Swarm Intell
DOI 10.1007/s11721-016-0128-z

# Inertia weight control strategies for particle swarm optimization

## Too much momentum, not enough analysis

**Kyle Robert Harrison**[1] · **Andries P. Engelbrecht**[1] · **Beatrice M. Ombuki-Berman**[2]

Received: 27 August 2016 / Accepted: 21 October 2016


**Abstract** Particle swarm optimization (PSO) is a population-based, stochastic optimization technique inspired by the social dynamics of birds. The PSO algorithm is rather sensitive to the control parameters, and thus, there has been a significant amount of research effort devoted to the dynamic adaptation of these parameters. The focus of the adaptive approaches has largely revolved around adapting the inertia weight as it exhibits the clearest relationship with the exploration/exploitation balance of the PSO algorithm. However, despite the significant amount of research efforts, many inertia weight control strategies have not been thoroughly examined analytically nor empirically. Thus, there are a plethora of choices when selecting an inertia weight control strategy, but no study has been comprehensive enough to definitively guide the selection. This paper addresses these issues by first providing an overview of 18 inertia weight control strategies. Secondly, conditions required for the strategies to exhibit convergent behaviour are derived. Finally, the inertia weight control strategies are empirically examined on a suite of 60 benchmark problems. Results of the empirical investigation show that none of the examined strategies, with the exception of a randomly selected inertia weight, even perform on par with a constant inertia weight.

**Keywords** Particle swarm optimization · Inertia weight strategies · Convergence

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

✉ Kyle Robert Harrison
krharrison28@gmail.com

Andries P. Engelbrecht
engel@cs.up.ac.za

Beatrice M. Ombuki-Berman
bombuki@brocku.ca

[1] Department of Computer Science, University of Pretoria, Pretoria, South Africa

[2] Department of Computer Science, Brock University, St. Catharines, Canada

Published online: 02 November 2016

## 1 Introduction

The particle swarm optimization (PSO) algorithm (Kennedy and Eberhart 1995) is a population-based, stochastic search algorithm inspired by the flocking behaviour of birds. The control parameters of the algorithm directly affect the movement of particles and, by extension, the searching capabilities of the algorithm. Thus, the performance of the PSO algorithm is rather sensitive to the control parameters (Beielstein et al. 2002; Carlisle and Dozier 2001; Trelea 2003; Bergh and Engelbrecht 2006). However, the best parameters to employ during a search may be time dependent in the sense that parameters which perform well early in a search, i.e. during exploration, may no longer be preferable during the later, exploitative phases. This was evidenced by Leonard and Engelbrecht (2013) where it was empirically found that parameters well suited for exploration were not well suited for exploitation and vice versa. The time sensitivity of the control parameters is further evidenced by heterogeneous PSO algorithms which have demonstrated that the most suitable velocity update scheme to employ varies during the search (Li et al. 2012; Nepomuceno and Engelbrecht 2013; de Oca et al. 2009; Wang et al. 2011). Similarly, Lynn and Suganthan (2015) found that using separate swarms for exploration and exploitation leads to improved performance.

There has been an extensive amount of research effort devoted to adapting the control parameters of the PSO algorithm during the search. A majority of this research has focused on adapting the inertia weight. While a number of recent studies have provided reviews of existing inertia weight control strategies (Nickabadi et al. 2011; Chauhan et al. 2013; Pandey et al. 2015; Van Zyl and Engelbrecht 2014; Taherkhani and Safabakhsh 2016; Hu et al. 2009; Bonyadi and Michalewicz 2016; Bansal et al. 2011), such studies typically lack in empirical investigation. Moreover, studies which empirically compared multiple inertia weight strategies often only examined a limited set of strategies; Nickabadi et al. (2011) examined five, Chauhan et al. (2013) examined only two, Pandey et al. (2015) examined six, Van Zyl and Engelbrecht (2014) examined six, Taherkhani and Safabakhsh (2016) examined seven, Hu et al. (2009) did not empirically examine any strategies apart from their proposed strategy, Bonyadi and Michalewicz (2016) did not empirically examine any strategies, and Bansal et al. (2011) examined 15 strategies. Note that the study of Bansal et al. (2011) was relatively comprehensive with regard to the inertia weight strategies but was severely limited by the use of only five benchmark problems.

As Taherkhani and Safabakhsh (2016) pointed out, none of the previous works on inertia weight control strategies focused on ensuring that the PSO algorithm retained convergent behaviour. Thus, there is a general tendency for adaptive PSO variants to exhibit divergent behaviour (Harrison et al. 2016; Taherkhani and Safabakhsh 2016; Van Zyl and Engelbrecht 2014). To circumvent divergent behaviour, many authors have employed velocity clamping as a means to limit particle step sizes (Chatterjee and Siarry 2006; Chauhan et al. 2013; Chen et al. 2009; Fan and Chiu 2007; Gao et al. 2008; Liu et al. 2005; Panigrahi et al. 2008; Tanweer et al. 2015; Van Zyl and Engelbrecht 2014). While velocity clamping does not necessarily prevent divergence, it does prevent particles from having overly large step sizes, thereby mitigating the issue. Additionally, velocity clamping may hinder the roaming behaviour of particles, a phenomenon known to be beneficial to the overall search capabilities of a particle swarm optimizer (Engelbrecht 2013b). Most importantly, the need for velocity clamping in an adaptive strategy provides a clear indication that the adaptation mechanism is flawed as overly large step sizes are often a direct result of poor parameter selection. Therefore, if a PSO algorithm with an adaptive parameter strategy requires the use of velocity clamping to prevent divergent behaviour, the control strategy is clearly ineffective at controlling the search.

The field of inertia weight control strategies is thus left with a plethora of strategies to chose from, with no clear guidance on which strategies perform best. There are also no studies which indicate whether or when the various control strategies adhere to the theoretical convergence criterion. Furthermore, none of the previous studies have examined the performance relative to fitness landscape features such as modality and separability. To address these shortcomings, this study examines 18 inertia weight control strategies and dissects them analytically to determine whether and when they will lead to convergent behaviour. Moreover, each of the strategies are empirically investigated on a suite of 60 boundary-constrained benchmark problems encompassing a variety of landscape features and complexities. In all cases, the parameters employed by the algorithms are taken from the literature. As such, this study makes no attempt to tune the parameters of the inertia weight control strategies, but rather reflects the algorithms' performance relative to their recommended parametrizations.

The remainder of this paper is structured as follows. Section 2 provides an overview of the PSO algorithm. Section 3 describes a number of non-adaptive, generally time-dependent inertia weight control strategies, while Sect. 4 introduces the truly adaptive variants which use the algorithmic state to control the inertia weight. The experimental procedure is described in Sect. 5, the results of which are presented in Sect. 6. Finally, the concluding remarks and directions for future work are given in Sect. 7.

## 2 Particle swarm optimization

The PSO algorithm consists of a swarm of agents, referred to as particles, where each particle represents a candidate solution to the optimization problem at hand. The basis of the algorithm is a repeated calculation and subsequent addition of a velocity vector to each particle's position vector, thereby providing movement. The calculation of a particle's velocity is based on its attraction towards two key locations in the search space, namely the best position found by the particle and the best position in the particle's neighbourhood. The neighbourhood topology 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 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. 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 (Engelbrecht 2013a). However, if optimal performance is desired, then 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 (Engelbrecht 2013a; Liu et al. 2016).

For the purposes of this study, a global-best topology is employed (for completeness, "Appendix 2" also provides results for the local-best topology). The velocity is then calculated for particle $i$ according to the inertia weight model of Shi and Eberhart (1998) as

$$v_{ij}(t+1) = \omega v_{ij}(t) + c_1 r_{1j}(t)(y_{ij}(t) - x_{ij}(t)) + c_2 r_{2j}(t)(\hat{y}_j(t) - x_{ij}(t)) \quad (1)$$

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 $\omega$, while $c_1$ and $c_2$ represent the cognitive and social coefficients, respectively. The stochastic component of the algorithm is provided by the random constants, $r_{1j}(t), r_{2j}(t) \sim U(0, 1)$. 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

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

### 2.1 Convergence of particle swarm optimization

Notice that the calculation of the velocity in Eq. (1) results in an unbounded vector and provides no limit on the step sizes of particles. Furthermore, the step sizes of particles are inherently dependent upon the values of the control parameters employed. Therefore, if the control parameters are not selected with caution, particle velocities may be extremely large and can lead to divergent behaviour.

According to the theoretical analyses of Poli (2009) and Poli and Broomhead (2007), control parameters which adhere to

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

will lead to convergent behaviour in the PSO algorithm. Alternatively, Eq. (3) can be formulated as a condition on $\omega$ as

$$\frac{5C - g(C)}{48} < \omega < \frac{5C + g(C)}{48} \tag{4a}$$

with

$$C = c_1 + c_2 \text{ and } g(C) = \sqrt{25C^2 - 672C + 2304}. \tag{4b}$$

The region defined by Eq. (3), as depicted in Fig. 1, has been empirically demonstrated to be the most accurate of the various proposed convergence criteria (Cleghorn and Engelbrecht 2014a, 2015) and was used in this work to analyse the convergence behaviour of the examined inertia weight strategies. Despite the influence on performance, the topology does not meaningfully influence the convergent region (Cleghorn and Engelbrecht 2014b).

For all examined inertia weight control strategies in this study, social and cognitive control parameters which have been demonstrated to lead to convergent behaviour (Bergh and Engelbrecht 2006), namely $c_1 = c_2 = 1.496180$, were used. It then follows from Eq. (4) that

$$-0.16199 < \omega(t) < 0.78540 \tag{5}$$

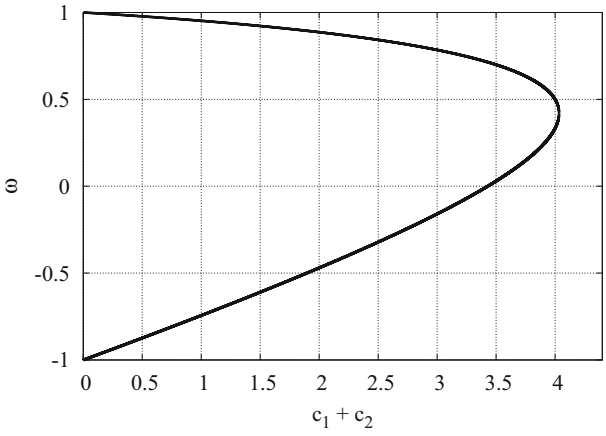


**Fig. 1** Visualization of Poli's theoretically derived region for convergent behaviour of PSO parameters. Parameters which lie within the parabolic region lead to convergent behaviour

is necessary for the algorithms to exhibit convergent behaviour. Eq. (5) was thus used as the basis of the convergence analysis in this study.

### 2.2 Velocity clamping

As previously mentioned, haphazard selection of parameter values may lead to divergent behaviour due to excessively large particle movements. To mitigate the issue of divergence, many authors, rather than selecting parameters known to exhibit convergent behaviour, have employed a technique known as velocity clamping (Eberhart and Kennedy 1995). Velocity clamping limits the step sizes of particles by providing a limit on the component sizes of the velocity vector. Particles are thus provided with an upper limit on the movement along each dimension, thereby preventing overly large step sizes. To implement velocity clamping, Eq. (1) above is amended to include

$$v_{ij}(t+1) = \begin{cases} -v_{\max} & \text{if } v_{ij}(t+1) < -v_{\max} \\ v_{\max} & \text{if } v_{ij}(t+1) > v_{\max} \\ v_{ij}(t+1) & \text{otherwise} \end{cases} \tag{6}$$

where $v_{max}$ is the largest allowable step size in any dimension. Commonly, $v_{\max}$ is set based on the size of the search space, e.g. $v_{\max} = \frac{x_{\max} - x_{\min}}{2}$, where $x_{\min}$ and $x_{\max}$ are the bounds of the search space in each dimension.

## 3 Non-adaptive and time-varying inertia control strategies

This section introduces a number of control strategies which vary the inertia weight without the use of introspective observation. That is, the strategies do not attempt to use the algorithmic state to adapt the inertia weight. These strategies often rely solely on the number of iterations which have passed to vary the inertia weight.

For the remainder of this section, symbols have the following meaning: $t$ is the current iteration, $\omega_{\min}$ is the minimum allowable inertia weight, and $\omega_{\max}$ is the maximum allowable inertia weight.

### 3.1 Particle swarm optimization with constant inertia weight

The inertia weight model of PSO by Shi and Eberhart (1998) employs a static inertia weight throughout the entirety of the search. For the purposes of this study, a constant inertia weight of $\omega = 0.729844$ was used based on the demonstrated ability to lead to convergent trajectories (Cleghorn and Engelbrecht 2014a; Eberhart and Shi 2000; Bergh and Engelbrecht 2006). Note that the employed value of $\omega$ trivially satisfies Eq. (5) and will thus lead to convergent behaviour.

### 3.2 Particle swarm optimization with random inertia weight

The particle swarm optimization with random inertia weight (PSO-RIW) by Eberhart and Shi (2001) randomly selects the inertia weight at each iteration according to

$$\omega(t) = 0.5 + \frac{r(t)}{2} \tag{7}$$

where $r(t) \sim U(0, 1)$. Effectively, this strategy samples the inertia weight at each iteration according to $\omega(t) \sim U(0.5, 1.0)$.

Following from Eq. (5), the PSO-RIW algorithm will exhibit convergent behaviour when $U(0.5, 1.0) < 0.78540$, which will, in theory, occur 57.08% of the time. Given that the PSO-RIW will employ convergent parameters more often than not, it is expected to exhibit convergent behaviour.

### 3.3 Particle swarm optimization with linearly decreasing inertia weight

The particle swarm optimization with linearly decreasing inertia weight (PSO-LDIW) by Shi and Eberhart (1998, 1999) was proposed as a method to linearly decrease the inertia weight over time based on the general consensus that exploration is favoured early in a search process while exploitation is favoured later. To this end, the inertia weight is defined at each iteration by

$$\omega(t) = \omega_{\max} - (\omega_{\max} - \omega_{\min})\frac{t}{T} \tag{8}$$

which linearly decreases the inertia weight from $\omega_{\max}$ at the beginning of the search to $\omega_{\min}$ at the end of the search. The parameters for the PSO-LDIW algorithm were set to $\omega_{\max} = 0.9$ and $\omega_{\min} = 0.4$ as employed by Shi and Eberhart (1999).

Given that $\forall t : \omega(t) \geq 0.4$, Eq. (5) simplifies to $\omega(t) < 0.78540$. From Eq. (8), it follows that $\omega(t) = 0.78540$ occurs when $\frac{t}{T} = 0.22920$, i.e. when 22.920% of the search process has completed. Thus, the PSO-LDIW algorithm is not expected to exhibit convergent behaviour until nearly a quarter of the search process has completed, but will exhibit convergent behaviour overall.

### 3.4 Particle swarm optimization with nonlinear inertia coefficient

Yang et al. (2015) posited that a nonlinear, time-varying inertia weight would demonstrate superior performance over the linearly decreasing PSO-LDIW variant and thus proposed the PSO-NL algorithm. To this end, the nonlinear inertia weight at time $t$ is given by

$$\omega(t) = \omega_{\max} - (\omega_{\max} - \omega_{\min})\left(\frac{t}{T}\right)^{\alpha} \tag{9}$$

where $\alpha$ is a user-supplied constant. The authors empirically suggested the use of $\alpha = 1/\pi^2$, $\omega_{\min} = 0.4$, and $\omega_{\max} = 0.9$. The resulting inertia weight over time is visualized in Fig. 2.

