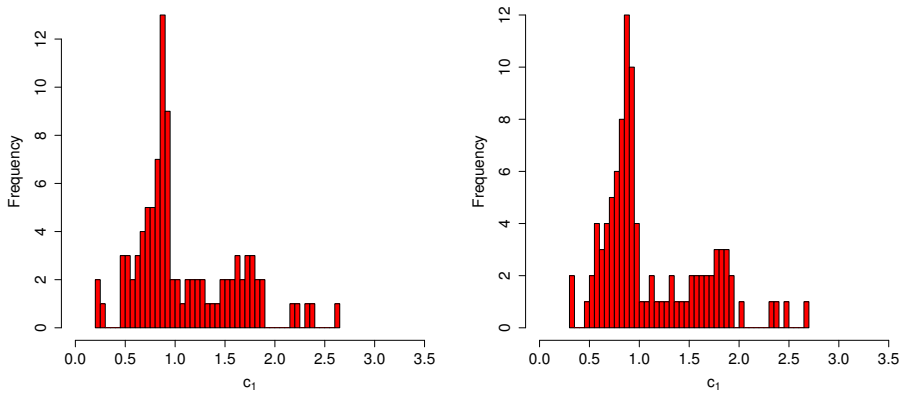
(a) 500 Iterations.

(b) 1000 Iterations.

(c) 2500 Iterations.

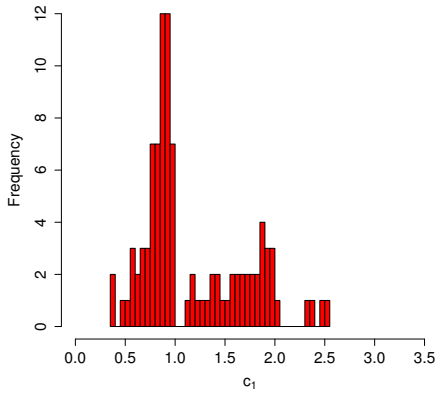
(d) 5000 Iterations.

Figure 11: Distribution of the inertia weight values of the best 100 parameter configurations at various iterations.

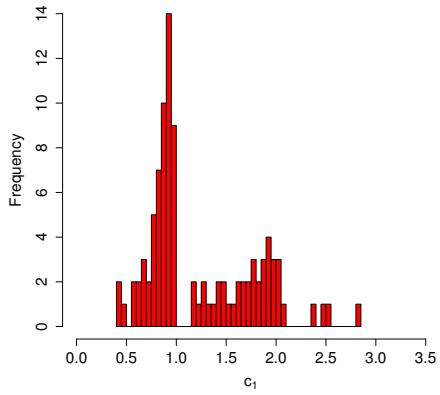


(a) 500 Iterations.

(b) 1000 Iterations.



(c) 2500 Iterations.

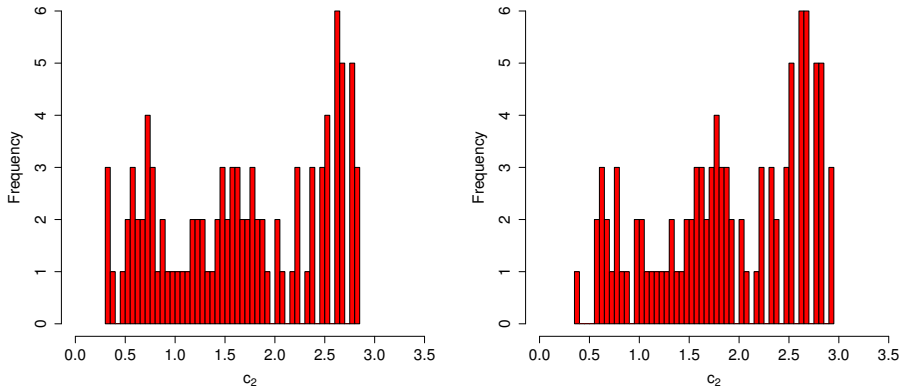


(d) 5000 Iterations.

Figure 12: Distribution of the cognitive acceleration values of the best 100 parameter configurations at various iterations.

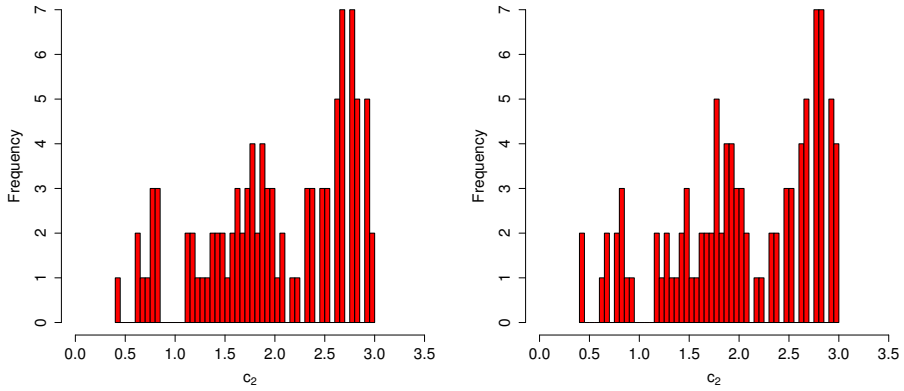
4. Conclusions

405 This work provided an empirical investigation into the relative performance of PSO parameter configurations. The overall objective was to identify the regions of parameter space that lead to the best performance. Specifically, two important questions regarding the parametrization of a global-best PSO were



(a) 500 Iterations.

(b) 1000 Iterations.



(c) 2500 Iterations.

(d) 5000 Iterations.

Figure 13: Distribution of the social acceleration values of the best 100 parameter configurations at various iterations.

addressed in this study. Firstly, the question of where the optimal parameter configurations reside when the respective values of the acceleration coefficients were different was examined. Secondly, this study examined the question of whether the optimal parameters to employ are dependent on time. To investigate these questions, a total of 3036 parameter configurations were examined on a set of 22 benchmark functions. The modality of the benchmark problems

was examined to ascertain whether the optimal regions of parameter space were dependent on the modality of the problem. Furthermore, the results were correlated with the best-known convergence criterion to determine whether particle convergence/stability had any effect on the performance of the PSO algorithm.

To address the first question, the experiments were divided into three categories based on the ratio between the social and cognitive acceleration coefficients (i.e., equal, larger social, and larger cognitive). The results indicated that the regions leading to the best performance were notably different among each of the different types of parameter configurations. That is, the balance between social and cognitive coefficients does have a significant effect on the regions of parameter space that lead to optimal performance. To address the second question, the performance of the parameter configurations was captured at various points throughout the search. Results indicated that the optimal values for the acceleration coefficients increase as the search progresses, irrespective of the balance between the social and cognitive coefficients. Specifically, this indicates that the optimal parameters are, in fact, time-dependent, thereby providing further justification for SAPSO algorithms that can alter the values of their control parameters over time. Despite the observed dependence on time, a general recommendation for selecting values for the PSO control parameters is to set $\omega \in [0.4, 0.8], c_1 \in [0.5, 1.0]$, and $c_2 \in [2.5, 3.0]$.

An immediate avenue of future work lies in the development of new SAPSO algorithms that leverage the information found in this study. Given that this study only examined the modality of the benchmark problems, another avenue of future work is to ascertain the relationship between various landscape characteristics and performance. Landscape analysis may provide further valuable information that can be used in the design of PSO algorithms that adapt to their environment.

5. Acknowledgements

This work is based on the research supported by the National Research Foundation (NRF) of South Africa (Grant Number 46712). The opinions, findings and conclusions or recommendations expressed in this article is that of the author(s) alone, and not that of the NRF. The NRF accepts no liability whatsoever in this regard. This work was also supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] T. Beielstein, Tuning PSO parameters through sensitivity analysis, Tech. Rep., Universitat Dortmund, 2002.

[2] I. C. Trelea, The particle swarm optimization algorithm: convergence analysis and parameter selection, Information Processing Letters 85 (6) (2003) 317–325.

[3] F. van den Bergh, A. P. Engelbrecht, A study of particle swarm optimization particle trajectories, Information Sciences 176 (8) (2006) 937–971.

[4] D. Bratton, J. Kennedy, Defining a Standard for Particle Swarm Optimization, in: 2007 IEEE Swarm Intelligence Symposium, IEEE, 120–127, 2007.

[5] B. J. Leonard, A. P. Engelbrecht, On the optimality of particle swarm parameters in dynamic environments, in: 2013 IEEE Congress on Evolutionary Computation, IEEE, 1564–1569, 2013.

[6] M. Bonyadi, Z. Michalewicz, Impacts of coefficients on movement patterns in the particle swarm optimization algorithm, IEEE Transactions on Evolutionary Computation 21 (3) (2016) 1–1.

[7] A. Carlisle, G. Dozier, An Off-The-Shelf PSO, in: Proceedings of the Workshop on Particle Swarm Optimization, vol. 1, 1–6, 2001.

[8] F. V. D. Bergh, An Analysis of Particle Swarm Optimizers, Ph.D. thesis, University of Pretoria, 2001.