Despite the initial value of 0.9, the inertia weight rapidly decreases during the beginning of the search and thus $\omega(t) < 0.78540$ occurs after $\frac{t}{T} = 4.8481 \times 10^{-7}$. Therefore, the PSO-NL algorithm will exhibit convergent behaviour.

### 3.5 Nonlinear improved particle swarm optimization

The particle swarm optimization with nonlinear improved inertia weight (PSO-NLI) algorithm by Jiao et al. (2008) uses a nonlinear decreasing inertia weight defined by

$$\omega(t) = \omega u^{-t} \tag{10}$$

where $\omega \in [0, 1]$ and $u \in [1.0001, 1.005]$.

Jiao et al. (2008) employed parameters of $u = 1.0002$ and $\omega = 0.3$. Given that $\forall t : 0 < \omega(t) \leq 0.3$, the algorithm will always exhibit convergent behaviour. However, because of the very low inertia weight the PSO-NLI is expected to exhibit premature convergence.

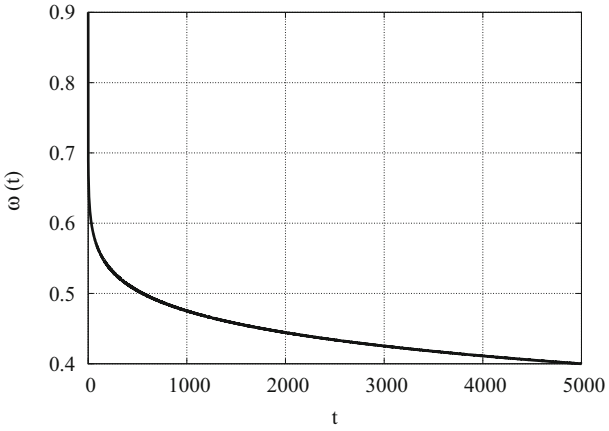


**Fig. 2** Inertia weight value in the nonlinear inertia weight strategy of Yang et al. (2015)

### 3.6 Decreasing inertia weight particle swarm optimization

The decreasing inertia weight particle swarm optimization (DW-PSO) algorithm by Fan and Chiu (2007) employs a time-dependent inertia weight strategy according to

$$\omega(t) = \left(\frac{2}{t}\right)^{0.3}, \tag{11}$$

such that the inertia weight decreases over time in a nonlinear fashion.

The range of the inertia weight for the DW-PSO algorithm is then given by $0 < \omega(t) < 1.23114$. Furthermore, $\omega(t) < 0.78540$ occurs when $t = 4.47432$ due to the rapidly decreasing inertia weight. Therefore, the DW-PSO will exhibit overall convergent behaviour after only five iterations have passed.

### 3.7 Chaotic descending inertia weight particle swarm optimization

The chaotic descending inertia weight particle swarm optimization (CDIWPSO) algorithm by Feng et al. (2007) adopts the use of chaotic dynamics to adapt the inertia weight over time according to

$$\omega(t) = z(t)\omega_{\min} + (\omega_{\max} - \omega_{\min})\frac{T - t}{T} \tag{12}$$

where $z(t)$ is the value of the logistic map

$$z(t+1) = 4z(t)(1 - z(t)) \tag{13}$$

with $z(0) \sim U(0, 1)$. The authors used $\omega_{\max} = 0.9$ and $\omega_{\min} = 0.4$.

The CDIW-PSO will then exhibit convergent behaviour when

$$z(t) < \frac{t + 2854}{4000}$$

assuming $T = 5000$. Note that the maximum value of $z(t)$ is 1.0 and that

$$\frac{t + 2854}{4000} = 1.0$$

when $t = 1146$, i.e. when $\frac{t}{T} = 0.22920$. Thus, if $z(t)$ is removed the convergence behaviour of the CDIW-PSO is the same as that of the PSO-LDIW algorithm. However, the convergence of the CDIW-PSO algorithm during the initial 22.92% of the search is dependent upon on the value of $z(t)$, whereas the PSO-LDIW will always be non-convergent during this initial phase. Based on the above analysis, the CDIW-PSO will exhibit overall convergent behaviour.

### 3.8 Particle swarm optimization with natural exponent inertia weight

The particle swarm optimization with natural exponent inertia weight (PSONEIW) algorithm by Chen et al. (2006) uses a decreasing inertia weight based on the exponential function according to

$$\omega(t) = \omega_{\min} + (\omega_{\max} - \omega_{\min})e^{-\frac{10t}{T}}. \tag{14}$$

The authors suggested $\omega_{\min} = 0.4$ and $\omega_{\max} = 0.9$.

The PSO-NEIW algorithm will exhibit convergent behaviour when $\frac{t}{T} > 0.02603$, i.e. after approximately 2.6% of the search has completed. Thus, the PSO-NEIW algorithm will exhibit convergent behaviour.

### 3.9 Particle swarm optimization with oscillating inertia weight

The particle swarm optimization with oscillating inertia weight (PSO-OIW) algorithm by Kentzoglanakis and Poole (2009) was proposed as an inertia weight control strategy which did not monotonically decrease the inertia weight, but rather provided an oscillating inertia weight during the course of execution. The PSO-OIW employs a sinusoidal inertia weight strategy controlled by

$$\omega(t) = \begin{cases} \frac{\omega_{\min}+\omega_{\max}}{2} + \frac{\omega_{\max}-\omega_{\min}}{2}\cos\left(\frac{2\pi t(4k+6)}{3T}\right) & \text{if } t < \frac{3T}{4} \\ \omega_{\min} & \text{otherwise} \end{cases} \tag{15}$$

The authors employed values of $k = 7$, $\omega_{\min} = 0.3$, and $\omega_{\max} = 0.9$ which then simplifies Eq. (15) to

$$\omega(t) = \begin{cases} 0.6 + 0.3\cos\left(\frac{17\pi t}{3750}\right) & \text{if } t < \frac{3T}{4} \\ 0.3 & \text{otherwise} \end{cases} \tag{16}$$

Kentzoglanakis and Poole (2009) claimed that the sinusoidal wave specified by Eq. (15), and consequently Eq. (16), should complete $k + \frac{3}{2}$ cycles within a single PSO execution. The oscillating inertia weight of PSO-OIW is visualized in Fig. 3.

By rearranging Eq. (16), it can be shown that the PSO-OIW algorithm will exhibit convergent behaviour when

$$\cos\left(\frac{17\pi t}{3750}\right) < 0.61800$$

which, assuming $T = 5000$, accounts for 2670 (71.2%) of the 3750 iterations in which the algorithms oscillates. Additionally, the final quarter of the search process employs an inertia weight of 0.3, which leads to trivially convergent behaviour. Thus, the algorithm is expected to exhibit convergent behaviour.

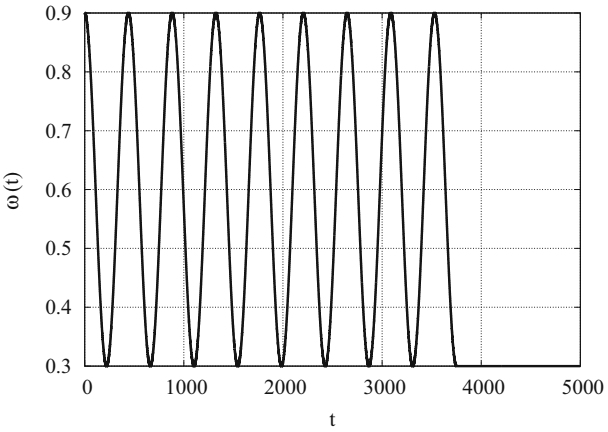


**Fig. 3** Oscillating inertia weight of Kentzoglanakis and Poole (2009)

### 3.10 Particle swarm optimization with Sugeno inertia weight

The particle swarm optimization with Sugeno inertia weight (PSO-SIW) algorithm by Lei et al. (2006) employs the use of a Sugeno function to control the inertia weight over time. This strategy provides a monotonically decreasing inertia weight according to

$$\omega(t) = \frac{1 - \beta(t)}{1 + s\beta(t)} \tag{17}$$

where $\beta(t) = \frac{t}{T}$ and $s > -1$ is a constant which controls the shape of the curve controlling the inertia weight. The inertia weight will thus decrease from an initial value of 1 to a final value of 0. Additionally, when $s < 0$, the inertia weight follows a convex curve while $s > 0$ leads to a concave curve; when $s = 0$, the inertia weight linearly decreases.

The PSO-SIW strategy will exhibit convergent behaviour when

$$\frac{t}{T} > \frac{1073}{2927s + 5000}$$

and thus the convergence depends largely on the employed value of $s$. For the purposes of this study, a value of $s = 2$ was employed and thus the algorithm will exhibit convergent behaviour when

$$\frac{t}{T} > \frac{1073}{10854} = 0.09886,$$

i.e. after approximately 10% of the search has completed. Despite the initially non-convergent behaviour due to the initial inertia weight of 1, the PSO-SIW algorithm is expected to exhibit convergent behaviour.

### 3.11 Logarithm decreasing inertia weight particle swarm optimization

The logarithm decreasing particle swarm optimization (LD-PSO) algorithm by Gao et al. (2008) employs a logarithmically decreasing inertia weight according to

$$\omega(t) = \omega_{\max} + (\omega_{\min} - \omega_{\max}) \log_{10}\left(a + \frac{10t}{T}\right) \tag{18}$$

where $a$ is a user-supplied constant which controls the convergence speed.

The authors used $\omega_{\min} = 0.4$ and $\omega_{\max} = 0.9$ and further suggested the use of $a = 1$. Given these parameters, the LD-PSO algorithm will exhibit convergent behaviour when $\frac{t}{T} > 0.02576$, i.e. after only approximately 2.6% of the search has completed. Therefore, the LD-PSO algorithm will exhibit convergent behaviour.

## 4 Adaptive inertia weight control strategies

In contrast to the non-adaptive inertia weight control strategies presented in the previous section, adaptive strategies control the inertia weight dynamically using introspective observation. These control strategies rely on the use of various characterizations of the algorithmic state, such as average velocity or distance to the global best, as a means to adapt the inertia weight.

### 4.1 Self-regulating particle swarm optimization

The self-regulating particle swarm optimization (SRPSO) algorithm by Tanweer et al. (2015) is premised on regulating the inertia weight of each particle such that the inertia is increased for the best particle while being decreased for all other particles. The justification for this behaviour was that the best particle should have a high level of confidence in its direction and thus accelerate quicker, while the remainder of particles should follow a linearly decreasing inertia weight strategy similar to that of PSO-LDIW (Tanweer et al. 2015). The self-regulating inertia weight is given by

$$\omega_i(t) = \begin{cases} \omega_i(t-1) + \eta\Delta\omega & \text{for the best particle} \\ \omega_i(t-1) - \Delta\omega & \text{for all other particles} \end{cases} \tag{19}$$

where $\eta$ is a constant to control the rate of acceleration and

$$\Delta\omega = \frac{\omega_s - \omega_f}{T} \tag{20}$$

where $\omega_s$ and $\omega_f$ are the initial and final values of the inertia weight, respectively.

While the authors employed parameters of $\omega_s = 1.05, \omega_f = 0.5$, and $\eta = 1$ in their study, such parameters lead to non-convergent behaviour and thus are not used in this study. Moreover, the authors clamped the velocity to 6.708% of the feasible range (Tanweer et al. 2015), suggesting that there were issues observed with overly large particle movements. In this study, the parameters $\omega_s = 0.9$ and $\omega_f = 0.4$ were used while the $\eta$ parameter remained unchanged (i.e. $\eta = 1$).

Given the aforementioned parameters, Eq. (20) simplifies to $\Delta\omega = 0.0001$. Note that in Eq. (19), the inertia weight increases for the best particle but decreases for all other particles. To continue with the analysis, the assumption that each particle is equally likely to be the best particle, and thereby increases their inertia weight accordingly, is made. Thus, given a swarm size of $n_s$ particles, the inertia weight of each particle will exhibit an inertia weight decrease of $0.0001(n_s - 2)$ every $n_s$ iterations or $\frac{0.0001*(n_s-2)}{n_s}$ each iteration, on average. Substitution of the resulting inertia weights in Eq. (5) along with an assumed swarm size of 30 particles leads to the necessary condition of

$$1227.86 < t < 11378.50$$

for convergent behaviour. This indicates that the SRPSO algorithm should exhibit convergent behaviour starting at iteration 1228, i.e. after approximately 24.6% of the search. Note that if the SRPSO algorithm is employed for longer than 11378 iterations without leading to complete stagnation, it will once again exhibit divergent behaviour as the inertia weight will decrease under −0.16199.

### 4.2 Adaptive particle swarm optimization based on velocity information

The adaptive parameter tuning of particle swarm optimization based on velocity information (APSO-VI) algorithm by Xu (2013) adapts the inertia weight based on the current velocities of the particles, with the intention of pushing the velocity closer to an 'ideal' velocity. The average velocity of the swarm is calculated as

$$\overline{v(t)} = \frac{1}{n_{\mathrm{d}} n_{\mathrm{s}}} \sum_{i=1}^{n_{\mathrm{s}}} \sum_{j=1}^{n_{\mathrm{d}}} |v_{ij}(t)| \tag{21}$$

where $n_{\mathrm{d}}$ and $n_{\mathrm{s}}$ represent the number of problem dimensions and the size of the swarm, respectively. An equation defining the 'ideal' (average) velocity was proposed whereby the ideal velocity decreases over time, leading to heightened exploitation nearer to the end of the search. Furthermore, Xu (2013) posited that such an ideal velocity will be nonlinear, having long exploratory and exploitative phases with a minimal transition period. The ideal velocity over time is defined by

$$v_{\mathrm{ideal}}(t) = v_s \left( \frac{1 + \cos\left(\pi \frac{t}{T_{0.95}}\right)}{2} \right) \tag{22}$$

where $v_s$ is the initial ideal velocity, given by $\frac{x_{\max} - x_{\min}}{2}$, and $T_{0.95}$ is the point at which 95% of the search is complete. The ideal velocity specified by Eq. (22) is visualized in Fig. 4.

The APSO-VI algorithm then dynamically adapts the inertia weight each iteration based on the average velocity in relation to the ideal velocity as

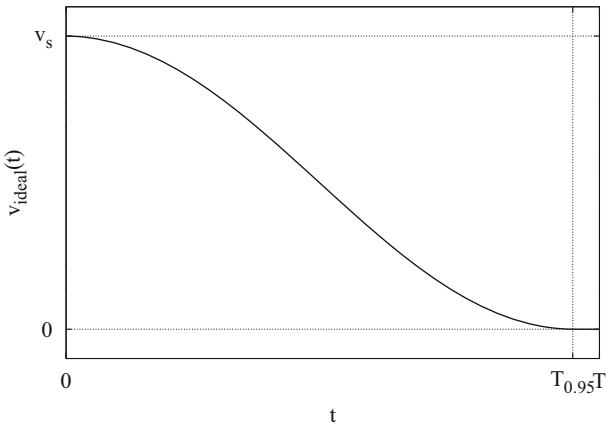


**Fig. 4** Ideal velocity of APSO-VI (Xu 2013) as a function of time

$$\omega(t+1) = \begin{cases} \max\{\omega(t) - \Delta\omega, \omega_{\min}\} & \text{if } \overline{v(t)} \geq v_{\text{ideal}}(t+1) \\ \min\{\omega(t) + \Delta\omega, \omega_{\max}\} & \text{if } \overline{v(t)} < v_{\text{ideal}}(t+1) \end{cases} \tag{23}$$

where $\Delta\omega$ is the step size of the inertia weight.

The authors used $\omega_{\min} = 0.3$ and $\omega_{\max} = 0.9$. As the ideal velocity decreases over time, the employed value of $\omega(t)$ will also decrease over time and thus will eventually stabilize within the convergent range. Furthermore, given that the ideal velocity tends towards 0, by definition the average velocity will also tend towards 0 regardless of the inertia weight. Therefore, the APSO-VI algorithm is expected to exhibit convergent behaviour.

### 4.3 Fine-grained inertia weight particle swarm optimization

The fine-grained inertia weight particle swarm optimization (FG-PSO) algorithm (Deep et al. 2011; Chauhan et al. 2013) provides individualized inertia weights for each particle given by

$$\omega_i(t+1) = \omega_i(t) - \left((\omega_i(t) - 0.4)e^{-(d(\hat{\mathbf{y}}(t), \mathbf{y}_i(t)) * \frac{t}{T})}\right) \tag{24}$$

where $d$ is the Euclidean distance function. Additionally, the inertia weight of each particle is initialized to 0.9.

Firstly, it is recognized that the distance term will always be positive. As a result,

$$0 < e^{-(d(\hat{\mathbf{y}}(t), \mathbf{y}_i(t)) * \frac{t}{T})} \leq 1$$

and using the substitution

$$c = e^{-(d(\hat{\mathbf{y}}(t), \mathbf{y}_i(t)) * \frac{t}{T})},$$

Equation (24) can be expressed as

$$\omega_i(t+1) = \omega_i(t) - c(\omega_i(t) - 0.4)$$

where $c \in (0, 1]$. Therefore, as the distance between a particle and the global best decreases, the inertia weight will also decrease to a minimum of 0.4. An important note about the FG-PSO strategy is that the inertia weight never increases, and thus the inertia weight will always tend towards 0.4. This behaviour will likely cause the FG-PSO strategy to exhibit convergent behaviour in the long run, as even large distances from the global best will cause a nonzero decrease in the inertia weight given the asymptotic nature of the exponential term. However, the inertia weight is expected to decrease rather quickly, especially for the global-best particle (which will have a distance of 0), and therefore, the FG-PSO strategy is expected to suffer from premature convergence.

### 4.4 Double exponential self-adaptive inertia weight particle swarm optimization

The double exponential self-adaptive inertia weight particle swarm optimization (DE-PSO) algorithm by Chauhan et al. (2013) incorporates a double exponential function, referred to as a 'Gompertz function', to select the inertia weight according to

$$\omega_i(t) = e^{-e^{-R_i(t)}} \tag{25a}$$

with

$$R_i(t) = d(\hat{\mathbf{y}}(t), \mathbf{y}_i(t)) \left(\frac{T-t}{T}\right) \tag{25b}$$

where $d$ is the Euclidean distance function. The inertia weight of each particle is initialized to 0.9.

The first observation about the DE-PSO strategy is that $R_i(t)$ must be positive. Given that $R_i(t) > 0$, it follows that $0 < e^{-R_i(t)} \leq 1$ and, by using the substitution $c = e^{-R_i(t)}$, Eq. (25a) can be expressed as

$$\omega_i(t) = e^{-c}$$

with $c \in (0, 1]$. This now provides a definite range for the inertia weight given by

$$0.36788 < \omega_i(t) < 1 \tag{26}$$

where $\omega_i(t)$ is minimized when $R_i(t)$ is 0. Conversely, the DE-PSO algorithm provides larger inertia weights when the distance from the global best is larger. The inertia weight provided by the DE-PSO algorithm is thus expected to be large initially due to the roaming behaviour of unconstrained particles (Engelbrecht 2013b), but is expected to decrease over time as particles tend towards the global best. Therefore, the DE-PSO algorithm will exhibit convergent behaviour.

### 4.5 Particle swarm optimization with rank-based inertia weight

Panigrahi et al. (2008) claimed that the PSO algorithm should be redefined such that the movement of the swarm is controlled by the objective function. To this end, the particle swarm optimization with rank-based inertia weight (PSO-RBI) algorithm (Panigrahi et al. 2008) calculates the inertia weight of each particle based on the rank of its fitness relative to the remainder of the swarm. The inertia weight of each particle is given by

$$\omega_i(t) = \omega_{\min} + (\omega_{\max} - \omega_{\min}) \frac{R_i(t)}{n_s} \tag{27}$$

where $R_i(t)$ is the fitness rank of particle $i$. From Eq. (27), it can be seen that the best-fit particle (i.e. rank 1) will be assigned the lowest inertia weight while the worst fit particle will be assigned the maximal inertia weight.

Parameters of $\omega_{\min} = 0.4$ and $\omega_{\max} = 0.9$ were employed by Panigrahi et al. (2008). Assuming a swarm size of 30 particles, it follows from Eq. (27) that the PSO-RBI algorithm will exhibit convergent behaviour when $\frac{R_i(t)}{n_s} \geq 0.22920$. That is, when a particle is ranked within the worst 22.920% of the swarm, the particle will exhibit divergent behaviour. Therefore, more than one-fifth of the swarm will exhibit divergent behaviour at any given time, causing the PSO-RBI to exhibit overall divergent behaviour.

### 4.6 Adaptive inertia weight particle swarm optimization

The adaptive inertia weight particle swarm optimization (AIWPSO) algorithm by Nickabadi et al. (2011) uses the 'success' rate of the swarm as feedback to adapt the inertia weight. The success rate of the swarm at time $t$ is defined as the proportion of particles which improved their personal best during iteration $t$. Specifically, the inertia weight is adapted according to

$$\omega(t) = \omega_{\min} + (\omega_{\max} - \omega_{\min}) P_s(t) \tag{28a}$$

with

$$P_s(t) = \frac{\sum_{i=1}^{n_s} S_i(t)}{n_s} \tag{28b}$$

and

$$S_i(t) = \begin{cases} 1 & \text{if } f(\mathbf{y}_i(t)) < f(\mathbf{y}_i(t-1)) \\ 0 & \text{otherwise.} \end{cases} \tag{28c}$$

such that the inertia weight is increased when the particles are demonstrating a high rate of success.

Typically, higher success rates are attained early in a search when the fitness of particles are rapidly increasing and therefore the inertia weight of the AIWPSO algorithm will be relatively large initially but will decrease over time.

Given the authors' employed values of $\omega_{\min} = 0.0$ and $\omega_{\max} = 1.0$, Eq. (28a) simplifies to

$$\omega(t) = P_s(t) \tag{29}$$

and therefore, $\omega(t) < 0.78540$ occurs when $P_s(t) < 0.78540$. This behaviour indicates that the swarm may not be convergent when more than 78.54% of the particles improve their personal best fitness in any given iteration. However, it is unrealistic to sustain such a high success rate for any extended period of time due to the stochastic, exploratory nature of particle movement, which effectively prevents particles from always moving in the optimal direction. Thus, the AIWPSO algorithm is expected to exhibit convergent behaviour.

### 4.7 Improved particle swarm optimization

The improved particle swarm optimization algorithm by Li and Tan (2008) (IPSOLT) is based on the idea that the inertia weight should be in direct relation to the convergence factor,

$$c_i(t) = \frac{|f(\mathbf{y}_i(t-1)) - f(\mathbf{y}_i(t))|}{f(\mathbf{y}_i(t-1)) + f(\mathbf{y}_i(t))}, \tag{30}$$

as well as the diffusion factor,

$$d_i(t) = \frac{|f(\mathbf{y}_i(t)) - f(\hat{\mathbf{y}}_i(t))|}{f(\mathbf{y}_i(t)) + f(\hat{\mathbf{y}}_i(t))}, \tag{31}$$

which characterize the state of the algorithm. The inertia weight of each particle is then controlled by

$$\omega_i(t) = 1 - \left| \frac{\alpha(1 - c_i(t))}{(1 + d_i(t))(1 + \beta)} \right| \tag{32}$$

where $\alpha, \beta \in [0, 1]$ are user-supplied constants.

Li and Tan (2008) provided no guidance for the selection of the $\alpha$ and $\beta$ parameter values, and thus, the mid-point of the allowable range is used in this study, namely $\alpha = \beta = 0.5$. Substituting $\alpha = \beta = 0.5$ in Eq. (32) provides a simpler equation for the inertia weight, given by

$$\omega_i(t) = 1 - \left| \frac{1 - c_i(t)}{3(d_i(t) + 1)} \right| \tag{33}$$

which further simplifies the convergence criterion such that

$$\left| \frac{1 - c_i(t)}{d_i(t) + 1} \right| > 0.64380 \tag{34}$$

is required for convergent behaviour to be exhibited. Alternatively, Eq. (34) can be reformulated as

$$\begin{cases} 1.55328c - 2.55328 < d < -1.55328c + 0.55328, & \text{if } c < 1 \\ -1.55328c + 0.55328 < d < 1.55328c - 2.55328, & \text{if } c > 1 \end{cases} \tag{35}$$

to provide a more direct view of the relationship between $c$ and $d$. As the search progresses, it is expected that both $c_i(t)$ and $d_i(t)$ will tend towards 0 and thus the IPSO-LT algorithm should exhibit overall convergent behaviour.

## 5 Experimental procedure

To examine the performance of each of the strategies reviewed in Sects. 3 and 4, 50 independent runs were executed for each strategy on each benchmark problem (see Sect. 5.1) using the parameters listed in Table 1. Note that the parameters employed in this section were identical to those used during the derivation of convergence conditions. Each execution consisted of 5000 iterations and all strategies used an identical base PSO configuration, with the obvious exception of the inertia weight control strategy. The base configuration consisted of 30 particles arranged in a global-best (star) topology using synchronous updates. Additionally, "Appendix 2" presents a summary of the results obtained using a local-best (ring) topology. Particle positions were randomly initialized within the feasible range, while velocities were initialized to zero (Engelbrecht 2012). To ensure that the particle attractors remained inside the feasible region, personal best positions were only updated if a new, superior-fit position was found that satisfied the boundary constraints. As with the convergence analysis, all algo-

**Table 1** Parameters employed by the inertia weight control strategies

| Strategy | Equation | Parameters |
|---|---|---|
| AIWPSO | (28) | $\omega_{\min} = 0.0, \omega_{\max} = 1.0$ |
| APSO-VI | (23) | $\omega_{\min} = 0.3, \omega_{\max} = 0.9$ |
| CDIW-PSO | (12) | $\omega_{\min} = 0.4, \omega_{\max} = 0.9$ |
| Constant | – | $\omega = 0.729844$ |
| DE-PSO | (25) | $\omega(0) = 0.9$ |
| DW-PSO | (11) | – |
| FG-PSO | (24) | $\omega(0) = 0.9$ |
| IPSO-LT | (32) | $\alpha = 0.5, \beta = 0.5$ |
| LD-PSO | (18) | $\omega_{\min} = 0.4, \omega_{\max} = 0.9$ |
| PSO-NL | (9) | $\omega_{\min} = 0.4, \omega_{\max} = 0.9, \alpha = 1/\pi^2$ |
| PSO-NLI | (10) | $\omega = 0.3, u = 1.0002$ |
| PSO-NEIW | (14) | $\omega_{\min} = 0.4, \omega_{\max} = 0.9$ |
| PSO-OIW | (15) | $\omega_{\min} = 0.3, \omega_{\max} = 0.9, k = 7$ |
| PSO-RIW | (7) | – |
| PSO-RBI | (27) | $\omega_{\min} = 0.4, \omega_{\max} = 0.9$ |
| SRPSO | (19) | $\omega_s = 0.9, \omega_f = 0.4, \eta = 1$ |
| PSO-SIW | (17) | $s = 2$ |
| PSO-LDIW | (8) | $\omega_{\min} = 0.4, \omega_{\max} = 0.9$ |

rithms employed acceleration coefficients of $c_1 = c_2 = 1.496180$. Details for the remainder of the experimental procedure were as follows.

### 5.1 Benchmark problems

As summarized in Table 2, a suite of 60 minimization problems proposed by Engelbrecht (2013a) were used in this study. The benchmark problems provide a wide variety of problem characteristics including unimodal, multimodal, separable, non-separable, rotated, shifted, noisy, and composition functions. Moreover, the suite has been demonstrated to include a range of different landscape characteristics, including both smooth and rugged fitness landscapes along with a variety of gradients (Garden and Engelbrecht 2014). In Table 2, the column 'F' specifies the unique identifier associated with each function while the column 'Eq' specifies the equation number of the base function (equations are provided in "Appendix 1"). The column 'M' specifies the modality of the problem, where the values 'U' and 'M' denote unimodal and multimodal, respectively, and the column 'S' denotes the separability, where 'S' and 'NS' denote separable and non-separable, respectively. Note that while there are only 37 base functions, shifted, rotated, rotated and shifted, and noisy versions of some functions were used, resulting in a total of 60 functions. The configuration of the modified versions are indicated in the columns 'Sh', 'R', 'ShR', and 'N', respectively. Finally, a ✓ in column 'E' denotes an expanded function while a ✓ in column 'C' denotes a composition function. All functions were optimized in 30 dimensions. Further information about the benchmark functions is provided in "Appendix 1".

### 5.2 Performance measures

The inertia weight control strategies were compared with respect to the following performance measures:

- *Accuracy* the fitness of the global-best particle after 5000 iterations.
- *Consistency* the squared difference between the accuracy of each independent run and the average accuracy obtained across all runs.
- *Success rate* the percentage of the 50 independent runs which reached specified accuracy levels. A total of 1000 accuracy levels were considered, with the accuracy levels starting at the best accuracy obtained by all algorithms and logarithmically increasing towards the worst accuracy.