[9] M. Jiang, Y. Luo, S. Yang, Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm, Information Processing Letters 102 (1) (2007) 8–16.

[10] Q. Liu, Order-2 Stability Analysis of Particle Swarm Optimization, Evolutionary Computation 23 (2) (2015) 187–216.

[11] M. Birattari, T. Stützle, L. Paquete, K. Varrentrapp, A Racing Algorithm for Configuring Metaheuristics, in: Proceedings of the Genetic and Evolutionary Computation Conference, ACM, 11–18, 2002.

[12] Y. Shi, R. Eberhart, A modified particle swarm optimizer, in: 1998 IEEE International Conference on Evolutionary Computation Proceedings., IEEE, 69–73, 1998.

[13] Y. Shi, R. Eberhart, Empirical study of particle swarm optimization, in: Proceedings of the 1999 Congress on Evolutionary Computation, vol. 3, IEEE, 1945–1950, 1999.

[14] M. A. Montes de Oca, J. Pena, T. Stutzle, C. Pinciroli, M. Dorigo, Heterogeneous particle swarm optimizers, in: 2009 IEEE Congress on Evolutionary Computation, IEEE, 698–705, 2009.

[15] Y. Wang, B. Li, T. Weise, J. Wang, B. Yuan, Q. Tian, Self-adaptive learning based particle swarm optimization, Information Sciences 181 (20) (2011) 4515–4538.

[16] Changhe Li, Shengxiang Yang, Trung Thanh Nguyen, A Self-Learning Particle Swarm Optimizer for Global Optimization Problems, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 42 (3) (2012) 627–646.

[17] F. V. Nepomuceno, A. P. Engelbrecht, A self-adaptive heterogeneous pso for real-parameter optimization, in: 2013 IEEE Congress on Evolutionary Computation, IEEE, 361–368, 2013.

[18] A. Ratnaweera, S. Halgamuge, H. Watson, Self-Organizing Hierarchical Particle Swarm Optimizer With Time-Varying Acceleration Coefficients, IEEE Transactions on Evolutionary Computation 8 (3) (2004) 240–255.

[19] Z.-H. Zhan, J. Zhang, Y. Li, H. S.-H. Chung, Adaptive Particle Swarm Optimization, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 39 (6) (2009) 1362–1381.

[20] G. Xu, An adaptive parameter tuning of particle swarm optimization algorithm, Applied Mathematics and Computation 219 (9) (2013) 4560–4569.

[21] F. Olivas, F. Valdez, O. Castillo, P. Melin, Dynamic parameter adaptation in particle swarm optimization using interval type-2 fuzzy logic, Soft Computing 20 (3) (2016) 1057–1070.

[22] P. Melin, F. Olivas, O. Castillo, F. Valdez, J. Soria, M. Valdez, Optimal design of fuzzy classification systems using PSO with dynamic parameter adaptation through fuzzy logic, Expert Systems with Applications 40 (8) (2013) 3196–3206.

[23] M. Tanweer, S. Suresh, N. Sundararajan, Self regulating particle swarm optimization algorithm, Information Sciences 294 (2015) 182–202.

[24] E. van Zyl, A. Engelbrecht, Comparison of self-adaptive particle swarm optimizers, in: 2014 IEEE Symposium on Swarm Intelligence, IEEE, 1–9, 2014.

[25] K. R. Harrison, A. P. Engelbrecht, B. M. Ombuki-Berman, Inertia weight control strategies for particle swarm optimization, Swarm Intelligence 10 (4) (2016) 267–305.

[26] K. R. Harrison, A. P. Engelbrecht, B. M. Ombuki-Berman, The sad state of self-adaptive particle swarm optimizers, in: 2016 IEEE Congress on Evolutionary Computation, IEEE, 431–439, 2016.

[27] K. R. Harrison, A. P. Engelbrecht, B. M. Ombuki-Berman, Self-Adaptive Particle Swarm Optimization: A Review and Analysis of Convergence, Swarm Intelligence (In Press) URL https://doi.org/10.1007/s11721-017-0150-9.

[28] C. W. Cleghorn, A. Engelbrecht, Particle swarm optimizer: The impact of unstable particles on performance, in: 2016 IEEE Symposium Series on Computational Intelligence, IEEE, 1–7, 2016.

[29] K. R. Harrison, B. M. Ombuki-Berman, A. P. Engelbrecht, Optimal parameter regions for particle swarm optimization algorithms, in: 2017 IEEE Congress on Evolutionary Computation, IEEE, 349–356, 2017.