In addition to the aforementioned performance measures, the average particle movement (Cleghorn and Engelbrecht 2014a, 2015) given by

$$\Delta(t+1) = \frac{1}{n_s} \sum_{i=1}^{n_s} ||\mathbf{x}_i(t+1) - \mathbf{x}_i(t)||, \tag{36}$$

where $n_s$ is the number of particles, which was also recorded as a measure of empirical convergence. A sensible upper threshold, $\Delta_{max}$, was defined as the maximal distance between any two points in the feasible search space, according to

$$\Delta_{max} = \sqrt{n_d (l-u)^2} \tag{37}$$

where $[l, u]$ is the (feasible) domain of the objective function and $n_d$ is the dimensionality of the problem (Cleghorn and Engelbrecht 2015). Thus, particles which exhibited movement values above $\Delta_{max}$ were considered to be demonstrating divergent behaviour. The use of $\Delta_{max}$

**Table 2** Characteristics of the benchmark functions

| F | Equations | M | S | Sh | R | ShR | N | E | C |
|---|---|---|---|---|---|---|---|---|---|
| $f_1$ | (38) | U | S | | | | | | |
| $f_2$ | (39) | M | NS | $\beta = -140$<br>$\gamma = 10$ | ortho (1) | $\beta = -140$<br>$\gamma = -32$<br>linear (100) | | | |
| $f_3$ | (40) | M | S | | | | | | |
| $f_4$ | (41) | M | NS | | | | | | |
| $f_5$ | (42) | U | S | $\beta = -450$<br>$\gamma = 10$ | ortho (1)<br>ortho (1) | $\beta = -450$<br>$\gamma = 10$<br>ortho (1) | | | |
| $f_6$ | (43) | M | NS | $\beta = -180$<br>$\gamma = 10$ | othro (1) | $\beta = -180$<br>$\gamma = -60$<br>linear (3) | | | |
| $f_7$ | (44) | U | S | | | | | | |
| $f_8$ | (45) | M | S | | | | | | |
| $f_9$ | (46) | M | NS | | | | | | |
| $f_{10}$ | (47) | U | NS | | | | | | |
| $f_{11}$ | (48), (49) | U | S | | | | $N(0, 1)$ | | |
| $f_{12}$ | (50) | M | S | $\beta = -330$<br>$\gamma = 2$ | ortho (1) | $\beta = -330$<br>$\gamma = 1$<br>linear (2) | | | |
| $f_{13}$ | (51) | M | NS | $\beta = 390$<br>$\gamma = 10$ | ortho (1) | | | | |
| $f_{14}$ | (52) | M | NS | | | | | | |
| $f_{15}$ | (53) | M | NS | | | $\beta = -300$<br>$\gamma = 20$<br>linear (3) | | ✓ | |
| $f_{16}$ | (54), (55) | U | NS | $\beta = -450$<br>$\gamma = 10$ | ortho (1) | | $N(0, 0.4)$<br>$\beta = -450$<br>$\gamma = 10$ | | |
| $f_{17}$ | (56) | U | NS | $\beta = -310$ | | | | | |
| $f_{18}$ | (57) | M | NS | $\beta = -460$ | | | | | |
| $f_{19}$ | (58) | U | S | | | | | | |
| $f_{20}$ | (59) | U | S | | | | | | |
| $f_{21}$ | (60) | M | NS | | | | | | |
| $f_{22}$ | (61) | U | S | $\beta = -450$<br>$\gamma = 10$ | | | | | |
| $f_{23}$ | (62) | M | S | | | | | | |
| $f_{24}$ | (63) | M | S | | | | | | |
| $f_{25}$ | (64) | M | S | | | $\beta = 90$<br>$\gamma = 0.1$<br>linear (5) | | | |
| $f_{26}$ | CEC | M | NS | $\beta = -130$<br>$\gamma = 1$ | | | | ✓ | |

**Table 2** continued

| F | Eq | M | S | Sh | R | ShR | N | E | C |
|---|---|---|---|---|---|---|---|---|---|
| $f_{27}$ | CEC | M | NS | | | | | | ✓ |
| $f_{28}$ | CEC | M | NS | | Yes | | | | ✓ |
| $f_{29}$ | CEC | M | NS | | | | Yes | | ✓ |
| $f_{30}$ | CEC | M | NS | | Yes | | | | ✓ |
| $f_{31}$ | CEC | M | NS | | Yes | | | | ✓ |
| $f_{32}$ | CEC | M | NS | | Yes | | | | ✓ |
| $f_{33}$ | CEC | M | NS | | Yes | | | | ✓ |
| $f_{34}$ | CEC | M | NS | | Yes | | | | ✓ |
| $f_{35}$ | CEC | M | NS | | Yes | | | | ✓ |
| $f_{36}$ | CEC | M | NS | | Yes | | | | ✓ |
| $f_{37}$ | CEC | M | NS | | Yes | | | | ✓ |

'F' specifies the unique identifier associated with each function. 'Eq' specifies the equation number of the base function. 'M' specifies the modality ('U' for unimodal, 'M' for multimodal). 'S' denotes the separability ('S' for separable, 'NS' for non-separable). Shifted, rotated, rotated and shifted, and noisy versions of some functions were also used. The configuration of the modified versions is indicated in the columns 'Sh', 'R', 'ShR', and 'N', respectively. A ✓ in column 'E' denotes an expanded function, while a ✓ in column 'C' denotes a composition function

as the criterion to determine convergence has been empirically demonstrated to be 98.79% accurate when correlated with the convergence criterion of Poli (Cleghorn and Engelbrecht 2015).

### 5.3 Statistical analysis

For each benchmark problem and performance measure, pairwise two-tailed Mann–Whitney $U$ tests were performed at a significance level of 0.05 to identify performance differences. When the Mann–Whitney $U$ test indicated that a difference existed among two strategies, the median performance measure values were used to assign wins and losses; the better performing strategy was awarded a win, while the inferior strategy was awarded a loss. The strategies were then assigned an overall rank based on the difference between the number of wins and losses, i.e. difference = wins − losses. Additionally, the best rank frequency (BRF) defined as the number of benchmark problems for which the strategy attained the highest rank was recorded. Finally, the Bonferroni–Dunn post hoc test was employed to generate critical difference plots which visually depict the average ranks with respect to the solution accuracy. Note that the Mann–Whitney $U$ tests assessed the fine-grained, per-function performance differences while the Bonferroni–Dunn test was used to indicate differences in average rank across an entire set of problems.

## 6 Results and discussion

This section presents the results of the experiments described in Sect. 5. Table 3 summarizes the results obtained across all benchmark problems using the aforementioned Mann–Whitney U statistical analysis procedure, while Table 4 presents the average and standard deviation of the ranks across all benchmark problems. Figure 5 presents the critical difference plot, graph-

**Table 3** Summary of performance across all 60 benchmark problems

| Strategy | Accuracy | | | Success rate | | | Consistency | | |
|---|---|---|---|---|---|---|---|---|---|
| | Sum | Rank | BRF | Sum | Rank | BRF | Sum | Rank | BRF |
| AIWPSO | 143 | 8 | 7 | 96 | 7 | 16 | 129 | 7 | 10 |
| APSO-VI | 471 | 4 | 15 | 174 | 3 | 17 | 276 | 4 | 17 |
| CDIW-PSO | 313 | 6 | 5 | 134 | 5 | 14 | 98 | 8 | 8 |
| Constant | 630 | 2 | 23 | **387** | **1** | 27 | **364** | **1** | 22 |
| DE-PSO | −874 | 18 | 1 | −362 | 18 | 14 | −322 | 18 | 11 |
| DW-PSO | −565 | 16 | 1 | −247 | 16 | 14 | −275 | 15 | 9 |
| FG-PSO | −534 | 15 | 2 | −226 | 15 | 14 | −279 | 16 | 9 |
| IPSO-LT | −28 | 10 | 1 | 56 | 8 | 14 | −160 | 12 | 9 |
| LD-PSO | 303 | 7 | 1 | 48 | 9 | 14 | 146 | 6 | 9 |
| PSO-NL | −221 | 13 | 1 | −66 | 12 | 15 | −184 | 13 | 7 |
| PSO-NLI | −697 | 17 | 1 | −292 | 17 | 14 | −304 | 17 | 7 |
| PSO-NEIW | −87 | 11 | 1 | −51 | 11 | 14 | −92 | 11 | 8 |
| PSO-OIW | −327 | 14 | 1 | −157 | 14 | 14 | −245 | 14 | 7 |
| PSO-RIW | **637** | **1** | 11 | 349 | 2 | 19 | 352 | 2 | 18 |
| PSO-RBI | −97 | 12 | 1 | −71 | 13 | 14 | −81 | 10 | 7 |
| SRPSO | 494 | 3 | 3 | 155 | 4 | 14 | 288 | 3 | 10 |
| PSO-SIW | 2 | 9 | 2 | −50 | 10 | 15 | 20 | 9 | 10 |
| PSO-LDIW | 439 | 5 | 3 | 117 | 6 | 14 | 276 | 4 | 9 |

Bold entries indicate the strategies with the highest rank based on the sum of wins and losses across all benchmark problems

ically indicating the average rank of each algorithm with respect to the accuracy measure. Algorithms grouped with a line have no significant difference in average rank. Additionally, "Appendix 2" presents the overall ranks attained when the local-best topology was used in place of the global-best topology.

The first observation from Table 3 is that the random strategy (i.e. PSO-RIW) attained the best rank for the solution accuracy with a difference score of 637. However, on only 11 (18.3%) of the 60 problems did the random strategy attain the best fitness while a constant inertia weight strategy attained the best accuracy on 23 (36.7%) problems, and attained a difference score of 630. Furthermore, the APSO-VI strategy demonstrated the best performance on 15 (25.0%) problems, i.e. four more than the random strategy, but ranked fourth overall with a difference score of 471. Of the examined strategies, the DE-PSO depicted the worst overall accuracy with a difference score of −874. Considering the average ranks over all benchmark functions presented in Table 4, the constant and random strategies attained average ranks of 3.000 and 3.033 with standard deviations (SDs) of 2.456 and 2.484, respectively, and therefore demonstrated nearly indistinguishable accuracy overall. This is further evidenced by the critical difference plot in Fig. 5, which indicated that the average rank was not significantly different among the top six algorithms.

These results clearly indicate that when considering solution accuracy for an arbitrary problem, a constant or random inertia weight is preferable. Furthermore, the results show that none of the adaptive inertia weight strategies performed as well as the simpler constant and random strategies when considering the fine-grained, per-function analysis of the Mann–

**Table 4** Average rank and standard deviation (SD) across all benchmark problems

| Strategy | Accuracy | | Success rate | | Consistency | |
|---|---|---|---|---|---|---|
| | Average | SD | Average | SD | Average | SD |
| AIWPSO | 7.700 | 4.806 | 5.667 | 5.059 | 7.000 | 5.042 |
| APSO-VI | 5.183 | 4.094 | 3.923 | 3.542 | 5.183 | 4.678 |
| CDIW-PSO | 5.767 | 2.375 | 4.256 | 3.126 | 7.050 | 4.272 |
| Constant | 3.000 | 2.456 | 2.103 | 2.245 | 4.900 | 5.184 |
| DE-PSO | 17.200 | 2.910 | 11.795 | 8.192 | 11.267 | 6.996 |
| DW-PSO | 14.467 | 2.807 | 10.154 | 7.066 | 10.700 | 6.091 |
| FG-PSO | 13.783 | 3.683 | 9.795 | 6.921 | 10.683 | 5.685 |
| IPSO-LT | 9.317 | 4.710 | 5.179 | 3.684 | 10.133 | 6.074 |
| LD-PSO | 6.950 | 2.896 | 4.949 | 3.783 | 6.317 | 4.164 |
| PSO-NL | 11.183 | 2.920 | 7.436 | 5.510 | 10.133 | 4.918 |
| PSO-NLI | 15.733 | 3.399 | 10.846 | 7.524 | 11.333 | 6.337 |
| PSO-NEIW | 10.350 | 2.193 | 7.205 | 4.964 | 9.600 | 4.439 |
| PSO-OIW | 12.083 | 3.614 | 8.282 | 6.017 | 10.750 | 5.389 |
| PSO-RIW | 3.033 | 2.484 | 2.154 | 1.565 | 3.850 | 4.149 |
| PSO-RBI | 10.833 | 2.631 | 7.282 | 5.477 | 9.317 | 4.579 |
| SRPSO | 4.917 | 3.055 | 3.923 | 3.467 | 4.817 | 3.864 |
| PSO-SIW | 9.483 | 2.534 | 6.615 | 4.881 | 7.600 | 4.203 |
| PSO-LDIW | 6.117 | 3.669 | 4.513 | 4.322 | 4.900 | 3.516 |

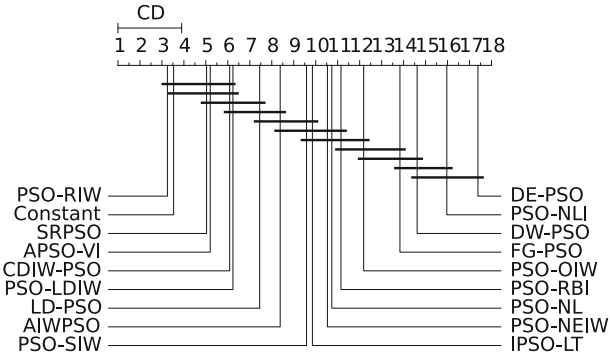


**Fig. 5** Critical difference plot using the Bonferroni–Dunn test on all benchmark problems

Whitney *U* tests. However, when the overall average rank was considered via the Bonferroni–Dunn test, far less discrepancies in accuracy were noted.

When considering the success rate, the constant inertia weight strategy attained both the best overall rank, with a difference score of 387, and the largest number of problems for which it obtained the best performance, with 27 (45.0%) of such problems. As with the accuracy measure, the random strategy was very close in performance to the constant strategy, obtaining the second highest rank, with a difference score of 349, and the best performance on 19 (31.7%) of the problems. A noteworthy observation is that the difference scores relative to the success rate were significantly lower than with the accuracy measure, indicating that

there were less discrepancies in performance, i.e. the various strategies had a more similar level of performance. When considering the average ranks, the constant strategy attained the best average rank of 2.103 with a standard deviation of 2.245 while the random strategy had a slightly higher average rank of 2.154 coupled with a lower standard deviation of 1.565. This indicates that the random strategy was much more consistent in terms of the success rate. The worst overall success rate was observed when using the DE-PSO strategy, with a difference score of −382.

In terms of consistency, the constant inertia weight strategy had the lowest overall deviations from the average accuracy, with a difference score of 364 while the second best strategy, the random strategy, attained a difference score of 352. Furthermore, the constant strategy had the best consistency on 22 (36.7%) of the problems while the random strategy was most consistent on 18 (30.0%). Despite having better performance overall, the average rank of the constant strategy, 4.900 with a SD of 5.184, was significantly higher than the average rank of the random strategy at 3.850 with a SD of 4.149. As with the other two measures, the DE-PSO strategy showed the worst overall performance with a difference score of −322.

To provide empirical evidence of the convergence analyses in Sects. 3 and 4, Figs. 6 and 7 depict the average particle movement values over time on two problem instances, namely $f_1$ and $f_{17}$, which are representative of the overall behaviour of each algorithm. An immediate observation was that the only algorithm to exhibit an average particle movement above $\Delta_{\max}$ after 5000 iterations was the PSO-RBI algorithm. Recall that of all the examined strategies, PSO-RBI was the only strategy which was not expected to exhibit convergent behaviour. Furthermore, recall that there were three algorithms expected to demonstrate immediate convergence, namely the PSO-NLI, DW-PSO, and FG-PSO algorithms. The average particle step sizes for these algorithms are shown in Figs. 6j and 7j for PSO-NLI, Figs. 6f and 7f for DW-PSO, and Figs. 6g and 7g for FG-PSO—each of which indicates nearly stagnant particles after only a few iterations. Additionally, recall that both PSO-NEIW and LD-PSO were expected to exhibit convergent behaviour after approximately 2.6% of the search had completed. Considering Figs. 6l, i and 7l, i, which visualize the particle step sizes for the PSO-NEIW and LD-PSO algorithms, it is evidenced that both algorithms exhibited divergent behaviour initially, but depicted rapid decreases in particle movement very early in the search.

Further examination of the particle movement plots in Figs. 6 and 7 highlighted a correlation between particle movement and performance. Specifically, the six worst performing algorithms, namely DE-PSO, PSO-NLI, DW-PSO, FG-PSO, PSO-OIW, and PSO-NL, each demonstrated prohibitively low particle movement values after only a few iterations. Thus, the worst performing strategies exhibited extremely premature convergence and thereby had insufficient particle movement to perform an effective search. Conversely, the two best performing strategies, namely the PSO-RIW and constant strategies, maintained a slightly higher level of particle activity than the worst performing strategies. However, both the PSO-RIW and constant strategies depicted higher initial levels of movement followed by a noticeably more gradual decline. Finally, the mid-performing algorithms, e.g. LD-PSO, typically depicted large initial movements, suggesting initially divergent behaviour, followed by a rapid decline in movement levels, suggesting stagnation. It is thus concluded that for the mid-performing strategies, the particles rapidly move outwards from their starting locations, finding moderately good solutions along the way, but eventually stagnate without locating any further promising areas.

In summary, the constant and random (PSO-RIW) strategies attained the highest ranks for all performance measures when aggregated across all 60 benchmark problems. Moreover, the constant and random control strategies performed relatively similar for all performance measures and both showed a significant improvement over the next (third) best strategy.

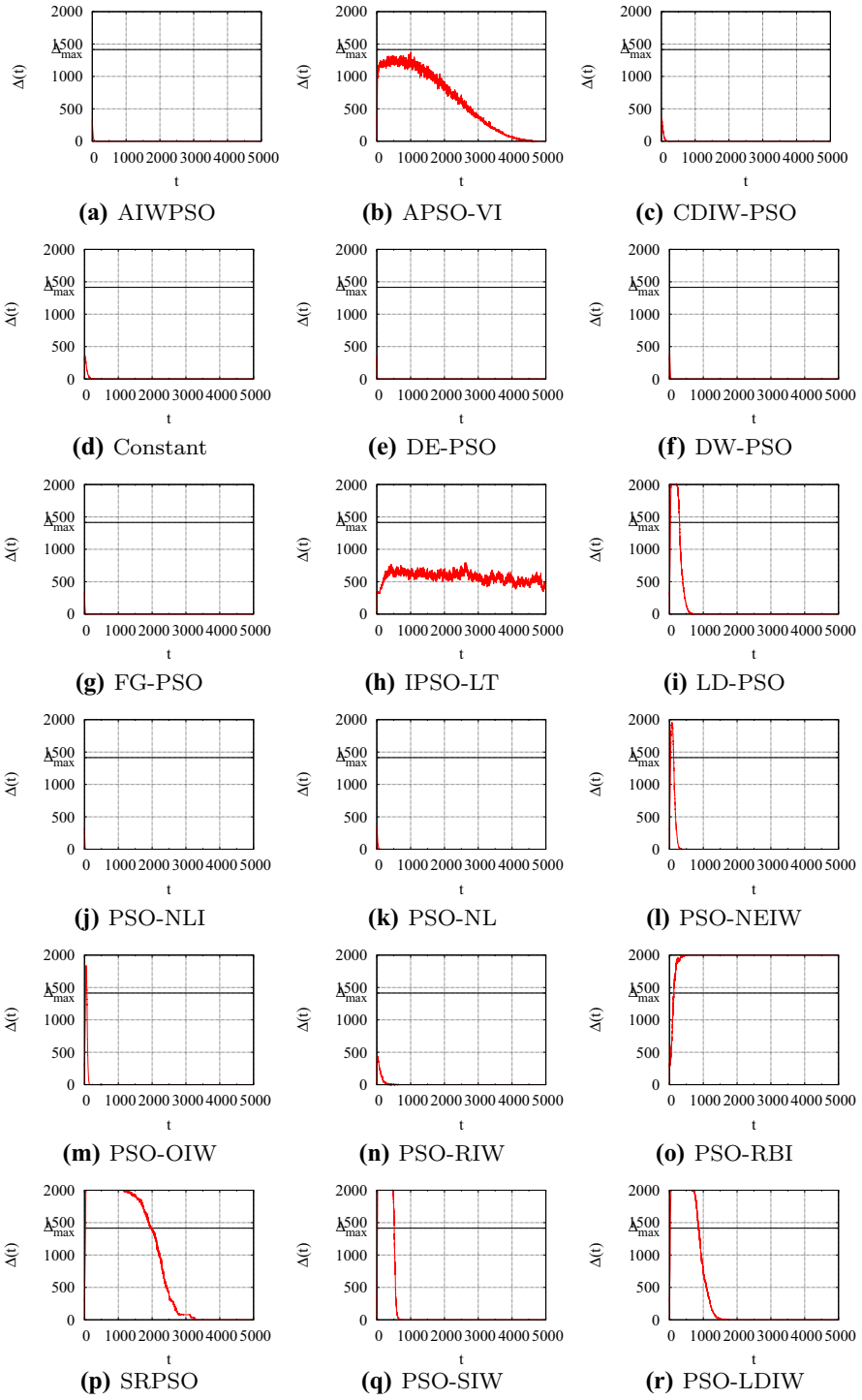


**Fig. 6** Average particle movement on function $f_1$ with $\Delta_{\max} = 1414.214$

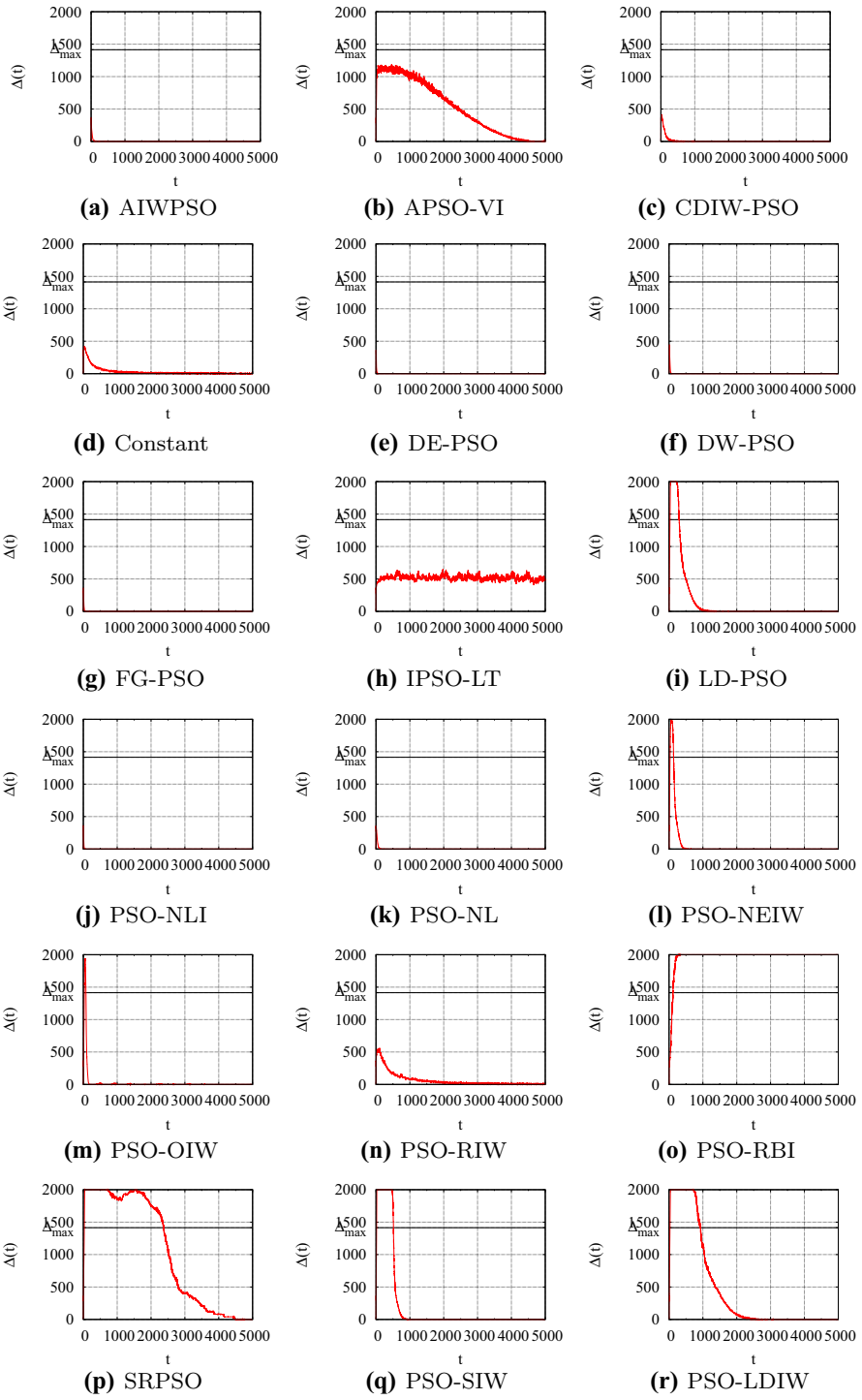


**Fig. 7** Average particle movement on function $f_{17}$ with $\Delta_{\max} = 1414.214$

**Table 5** Summary of performance on the 19 unimodal problems

| Strategy | Accuracy | | | Success rate | | | Consistency | | |
|---|---|---|---|---|---|---|---|---|---|
| | Sum | Rank | BRF | Sum | Rank | BRF | Sum | Rank | BRF |
| AIWPSO | 89 | 8 | 2 | 73 | 3 | 6 | 93 | 5 | 3 |
| APSO-VI | 113 | 6 | 1 | 21 | 6 | 4 | 92 | 6 | 1 |
| CDIW-PSO | 116 | 5 | 1 | 19 | 8 | 4 | 60 | 8 | 1 |
| Constant | **291** | **1** | 12 | **180** | **1** | 14 | **255** | **1** | 13 |
| DE-PSO | −302 | 18 | 0 | −96 | 18 | 4 | −203 | 18 | 1 |
| DW-PSO | −212 | 16 | 0 | −82 | 16 | 4 | −149 | 15 | 1 |
| FG-PSO | −202 | 15 | 0 | −76 | 15 | 4 | −154 | 16 | 1 |
| IPSO-LT | −42 | 11 | 0 | −12 | 9 | 4 | −96 | 13 | 2 |
| LD-PSO | 112 | 7 | 0 | 21 | 6 | 4 | 74 | 7 | 1 |
| PSO-NL | −101 | 13 | 0 | −50 | 13 | 4 | −82 | 12 | 1 |
| PSO-NLI | −256 | 17 | 0 | −91 | 17 | 4 | −174 | 17 | 1 |
| PSO-NEIW | −58 | 12 | 0 | −28 | 10 | 4 | −50 | 11 | 1 |
| PSO-OIW | −121 | 14 | 0 | −68 | 14 | 4 | −124 | 14 | 1 |
| PSO-RIW | 262 | 2 | 5 | 152 | 2 | 7 | 227 | 2 | 5 |
| PSO-RBI | −17 | 9 | 0 | −28 | 10 | 4 | −30 | 10 | 1 |
| SRPSO | 187 | 3 | 0 | 57 | 4 | 4 | 147 | 3 | 1 |
| PSO-SIW | −32 | 10 | 0 | −30 | 12 | 4 | −18 | 9 | 1 |
| PSO-LDIW | 163 | 4 | 0 | 50 | 5 | 4 | 134 | 4 | 1 |

Bold entries indicate the strategies with the highest rank based on the sum of wins and losses

Therefore, it can be concluded that when faced with the optimization of an arbitrary function, the use of an inertia weight control strategy is not recommended because the constant and random strategies both lead to consistently better results. Furthermore, the results presented here highlight the drastic misrepresentations of the various algorithms' performance in the literature which stems from the limited analyses when they were proposed. Finally, an examination of the average particle movement over time highlighted a correlation between the level of particle movement and performance.

In the following sections, the performance of the inertia weight control strategies is analysed with respect to various characteristics of the benchmark problems. The performance is examined with respect to unimodal, multimodal, separable, non-separable, and composition problems to ascertain whether such characteristics of the problem affect the performance.

### 6.1 Unimodal problems

Table 5 depicts the overall rank and best rank frequency of the examined strategies across the 19 unimodal problems, while Fig. 8 shows the critical difference plot for the accuracy measure. For all three performance measures, the constant strategy attained the highest rank using the Mann–Whitney $U$ tests, followed by the random strategy. The constant strategy had difference scores of 291, 180, and 255 for the accuracy, success rate, and consistency measures, respectively. However, the critical difference plot in Fig. 8 depicts insignificant differences in average rank for the accuracy measure across the top six performing algorithms, despite the finer-grained differences noted by the Mann–Whitney $U$ tests. Additionally, the

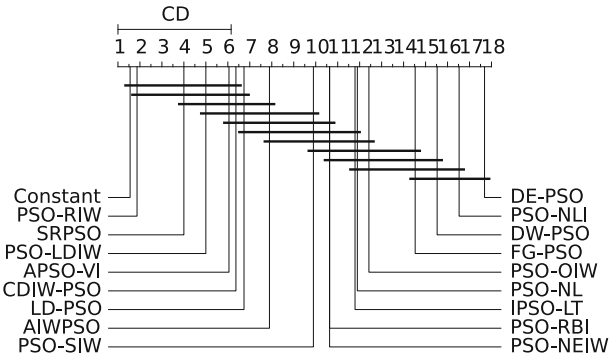


**Fig. 8** Critical difference plot using the Bonferroni–Dunn test on the 19 unimodal benchmark problems

**Table 6** Summary of performance on the 41 multimodal problems

| Strategy | Accuracy | | | Success rate | | | Consistency | | |
|---|---|---|---|---|---|---|---|---|---|
| | Sum | Rank | BRF | Sum | Rank | BRF | Sum | Rank | BRF |
| AIWPSO | 54 | 8 | 5 | 23 | 9 | 20 | 36 | 9 | 7 |
| APSO-VI | 358 | 2 | 14 | 153 | 3 | 23 | **184** | **1** | 16 |
| CDIW-PSO | 197 | 6 | 4 | 115 | 4 | 17 | 38 | 7 | 7 |
| Constant | 339 | 3 | 11 | **207** | **1** | 25 | 109 | 5 | 9 |
| DE-PSO | −572 | 18 | 1 | −266 | 18 | 17 | −119 | 14 | 10 |
| DW-PSO | −353 | 16 | 1 | −165 | 16 | 17 | −126 | 17 | 8 |
| FG-PSO | −332 | 15 | 2 | −150 | 15 | 17 | −125 | 16 | 8 |
| IPSO-LT | 14 | 10 | 1 | 68 | 6 | 17 | −64 | 12 | 7 |
| LD-PSO | 191 | 7 | 1 | 27 | 8 | 17 | 72 | 6 | 8 |
| PSO-NL | −120 | 13 | 1 | −16 | 10 | 18 | −102 | 13 | 6 |
| PSO-NLI | −441 | 17 | 1 | −201 | 17 | 17 | −130 | 18 | 6 |
| PSO-NEIW | −29 | 11 | 1 | −23 | 12 | 17 | −42 | 10 | 6 |
| PSO-OIW | −206 | 14 | 1 | −89 | 14 | 17 | −121 | 15 | 6 |
| PSO-RIW | **375** | **1** | 6 | 197 | 2 | 22 | 125 | 4 | 13 |
| PSO-RBI | −80 | 12 | 1 | −43 | 13 | 17 | −51 | 11 | 6 |
| SRPSO | 307 | 4 | 3 | 98 | 5 | 17 | 141 | 3 | 9 |
| PSO-SIW | 34 | 9 | 2 | −20 | 11 | 18 | 38 | 7 | 9 |
| PSO-LDIW | 276 | 5 | 3 | 67 | 7 | 17 | 142 | 2 | 8 |

Bold entries indicate the strategies with the highest rank based on the sum of wins and losses

constant strategy had the largest number of problems for which it attained the highest rank with respect to each performance measure, with 12 (63.2%), 14 (73.7%), and 13 (68.4%) problems, respectively. The worst overall strategy for all performance measures was DE-PSO indicating that it not only showed the worst accuracy, but also the lowest consistency.