[30] R. Poli, D. Broomhead, Exact analysis of the sampling distribution for the canonical particle swarm optimiser and its convergence during stagnation, in: Proceedings of the 9th annual conference on Genetic and evolutionary computation, ACM Press, 134, 2007.

[31] R. Poli, Mean and Variance of the Sampling Distribution of Particle Swarm Optimizers During Stagnation, IEEE Transactions on Evolutionary Computation 13 (4) (2009) 712–721.

[32] J. Kennedy, R. Mendes, Population structure and particle swarm performance, in: Proceedings of the 2002 Congress on Evolutionary Computation., vol. 2, IEEE, 1671–1676, 2002.

[33] A. Engelbrecht, Particle Swarm Optimization: Global Best or Local Best?, in: 2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence, IEEE, 124–135, 2013.

[34] Q. Liu, W. Wei, H. Yuan, Z.-H. Zhan, Y. Li, Topology selection for particle swarm optimization, Information Sciences 363 (2016) 154–173.

[35] V. Kadirkamanathan, K. Selvarajah, P. Fleming, Stability analysis of the particle dynamics in particle swarm optimizer, IEEE Transactions on Evolutionary Computation 10 (3) (2006) 245–255.

[36] V. Gazi, Stochastic stability analysis of the particle dynamics in the PSO algorithm, in: 2012 IEEE International Symposium on Intelligent Control, IEEE, 708–713, 2012.

[37] C. W. Cleghorn, A. P. Engelbrecht, A generalized theoretical deterministic particle swarm model, Swarm Intelligence 8 (1) (2014) 35–59.

[38] M. R. Bonyadi, Z. Michalewicz, Analysis of Stability, Local Convergence, and Transformation Sensitivity of a Variant of the Particle Swarm Optimization Algorithm, IEEE Transactions on Evolutionary Computation 20 (3) (2016) 370–385.

[39] C. W. Cleghorn, A. P. Engelbrecht, Particle swarm convergence: An empirical investigation, in: 2014 IEEE Congress on Evolutionary Computation, IEEE, 2524–2530, 2014.

[40] M. R. Bonyadi, Z. Michalewicz, Stability Analysis of the Particle Swarm Optimization Without Stagnation Assumption, IEEE Transactions on Evolutionary Computation 20 (5) (2016) 814–819.

[41] J. Rada-Vilela, M. Zhang, W. Seah, A performance study on synchronous and asynchronous updates in particle swarm optimization, in: Proceedings of the 13th annual conference on Genetic and evolutionary computation, ACM Press, ISBN 9781450305570, 21, 2011.

[42] A. Engelbrecht, Particle swarm optimization: Velocity initialization, in: 2012 IEEE Congress on Evolutionary Computation, IEEE, 1–8, 2012.

[43] C. W. Cleghorn, A. P. Engelbrecht, Particle swarm stability: a theoretical extension using the non-stagnate distribution assumption, Swarm Intelligence (2017) 1–22.

Appendix A. Benchmark problems

This appendix provides both the equation and feasible domain for each benchmark problem used in this work.

f_1, the absolute value function, defined as

$$f_1(\vec{x}) = \sum_{j=1}^{n_x} |x_j| \tag{A.1}$$

with each $x_j \in [-100, 100]$.

f_2, the ackley function, defined as

$$f_2(\vec{x}) = -20e^{-0.2\sqrt{\frac{1}{n}\sum_{j=1}^{n_x} x_j^2}} - e^{\frac{1}{n}\sum_{j=1}^{n_x}\cos(2\pi x_j)} + 20 + e \tag{A.2}$$

with each $x_j \in [-32.768, 32.768]$.

f_3, the alpine function, defined as

$$f_3(\vec{x}) = \sum_{j=1}^{n_x} |x_j \sin(x_j) + 0.1x_j| \tag{A.3}$$

with each $x_j \in [-10, 10]$.

f_4, the egg holder function, defined as

$$f_4(\vec{x}) = \sum_{j=1}^{n_x-1} \left(-(x_{j+1} + 47) \sin\left(\sqrt{|x_{j+1} + x_j/2 + 47|} \right) + \sin\left(\sqrt{|x_j - (x_{j+1} + 47)|} \right) (-x_j) \right) \tag{A.4}$$

with each $x_j \in [-512, 512]$.

f_5, the elliptic function, defined as

$$f_5(\vec{x}) = \sum_{j=1}^{n_x} (10^6)^{\frac{j-1}{n_x-1}} \tag{A.5}$$

with each $x_j \in [-100, 100]$.