## 6.2 Multimodal problems

A summary of the results for the 41 multimodal problems is presented in Table 6, while Fig. 9 presents the critical difference plot for the accuracy measure. With respect to accuracy, the

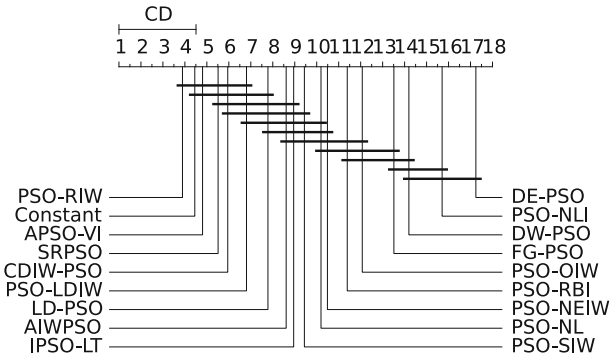


**Fig. 9** Critical difference plot using the Bonferroni–Dunn test on the 41 multimodal benchmark problems

random strategy had the best overall performance with a difference score of 375. However, the random strategy had the best accuracy on only six (14.6%) problems, outnumbered by both APSO-VI, with 14 (34.1%) problems, and the constant strategy, with 11 (26.8%) problems. A noteworthy observation was that the constant inertia weight ranked third best on multimodal problems, with a difference score of 339, while the second best accuracy was attained by the APSO-VI strategy, with a difference score of 358. Thus, the constant inertia weight had degraded accuracy when faced with multimodal problems. The DE-PSO again had the worst performance, with a difference score of −572.

Despite the degraded performance relative to the accuracy measure, the constant inertia weight strategy attained the highest rank for the success rate measure, with a difference score of 207. This indicates that while the constant strategy did not attain the best accuracy, it did provide convergence to more accurate solutions, in general. Furthermore, the constant strategy depicted the best success rate on 25 (61.0%) of the multimodal problems. However, it should be noted that the success rate was insignificantly different among the strategies on many problems as even the worst performing strategy, namely DE-PSO, displayed the highest rank on 17 (41.5%) of problems.

The APSO-VI strategy had the most consistent results on multimodal problems with a difference score of 184, followed by the PSO-LDIW strategy with a score of 142. The APSO-VI also had the largest number of problems, 16 (39.0%), for which it attained the highest rank. Note that the constant and random strategies attained ranks of fifth and fourth, respectively, and that the consistency measure on multimodal problems is one of only two scenarios in which the constant or random strategy did not attain the highest rank. The least consistent results were attained by the PSO-NLI strategy, which attained a difference score of −130.

### 6.3 Separable problems

The performance on the 21 separable problems is summarized in Table 7. Figure 10 depicts the critical differences with respect to the accuracy measure. For all three performance measures, the constant strategy showed the best performance with difference scores of 244, 134, and 185 for the accuracy, success rate, and consistency measures, respectively. Furthermore, the constant strategy also had the highest number of functions for which it attained the highest rank with 9 (24.9%), 12 (57.1%), and 9 (24.9%), respectively. The worst performance on separable problems was depicted by the DE-PSO with difference scores of −320, −94, and −162, respectively.

**Table 7** Summary of performance on 21 separable problems

| Strategy | Accuracy | | | Success rate | | | Consistency | | |
|---|---|---|---|---|---|---|---|---|---|
| | Sum | Rank | BRF | Sum | Rank | BRF | Sum | Rank | BRF |
| AIWPSO | 94 | 6 | 3 | 70 | 3 | 10 | 87 | 6 | 5 |
| APSO-VI | 158 | 4 | 5 | 46 | 4 | 10 | 122 | 3 | 7 |
| CDIW-PSO | 90 | 7 | 1 | 20 | 7 | 7 | 38 | 8 | 2 |
| Constant | **244** | **1** | 9 | **134** | **1** | 12 | **185** | **1** | 9 |
| DE-PSO | −320 | 18 | 0 | −94 | 18 | 7 | −162 | 18 | 2 |
| DW-PSO | −195 | 16 | 0 | −66 | 16 | 7 | −125 | 15 | 3 |
| FG-PSO | −184 | 15 | 0 | −56 | 15 | 7 | −135 | 16 | 2 |
| IPSO-LT | −18 | 9 | 0 | −3 | 8 | 7 | −36 | 10 | 2 |
| LD-PSO | 82 | 8 | 0 | −5 | 9 | 7 | 49 | 7 | 2 |
| PSO-NL | −78 | 13 | 0 | −29 | 11 | 7 | −81 | 13 | 2 |
| PSO-NLI | −242 | 17 | 0 | −72 | 17 | 7 | −146 | 17 | 2 |
| PSO-NEIW | −52 | 12 | 0 | −30 | 12 | 7 | −51 | 12 | 2 |
| PSO-OIW | −105 | 14 | 0 | −44 | 14 | 7 | −108 | 14 | 2 |
| PSO-RIW | 243 | 2 | 5 | 123 | 2 | 10 | 179 | 2 | 7 |
| PSO-RBI | −21 | 10 | 0 | −23 | 10 | 7 | −48 | 11 | 2 |
| SRPSO | 179 | 3 | 0 | 39 | 5 | 7 | 116 | 5 | 2 |
| PSO-SIW | −29 | 11 | 0 | −30 | 12 | 7 | −14 | 9 | 2 |
| PSO-LDIW | 153 | 5 | 0 | 28 | 6 | 7 | 117 | 4 | 2 |

Bold entries indicate the strategies with the highest rank based on the sum of wins and losses

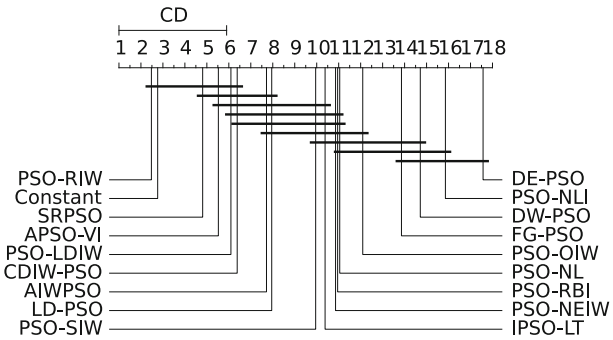


**Fig. 10** Critical difference plot using the Bonferroni–Dunn test on the 21 separable benchmark problems

## 6.4 Non-separable problems

Table 8 depicts the performance of the various strategies on the 39 non-separable problems, while Fig. 11 presents the critical differences for the accuracy measure. Considering the accuracy measure, the random strategy attained the most accurate solutions, with a difference score of 394. Despite having the highest rank overall, the random strategy attained the best performance on only six (15.4%) of the problems, while the constant strategy had the highest rank on 14 (35.9%) problems and APSO-VI performed best on 10 (25.6%) problems. Thus,

**Table 8** Summary of performance on the 39 non-separable problems

| Strategy | Accuracy | | | Success rate | | | Consistency | | |
|---|---|---|---|---|---|---|---|---|---|
| | Sum | Rank | BRF | Sum | Rank | BRF | Sum | Rank | BRF |
| AIWPSO | 49 | 8 | 4 | 26 | 9 | 16 | 42 | 8 | 5 |
| APSO-VI | 313 | 4 | 10 | 128 | 3 | 17 | 154 | 5 | 10 |
| CDIW-PSO | 223 | 6 | 4 | 114 | 5 | 14 | 60 | 7 | 6 |
| Constant | 386 | 2 | 14 | **253** | **1** | 27 | **179** | **1** | 13 |
| DE-PSO | −554 | 18 | 1 | −268 | 18 | 14 | −160 | 18 | 9 |
| DW-PSO | −370 | 16 | 1 | −181 | 16 | 14 | −150 | 16 | 6 |
| FG-PSO | −350 | 15 | 2 | −170 | 15 | 14 | −144 | 15 | 7 |
| IPSO-LT | −10 | 10 | 1 | 59 | 7 | 14 | −124 | 13 | 7 |
| LD-PSO | 221 | 7 | 1 | 53 | 8 | 14 | 97 | 6 | 7 |
| PSO-NL | −143 | 13 | 1 | −37 | 12 | 15 | −103 | 12 | 5 |
| PSO-NLI | −455 | 17 | 1 | −220 | 17 | 14 | −158 | 17 | 5 |
| PSO-NEIW | −35 | 11 | 1 | −21 | 11 | 14 | −41 | 11 | 5 |
| PSO-OIW | −222 | 14 | 1 | −113 | 14 | 14 | −137 | 14 | 5 |
| PSO-RIW | **394** | **1** | 6 | 226 | 2 | 19 | 173 | 2 | 11 |
| PSO-RBI | −76 | 12 | 1 | −48 | 13 | 14 | −33 | 10 | 5 |
| SRPSO | 315 | 3 | 3 | 116 | 4 | 14 | 172 | 3 | 8 |
| PSO-SIW | 31 | 9 | 2 | −20 | 10 | 15 | 34 | 9 | 8 |
| PSO-LDIW | 286 | 5 | 3 | 89 | 6 | 14 | 159 | 4 | 7 |

Bold entries indicate the strategies with the highest rank based on the sum of wins and losses

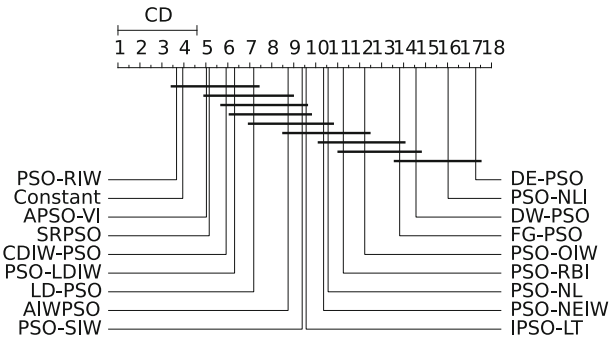


**Fig. 11** Critical difference plot using the Bonferroni–Dunn test on the 39 non-separable benchmark problems

despite attaining the highest accuracy on more than double the number of problems, the constant strategy performed slightly worse, with a difference score of 386, than the random strategy on non-separable problems. The worst accuracy was attained by the DE-PSO with a difference score of −554.

The observations for the success rate and consistency measures were the same: the best performance was noted for the constant strategy, with difference scores of 253 and 179, respectively, while the worst performance was attained by the DE-PSO with difference scores of −268 and −160, respectively. Additionally, the constant strategy had the largest number

**Table 9** Summary of performance on the 11 composition problems

| Strategy | Accuracy | | | Success rate | | | Consistency | | |
|---|---|---|---|---|---|---|---|---|---|
| | Sum | Rank | BRF | Sum | Rank | BRF | Sum | Rank | BRF |
| AIWPSO | −2 | 10 | 1 | −3 | 11 | 1 | 0 | 7 | 2 |
| APSO-VI | 67 | 2 | 2 | 69 | 4 | 2 | −6 | 10 | 3 |
| CDIW-PSO | 63 | 4 | 2 | 84 | 2 | 0 | −21 | 13 | 3 |
| Constant | **73** | **1** | 4 | **118** | **1** | 6 | −30 | 15 | 3 |
| DE-PSO | −134 | 18 | 0 | −143 | 18 | 0 | **74** | **1** | 6 |
| DW-PSO | −65 | 16 | 0 | −80 | 16 | 0 | 34 | 4 | 3 |
| FG-PSO | −64 | 15 | 0 | −69 | 15 | 0 | 42 | 3 | 3 |
| IPSO-LT | 42 | 6 | 0 | 47 | 5 | 0 | −35 | 17 | 2 |
| LD-PSO | 31 | 8 | 0 | 10 | 8 | 0 | −25 | 14 | 2 |
| PSO-NL | −6 | 12 | 0 | 11 | 7 | 1 | −17 | 12 | 2 |
| PSO-NLI | −89 | 17 | 0 | −106 | 17 | 0 | 56 | 2 | 2 |
| PSO-NEIW | −4 | 11 | 0 | 0 | 10 | 0 | −30 | 15 | 2 |
| PSO-OIW | −37 | 14 | 0 | −42 | 14 | 0 | 14 | 5 | 2 |
| PSO-RIW | 65 | 3 | 1 | 76 | 3 | 0 | −35 | 17 | 2 |
| PSO-RBI | −22 | 13 | 0 | −20 | 13 | 0 | 7 | 6 | 2 |
| SRPSO | 47 | 5 | 0 | 41 | 6 | 0 | −5 | 9 | 2 |
| PSO-SIW | 9 | 9 | 1 | −7 | 12 | 1 | −12 | 11 | 3 |
| PSO-LDIW | 33 | 7 | 0 | 6 | 9 | 0 | −4 | 8 | 2 |

Bold entries indicate the strategies with the highest rank based on the sum of wins and losses

of functions for which it attained the highest rank relative to both measures, with 27 (69.2%) and 13 (33.3%) problems, respectively.

### 6.5 Composition problems

The final type of problem that was considered in this study is composition problems. The results for the 11 composition problems are presented in Table 9. For both the accuracy and success rate measures, the constant strategy attained the highest rank overall, with difference scores of 73 and 118, respectively. For both measures, the constant inertia weight also attained the best performance on four (36.4%) and six (54.5%) problems, respectively. An additional observation was that the difference scores for the accuracy measure were lower than those of the success rate which indicates that there was significantly less deviation among the accuracy levels of the various strategies relative to the other problem classes. To further support this observation, Fig. 12 presents the critical difference plot on the composition problems. This plot indicates that very minute differences in average rank were observed among the various strategies when considering the accuracy measure. For both measures, the DE-PSO depicted the worst overall performance.