f_6, the griewank function, defined as

$$f_6(\vec{x}) = 1 + \frac{1}{4000}\sum_{j=1}^{n_x} x_j^2 - \prod_{j=1}^{n_x} \cos\left(\frac{x_j}{\sqrt{j}}\right) \tag{A.6}$$

with each $x_j \in [-600, 600]$.

f_7, the hyperellipsoid function, defined as

$$f_7(\vec{x}) = \sum_{j=1}^{n_x} j x_j^2 \tag{A.7}$$

with each $x_j \in [-5.12, 5.12]$.

f_8, the michalewicz function, defined as

$$f_8(\vec{x}) = -\sum_{j=1}^{n_x} \sin(x_j) \left(\sin\left(\frac{j x_j^2}{\pi} \right) \right)^{2m} \tag{A.8}$$

with each $x_j \in [0, \pi]$ and $m = 10$.

f_9, the norwegian function, defined as

$$f_9(\vec{x}) = \prod_{j=1}^{n_x} \left(\cos(\pi x_j^3) \left(\frac{99 + x_j}{100} \right) \right) \tag{A.9}$$

with each $x_j \in [-1.1, 1.1]$.

f_{10}, the quadric function, defined as

$$f_{10}(\vec{x}) = \sum_{i=1}^{n_x} \left(\sum_{j=1}^{i} x_j \right)^2 \tag{A.10}$$

with each $x_j \in [-100, 100]$.

f_{11}, the quartic function, defined as

$$f_{11}(\vec{x}) = \sum_{j=1}^{n_x} j x_j^4 \tag{A.11}$$

with each $x_j \in [-1.28, 1.28]$.

f_{12}, the rastrigin function, defined as

$$f_{12}(\vec{x}) = 10 n_x + \sum_{j=1}^{n_x} (x_j^2 - 10 \cos(2\pi x_j)) \tag{A.12}$$

with each $x_j \in [-5.12, 5.12]$.

f_{13}, the rosenbrock function, defined as

$$f_{13}(\vec{x}) = \sum_{j=1}^{n_x-1} \left(100(x_{j+1} - x_j^2) + (x_j - 1)^2\right) \tag{A.13}$$

with each $x_j \in [-30, 30]$.

f_{14}, the saloman function, defined as

$$f_{14}(\vec{x}) = -\cos\left(2\pi \sum_{j=1}^{n_x} x_j^2\right) + 0.1\sqrt{\sum_{j=1}^{n_x} x_j^2} + 1 \tag{A.14}$$

with each $x_j \in [-100, 100]$.

f_{15}, the generalized schaffer 6 function, also known as the pathological function, defined as

$$f_{15}(\vec{x}) = \sum_{j=1}^{n_x} \left(0.5 + \frac{\sin^2(100x_j^2 + x_{j+1}^2) - 0.5}{1 + 0.001(x_j^2 - 2x_j x_{j+1} + x_{j+1}^2)^2}\right) \tag{A.15}$$

with each $x_j \in [-100, 100]$.

f_{16}, the schwefel 1.2 function, defined as

$$f_{16}(\vec{x}) = \sum_{i=1}^{n_x} \left(\sum_{j=1}^{i} x_j\right)^2 \tag{A.16}$$

with each $x_j \in [-100, 100]$.

f_{17}, the schwefel 2.21 function, defined as

$$f_{17}(\vec{x}) = \max_j \{|x_j|, 1 \leq j \leq n_x\} \tag{A.17}$$

with each $x_j \in [-100, 100]$.

f_{18}, the schwefel 2.22 function, defined as

$$f_{18}(\vec{x}) = \sum_{j=1}^{n_x} |x_j| + \prod_{j=1}^{n_x} |x_j| \tag{A.18}$$

with each $x_j \in [-10, 10]$.

f_{19}, the shubert function, defined as

$$f_{19}(\vec{x}) = \prod_{j=1}^{n_x} \left(\sum_{i=1}^{5} (i \cos((i+1)x_j + i)) \right) \tag{A.19}$$

with each $x_j \in [-10, 10]$.

f_{20}, the spherical function, defined as

$$f_{20}(\vec{x}) = \sum_{j=1}^{n_x} x_j^2 \tag{A.20}$$

with each $x_j \in [-5.12, 5.12]$.

f_{21}, the step function, defined as

$$f_{21}(\vec{x}) = \sum_{j=1}^{n_x} (\lfloor x_j + 0.5 \rfloor)^2 \tag{A.21}$$

with each $x_j \in [-100, 100]$.

f_{22}, the vincent function, defined as

$$f_{22}(\vec{x}) = - \left(1 + \sum_{j=1}^{n_x} \sin(10\sqrt{x_j}) \right) \tag{A.22}$$

with each $x_j \in [0.25, 10]$.