When considering the consistency measure for composition problems, a few unexpected results arose. Firstly, the most consistent strategy was the DE-PSO with a difference score of 74 and the highest rank on six (54.5%) problems. This is one of only two scenarios in which the DE-PSO did not exhibit the worst performance. The second unexpected result was the rank of the constant and random strategies at 15 and 17, respectively. Note that the rank of

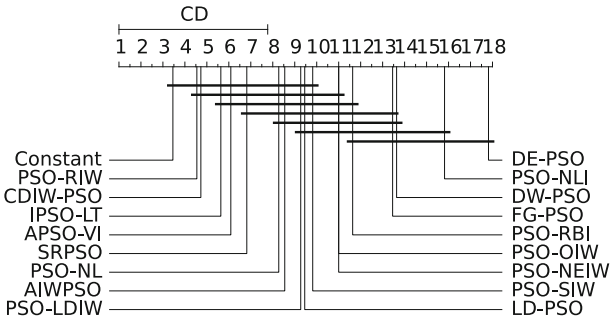


**Fig. 12** Critical difference plot using the Bonferroni–Dunn test on the 11 composition benchmark problems

17 for the random strategy was a tie with IPSO-LT and thus constituted the worst overall consistency and that the rank of 15 for the constant strategy was a tie with PSO-NEIW and thus constituted the penultimate consistency. The low rank attained by the random and constant strategies was quite unexpected given their performance on all other problem classes. This shows that despite their top performance, the random and constant strategies are extremely unstable on composition problems.

### 6.6 Summary

In summary, the previous sections provided an empirical investigation into 18 inertia weight control strategies and further drilled down into the results by looking at five prominent characteristics of the benchmark problems, namely unimodal, multimodal, separable, non-separable, and composition, to ascertain whether such characteristics have an impact on performance. A further examination of particle movement levels highlighted a correlation between particle movement and performance. Specifically, the best performing strategies depicted moderate initial movement levels followed by a gradual decrease in movement. Conversely, the remaining strategies depicted either prohibitively low particle movements or rapid divergence followed by rapid stagnation. A summary of the best and worst performing strategies for each problem type is given in Table 10. Thus, Table 10 can be taken as a recommendation of which strategy to employ based on the type of function and performance measure being examined. However, as indicated in "Appendix 2", the best strategy to employ changes with the topology selected. Therefore, topology should also be considered when selecting an inertia weight strategy. Note that the worst performance for the consistency measure on the composition problems was a tie between IPSO-LT and the random strategy. In general, the problem type had only minimal effects on the overall results as the same general trend was observed, with only a few exceptions, regardless of the problem type. Specifically, the solutions were most accurate when using a constant or random inertia weight strategy, the constant weight strategy always had the most solutions converging to the specified accuracy levels and was the most consistent with respect to accuracy for three of the five problem classes. The remaining two problem types, namely multimodal and composition, had the most consistent results from the APSO-VI and DE-PSO strategies, respectively. In fact, it was only when considering the consistency on the multimodal and composition problem types that the DE-PSO attained anything but the worst overall performance. Apart from the two aforementioned scenarios, the DE-PSO performed the worst with regard to all performance measures on all problem types.

**Table 10** Best and worst performing strategy by function type

| Problem | Accuracy | | Success rate | | Consistency | |
|---|---|---|---|---|---|---|
| | Best | Worst | Best | Worst | Best | Worst |
| All | PSO-RIW | DE-PSO | Constant | DE-PSO | Constant | DE-PSO |
| U | Constant | DE-PSO | Constant | DE-PSO | Constant | DE-PSO |
| M | PSO-RIW | DE-PSO | Constant | DE-PSO | APSO-VI | PSO-NLI |
| S | Constant | DE-PSO | Constant | DE-PSO | Constant | DE-PSO |
| NS | PSO-RIW | DE-PSO | Constant | DE-PSO | Constant | DE-PSO |
| C | Constant | DE-PSO | Constant | DE-PSO | DE-PSO | PSO-RIW |

## 7 Conclusions

The primary purpose of this study was to analyse the performance of a number of inertia weight control strategies for the PSO algorithm on a comprehensive set of benchmark problems. More specifically, there were three overall objectives. Firstly, this study served as a review of various control strategies for the inertia weight. Secondly, this study analysed the convergence implications of the inertia weight control strategies, deriving exact conditions for convergence to be exhibited where possible. Finally, a suite of 60 benchmark functions was employed to empirically examine the performance of the strategies relative to three performance measures. The benchmark problems encompassed a plethora of different characteristics, allowing the performance to be correlated with problem type.

All examined strategies were given equivalent PSO configurations, with the obvious exception of the way in which the inertia weight was controlled. Results of the empirical analysis indicated that despite their respective reported successes in the literature, the only inertia weight control strategy which performed on par with a constant inertia weight was the random weight strategy. Therefore, given an arbitrary optimization problem, it is preferable to have a constant inertia weight. If a non-static inertia is desired, then randomly selecting the inertia weight each iteration is the only worthwhile strategy. Furthermore, an examination of the average particle movement over time highlighted a correlation between the level of particle movement and performance. These results depict a grim state for the field of adaptive inertia weight strategies. The results clearly show a problem with the way in which authors examine their proposed strategies as there is a tendency for newly proposed strategies to be compared with a small, seemingly arbitrarily chosen set of existing strategies and benchmark problems. However, as shown in "Appendix 2", the poor performance of the adaptive inertia weight strategies when the global-best topology was used did not necessarily hold when the local-best topology was used. It is also noteworthy that similar studies have found that adaptive parameter strategies perform worse than constant or random control parameters for both ant colony optimization (Pellegrini et al. 2012) and tabu search (Mascia et al. 2014).

This study only examined control strategies for the inertia weight, and therefore, an immediate avenue of future work involves examining control strategies for the acceleration coefficients. Additionally, examining the various inertia weight strategies for other PSO variants may prove to be fruitful. This study made no attempt to discover the best performing parameters for each of the inertia weight control strategies. Therefore, another avenue of future work is to analyse the sensitivity of the examined algorithms to their respective control parameters. Similarly, the results were noticeably different when using the global-best

and local-best topologies, and thus, future work will more explicitly focus on examining the implications associated with the use of different neighbourhood topologies. Finally, the performance should be analysed with respect to fitness landscape characteristics as this would assist in identifying the specific landscape features which promote good/bad performance.

## Appendix 1: Benchmark problems

This appendix provides a further, in-depth definition of the benchmark problems employed in this study.

A function, $f$, was shifted using

$$f^{\mathrm{Sh}}(\mathbf{x}) = f(\mathbf{x} - \gamma) + \beta$$

where $\beta$ and $\gamma$ are constants. Rotation was implemented using either a randomly generated orthonormal rotation matrix, denoted by 'ortho', or a linear transformation matrix, denoted by 'linear'. In either scenario, the rotation was performed using Salomon's method (Salomon 1996) with a new rotation matrix computed for each of the independent runs using the condition number provided. The rotated functions, denoted by $f^{\mathrm{R}}$, were then computed by multiplying the decision vector $\mathbf{x}$ by the transpose of the rotation matrix. Noisy functions, denoted by $f^{\mathrm{N}}$, were generated by multiplying each decision variable by a noise value sampled from a Gaussian distribution with the specified mean and deviation. Table 2 provides the configuration parameters for the shifted, rotated, rotated and shifted, and noisy versions of the functions in the columns 'Sh', 'R', 'ShR', and 'N', respectively.

It should be noted that the composition functions $f_{27} - f_{37}$ are equal to functions $f_{15} - f_{25}$ from the CEC 2005 benchmark set (Suganthan et al. 2005). For each of these problems, the composed functions are each rotated using linear transformation matrices with independent condition numbers. Thus, the reader is directed to Suganthan et al. (2005) for specific details regarding the configurations of these functions.

The equations for each benchmark problem are as follows.

$f_1$, the absolute value function, defined as

$$f_1(\mathbf{x}) = \sum_{j=1}^{n_x} |x_j| \tag{38}$$

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

$f_2$, the Ackley function, defined as

$$f_2(\mathbf{x}) = -20e^{-0.2\sqrt{\frac{1}{n}\sum_{j=1}^{n_x} x_j^2}} - e^{\frac{1}{n}\sum_{j=1}^{n_x}\cos(2\pi x_j)} + 20 + e \tag{39}$$

with each $x_j \in [-32.768, 32.768]$. Shifted, rotated, and rotated and shifted versions of $f_2$ were also used.

$f_3$, the alpine function, defined as

$$f_3(\mathbf{x}) = \left(\prod_{j=1}^{n_x} \sin(x_j)\right)\sqrt{\prod_{j=1}^{n_x} x_j} \tag{40}$$

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

$f_4$, the egg holder function, defined as

$$f_4(\mathbf{x}) = \sum_{j=1}^{n_x-1} \left(-(x_{j+1}+47)\sin\left(\sqrt{|x_{j+1}+x_j/2+47|}\right) + \sin\left(\sqrt{|x_j-(x_{j+1}+47)|}\right)(-x_j)\right) \tag{41}$$

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

$f_5$, the elliptic function, defined as

$$f_5(\mathbf{x}) = \sum_{j=1}^{n_x} (10^6)^{\frac{j-1}{n_x-1}} \tag{42}$$

with each $x_j \in [-100, 100]$. Shifted, rotated, and rotated and shifted versions of $f_5$ were also used.

$f_6$, the Griewank function, defined as

$$f_6(\mathbf{x}) = 1 + \frac{1}{4000}\sum_{j=1}^{n_x} x_j^2 - \prod_{j=1}^{n_x} \cos\left(\frac{x_j}{\sqrt{j}}\right) \tag{43}$$

with each $x_j \in [-600, 600]$. Shifted, rotated, and rotated and shifted versions of $f_6$ were also used. For the rotated and shifted version of $f_6$, the range was modified to $x_j \in [0, 600]$ such that the global minimum was outside the bounds.

$f_7$, the hyperellipsoid function, defined as

$$f_7(\mathbf{x}) = \sum_{j=1}^{n_x} j x_j^2 \tag{44}$$

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

$f_8$, the Michalewicz function, defined as

$$f_8(\mathbf{x}) = -\sum_{j=1}^{n_x} \sin(x_j)\left(\sin\left(\frac{jx_j^2}{\pi}\right)\right)^{2m} \tag{45}$$

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

$f_9$, the norwegian function, defined as

$$f_9(\mathbf{x}) = \prod_{j=1}^{n_x} \left(\cos(\pi x_j^3)\left(\frac{99+x_j}{100}\right)\right) \tag{46}$$

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

$f_{10}$, the quadric function, defined as

$$f_{10}(\mathbf{x}) = \sum_{i=1}^{n_x} \left(\sum_{j=1}^{i} x_j\right)^2 \tag{47}$$

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

$f_{11}$, the quartic function, defined as

$$f_{11}(\mathbf{x}) = \sum_{j=1}^{n_x} j x_j^4 \tag{48}$$

with each $x_j \in [-1.28, 1.28]$. A noisy version of the quartic function, referred to as De Jong's f4 function, was generated according to

$$f_{11}^N(\mathbf{x}) = \sum_{j=1}^{n_x} (j x_j^4 + N(0, 1)) \tag{49}$$

using the same domain as the quartic function.

$f_{12}$, the Rastrigin function, defined as

$$f_{12}(\mathbf{x}) = 10 n_x + \sum_{j=1}^{n_x} (x_j^2 - 10\cos(2\pi x_j)) \tag{50}$$

with each $x_j \in [-5.12, 5.12]$. Shifted, rotated, and rotated and shifted versions of $f_{12}$ were also used.

$f_{13}$, the Rosenbrock function, defined as

$$f_{13}(\mathbf{x}) = \sum_{j=1}^{n_x - 1} \left(100(x_{j+1} - x_j^2) + (x_j - 1)^2\right) \tag{51}$$

with each $x_j \in [-30, 30]$. Shifted and rotated versions of $f_{13}$ were also used. Both the rotated and shifted versions of the $f_{13}$ used the domain $[-100, 100]$.

$f_{14}$, the Saloman function, defined as

$$f_{14}(\mathbf{x}) = -\cos\left(2\pi \sum_{j=1}^{n_x} x_j^2\right) + 0.1 \sqrt{\sum_{j=1}^{n_x} x_j^2} + 1 \tag{52}$$

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

$f_{15}$, the Schaffer 6 function, defined as

$$f_{15}(\mathbf{x}) = \sum_{j=1}^{n_x} \left(0.5 + \frac{\sin^2(x_j^2 + x_{j+1}^2) - 0.5}{(1 + 0.001(x_j^2 + x_{j+1}^2))^2}\right) \tag{53}$$

with each $x_j \in [-100, 100]$. A rotated and shifted version of $f_{15}$ was also used.

$f_{16}$, the Schwefel 1.2 function, defined as

$$f_{16}(\mathbf{x}) = \sum_{i=1}^{n_x} \left(\sum_{j=1}^{i} x_j\right)^2 \tag{54}$$

with each $x_j \in [-100, 100]$. Shifted and rotated versions of $f_{16}$ were also used. Additionally, a shifted, noisy version of the Schwefel 1.2 function, defined as

$$f_{16}^{ShN}(\mathbf{x}) = \sum_{i=1}^{n_x} \left( \sum_{j=1}^{i} x_j \right)^2 (1 + 0.4|N(0,1)|) \tag{55}$$

was also used, with the same domain as the base function.

$f_{17}$, the Schwefel 2.6 function, defined as

$$f_{17}(\mathbf{x}) = \max_j \{|\mathbf{A}_j \mathbf{x} - \mathbf{B}_j|\} \tag{56}$$

with each $x_j \in [-100, 100]$, each $a_{ij} \in \mathbf{A}$ is uniformly sampled from $U(-500, 500)$ such that $\det(\mathbf{A}) \neq 0$, and each $\mathbf{B}_j = \mathbf{A}_j \mathbf{r}$ where each $r_i \in \mathbf{r}$ is uniformly sampled from $U(-100, 100)$. A shifted version of $f_{17}$ was also used.

$f_{18}$, the Schwefel 2.13 function, defined as

$$f_{18}(\mathbf{x}) = \sum_{j=1}^{n_x} (\mathbf{A}_j - \mathbf{B}_j(\mathbf{x}))^2 \tag{57}$$

with each $x_j \in [-\pi, \pi]$, and

$$\mathbf{A}_j = \sum_{i=1}^{n_x} (a_{ij} \sin(\alpha_i) + b_{ij} \cos(\alpha_i))$$

and

$$\mathbf{B}_j(\mathbf{x}) = \sum_{i=1}^{n_x} (a_{ij} \sin(x_i) + b_{ij} \cos(x_i))$$

where $a_{ij} \in \mathbf{A}, b_{ij} \in \mathbf{B}$, $a_{ij}, b_{ij} \sim U(-100, 100)$, and $\alpha_i \sim U(-\pi, \pi)$. A shifted version of $f_{18}$ was also used.

$f_{19}$, the Schwefel 2.21 function, defined as

$$f_{19}(\mathbf{x}) = \max_j \{|x_j|, 1 \leq j \leq n_x\} \tag{58}$$

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

$f_{20}$, the Schwefel 2.22 function, defined as

$$f_{20}(\mathbf{x}) = \sum_{j=1}^{n_x} |x_j| + \prod_{j=1}^{n_x} |x_j| \tag{59}$$

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

$f_{21}$, the Shubert function, defined as

$$f_{21}(\mathbf{x}) = \prod_{j=1}^{n_x} \left( \sum_{i=1}^{5} (i \cos((i+1)x_j + i)) \right) \tag{60}$$

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

$f_{22}$, the spherical function, defined as

$$f_{22}(\mathbf{x}) = \sum_{j=1}^{n_x} x_j^2 \tag{61}$$

with each $x_j \in [-5.12, 5.12]$. A shifted version of $f_{22}$ was also used.

$f_{23}$, the step function, defined as

$$f_{23}(\mathbf{x}) = \sum_{j=1}^{n_x} (\lfloor x_j + 0.5 \rfloor)^2 \tag{62}$$

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

$f_{24}$, the Vincent function, defined as

$$f_{24}(\mathbf{x}) = -\left(1 + \sum_{j=1}^{n_x} \sin(10\sqrt{x_j})\right) \tag{63}$$

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

$f_{25}$, the Weierstrass function, defined as

$$\begin{aligned} f_{25}(\mathbf{x}) = & \sum_{j=1}^{n_x} \left( \sum_{i=1}^{20} (a^i \cos(2\pi b^i (x_j + 0.5))) \right) \\ & - n_x \sum_{i=1}^{20} (a^i \cos(\pi b^i)) \end{aligned} \tag{64}$$

with each $x_j \in [-0.5, 0.5]$, $a = 0.5$, and $b = 3$. A rotated and shifted version of $f_{25}$ was also used.

$f_{26}$, a shifted expansion of the Griewank and Rosenbrock functions [Eqs. (43) and (51), respectively] with each $x_j \in [-3, 1]$. Note that $f_{26}$ is equivalent to $f_{13}$ from the 2005 CEC benchmark suite.

$f_{27} - f_{37}$, composition functions equivalent to $f_{15} - f_{25}$ from the 2005 CEC benchmark suite. All functions have each $x_j \in [-5, 5]$, with the exception of $f_{37}$ which has each $x_j \in [2, 5]$.

## Appendix 2: Overall ranks using the local-best topology

Table 11 summarizes the results obtained across all benchmark problems using the Mann–Whitney U statistical analysis procedure described in Sect. 5.3. Table 11 clearly indicates that the influence of topology is significant with regard to the performance of the inertia weight strategies. Specifically, the constant and random strategies are no longer the top performing strategies when the local-best topology is considered. Rather, the constant strategy attained a rank of 5 and the random strategy attained a rank of 9 when considering the accuracy performance measure. Despite its relatively poor rank of 13 when using the global-best topology, the PSO-NL strategy showed the best overall accuracy when using the local-best topology. Thus, the poor performance of the adaptive inertia weight strategies observed when the global-best topology was used, does not necessarily hold when the local-best topology is used. Therefore, it can be concluded that the topology should be considered when tuning for optimal performance. This comes as no surprise given that it is well known that the topology should be included as a tuned parameter if optimal performance is desired Engelbrecht (2013a).

**Table 11** Summary of performance across all 60 benchmark problems using the local-best topology

| Strategy | Accuracy | | | Success rate | | | Consistency | | |
|---|---|---|---|---|---|---|---|---|---|
| | Sum | Rank | BRF | Sum | Rank | BRF | Sum | Rank | BRF |
| AIWPSO | 322 | 3 | 16 | 200 | 5 | 31 | 247 | 4 | 18 |
| APSO-VI | −257 | 13 | 4 | −135 | 12 | 22 | −151 | 13 | 8 |
| CDIW-PSO | 325 | 2 | 8 | 257 | 2 | 26 | 264 | 3 | 9 |
| Constant | 281 | 5 | 12 | 94 | 9 | 24 | 144 | 9 | 15 |
| DE-PSO | 658 | 18 | 0 | −413 | 18 | 21 | −503 | 18 | 6 |
| DW-PSO | −509 | 17 | 2 | −294 | 16 | 21 | −291 | 15 | 8 |
| FG-PSO | 217 | 10 | 12 | 106 | 8 | 34 | 188 | 6 | 14 |
| IPSO-LT | −454 | 16 | 3 | −174 | 14 | 24 | −460 | 17 | 10 |
| LD-PSO | 303 | 4 | 5 | **267** | **1** | 25 | 275 | 2 | 10 |
| PSO-NL | **340** | **1** | 8 | 201 | 4 | 23 | **277** | **1** | 10 |
| PSO-NLI | −338 | 14 | 2 | −236 | 15 | 24 | −212 | 14 | 11 |
| PSO-NEIW | 237 | 7 | 3 | 193 | 6 | 24 | 194 | 5 | 10 |
| PSO-OIW | 252 | 6 | 7 | 242 | 3 | 27 | 176 | 7 | 12 |
| PSO-RIW | 223 | 9 | 11 | 58 | 10 | 25 | 97 | 10 | 13 |
| PSO-RBI | −413 | 15 | 0 | −312 | 17 | 21 | −417 | 16 | 7 |
| SRPSO | −67 | 12 | 2 | −140 | 13 | 21 | −56 | 12 | 6 |
| PSO-SIW | −38 | 11 | 4 | −72 | 11 | 23 | 54 | 11 | 8 |
| PSO-LDIW | 234 | 8 | 4 | 158 | 7 | 24 | 174 | 8 | 11 |

Bold entries indicate the strategies with the highest rank based on the sum of wins and losses across all benchmark problems

## References


Bansal, J. C., Singh, P. K., Saraswat, M., Vermam A., Jadon, S. S., & Abraham, A. (2011). Inertia weight strategies in particle swarm. In *Proceedings of the third world congress on nature and biologically inspired computing* (pp. 633–640). IEEE.

Beielstein, T., Parsopoulos, K. E., & Vrahatis, M. N. (2002). *Tuning PSO parameters through sensitivity analysis*. Technical report. Universitat Dortmund.

Bonyadi, M. R., & Michalewicz, Z. (2016). Particle swarm optimization for single objective continuous space problems: A review. *Evolutionary Computation*. doi:10.1162/EVCO_r_00180.

Carlisle, A., & Dozier, G. (2001). An off-the-shelf PSO. In *Proceedings of the workshop on particle swarm optimization* (pp. 1–6). Indianapolis.

Chatterjee, A., & Siarry, P. (2006). Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization. *Computers & Operations Research*, *33*(3), 859–871.

Chauhan, P., Deep, K., & Pant, M. (2013). Novel inertia weight strategies for particle swarm optimization. *Memetic Computing*, *5*(3), 229–251.

Chen, G., Min, Z., Jia, J., & Xinbo, H. (2006). Natural exponential inertia weight strategy in particle swarm optimization. In *Proceedings of the 6th world congress on intelligent control and automation* (Vol. 1, pp. 3672–3675).

Chen, H. H., Li, G. Q., & Liao, H. L. (2009). A self-adaptive improved particle swarm optimization algorithm and its application in available transfer capability calculation. In *Proceedings of the fifth international conference on natural computation* (Vol. 3, pp. 200–205).

Cleghorn, C. W., & Engelbrecht, A. P. (2014a). Particle swarm convergence: An empirical investigation. In *Proceedings of the 2014 IEEE congress on evolutionary computation* (pp. 2524–2530).

Cleghorn, C. W., & Engelbrecht, A. P. (2014b). Particle swarm convergence: Standardized analysis and topological influence. In M. Dorigo, M. Birattari, S. Garnier, H. Hamann, M. de Oca, C. Solnon, & T. Sttzle

(Eds.), *Swarm intelligence* (Vol. 8667, pp. 134–145). Lecture Notes in Computer Science. Springer International Publishing.

Cleghorn, C. W., & Engelbrecht, A. P. (2015). Particle swarm variants: Standardized convergence analysis. *Swarm Intelligence*, *9*(2–3), 177–203.

de Oca, M., Pena, J., Stutzle, T., Pinciroli, C., & Dorigo, M. (2009). Heterogeneous particle swarm optimizers. In *Proceedings of the 2009 IEEE congress on evolutionary computation* (pp. 698–705).

Deep, K., Chauhan, P., & Pant, M. (2011). A new fine grained inertia weight particle swarm optimization. In *Proceedings of the 2011 world congress on information and communication technologies* (pp. 424–429). IEEE.

Eberhart, R., & Shi, Y. (2000). Comparing inertia weights and constriction factors in particle swarm optimization. In *Proceedings of the 2000 IEEE congress on evolutionary computation* (Vol. 1, pp. 84–88). IEEE.

Eberhart, R. C., & Kennedy, J. (1995). A new optimizer using particle swarm theory. In *Proceedings of the sixth international symposium on micro machine and human science* (Vol. 1, pp. 39–43). New York, NY.

Eberhart, R. C., & Shi, Y. (2001). Tracking and optimizing dynamic systems with particle swarms. In *Proceedings of the 2001 IEEE congress on evolutionary computation* (Vol. 1, pp. 94–100). IEEE.

Engelbrecht, A. P. (2012). Particle swarm optimization: Velocity initialization. In *Proceedings of the 2012 IEEE congress on evolutionary computation* (pp. 1–8).

Engelbrecht, A. P. (2013a). Particle swarm optimization: Global best or local best? In *Proceedings of the 2013 BRICS congress on computational intelligence and 11th Brazilian congress on computational intelligence* (pp. 124–135). IEEE.

Engelbrecht, A. P. (2013b). Roaming behavior of unconstrained particles. In *Proceedings of the 2013 BRICS congress on computational intelligence and 11th Brazilian congress on computational intelligence* (pp. 104–111).

Fan, S. K. S., & Chiu, Y. Y. (2007). A decreasing inertia weight particle swarm optimizer. *Engineering Optimization*, *39*(2), 203–228.

Feng, Y., Teng, G. F., Wang, A. X., & Yao, Y. M. (2007). Chaotic inertia weight in particle swarm optimization. In *Proceedings of the second international conference on innovative computing. Information and control* (pp. 475–479). IEEE.

Gao, Y. L., An, X. H., & Liu, J. M. (2008). A particle swarm optimization algorithm with logarithm decreasing inertia weight and chaos mutation. In *Proceedings of the 2008 international conference on computational intelligence and security* (pp. 61–65). IEEE.

Garden, R. W., & Engelbrecht, A. P. (2014). Analysis and classification of optimisation benchmark functions and benchmark suites. In *Proceedings of the 2014 IEEE congress on evolutionary computation* (Vol. 1, pp. 1641–1649).

Harrison, K. R., Engelbrecht, A. P., & Ombuki-Berman, B. M. (2016). The sad state of self-adaptive particle swarm optimizers. In *Proceedings of the 2016 IEEE congress on evolutionary computation* (pp. 431–439). IEEE.

Hu, J. Z., Xu, J., Wang, J. Q., & Xu, T. (2009). Research on particle swarm optimization with dynamic inertia weight. In *Proceedings of the 2009 international conference on management and service science* (Vol. 3, pp. 1–4).

Jiao, B., Lian, Z., & Gu, X. (2008). A dynamic inertia weight particle swarm optimization algorithm. *Chaos, Solitons and Fractals*, *37*(3), 698–705.

Kennedy, J., & Eberhart, R. C. (1995). Particle swarm optimization. In *Proceedings of the 1995 IEEE international joint conference on neural networks* (Vol. IV, pp 1942–1948).

Kentzoglanakis, K., & Poole, M. (2009). Particle swarm optimization with an oscillating inertia weight. In *Proceedings of the 11th annual conference on genetic and evolutionary computation* (pp. 1749–1750). ACM.

Lei, K., Qiu, Y., & He, Y. (2006). A new adaptive well-chosen inertia weight strategy to automatically harmonize global and local search ability in particle swarm optimization. In *Proceedings of the 1st international symposium on systems and control in aerospace and astronautics* (pp. 977–980). IEEE.

Leonard, B. J., & Engelbrecht, A. P. (2013). On the optimality of particle swarm parameters in dynamic environments. In *Proceedings of the 2013 IEEE congress on evolutionary computation* (pp. 1564–1569). doi:10.1109/CEC.2013.6557748.

Li, C., Yang, S., & Nguyen, T. T. (2012). A self-learning particle swarm optimizer for global optimization problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *42*(3), 627–646.

Li, Z., & Tan, G. (2008). A self-adaptive mutation-particle swarm optimization algorithm. In *Proceedings of the fourth international conference on natural computation* (Vol. 1, pp. 30–34). IEEE.

Liu, B., Wang, L., Jin, Y. H., Tang, F., & Huang, D. X. (2005). Improved particle swarm optimization combined with chaos. *Chaos, Solitons & Fractals*, *25*(5), 1261–1271.

Liu, Q., Wei, W., Yuan, H., Zhan, Z. H., & Li, Y. (2016). Topology selection for particle swarm optimization. *Information Sciences*, *363*(2015), 154–173. doi:10.1016/j.ins.2016.04.050.

Lynn, N., & Suganthan, P. N. (2015). Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation. *Swarm and Evolutionary Computation*, *24*, 11–24.

Mascia, F., Pellegrini, P., Stützle, T., & Birattari, M. (2014). An analysis of parameter adaptation in reactive tabu search. *International Transactions in Operational Research*, *21*(1), 127–152.

Nepomuceno, F. V., & Engelbrecht, A. P. (2013). A self-adaptive heterogeneous PSO for real-parameter optimization. In *Proceedings of the 2013 IEEE congress on evolutionary computation* (pp. 361–368). IEEE.

Nickabadi, A., Ebadzadeh, M. M., & Safabakhsh, R. (2011). A novel particle swarm optimization algorithm with adaptive inertia weight. *Applied Soft Computing*, *11*(4), 3658–3670.

Pandey, B. B., Debbarma, S., & Bhardwaj, P. (2015). Particle swarm optimization with varying inertia weight for solving nonlinear optimization problem. In *Proceedings of the 2015 international conference on electrical, electronics, signals, communication and optimization* (pp. 1–5). IEEE.

Panigrahi, B. K., Ravikumar Pandi, V., & Das, S. (2008). Adaptive particle swarm optimization approach for static and dynamic economic load dispatch. *Energy Conversion and Management*, *49*(6), 1407–1415.

Pellegrini, P., Stützle, T., & Birattari, M. (2012). A critical analysis of parameter adaptation in ant colony optimization. *Swarm Intelligence*, *6*(1), 23–48.

Poli, R. (2009). Mean and variance of the sampling distribution of particle swarm optimizers during stagnation. *IEEE Transactions on Evolutionary Computation*, *13*(4), 712–721.

Poli, R., & Broomhead, D. (2007). 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* (pp. 134–141). New York, NY: ACM.

Salomon, R. (1996). Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions. *BioSystems*, *39*(3), 263–278.

Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. In *Proceedings of the 1998 IEEE international conference on evolutionary computation*, (pp. 69–73).

Shi, Y., & Eberhart, R. C. (1999). Empirical study of particle swarm optimization. In *Proceedings of the 1999 IEEE congress on evolutionary computation* (Vol. 3, pp. 1945–1950). IEEE.

Suganthan, P. N., Hansen, N., Liang, J. J., Deb, K., Chen, Y., & Auger, A., et al. (2005). *Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization*. Technical report. Nanyang Technological University.

Taherkhani, M., & Safabakhsh, R. (2016). A novel stability-based adaptive inertia weight for particle swarm optimization. *Applied Soft Computing*, *38*(4), 281–295.

Tanweer, M. R., Suresh, S., & Sundararajan, N. (2015). Self regulating particle swarm optimization algorithm. *Information Sciences*, *294*, 182–202.

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

Van Den Bergh, F., & Engelbrecht, A. P. (2006). A study of particle swarm optimization particle trajectories. *Information Sciences*, *176*(8), 937–971.

Van Zyl, E., & Engelbrecht, A. (2014). Comparison of self-adaptive particle swarm optimizers. In *Proceedings of the 2014 IEEE symposium on swarm intelligence* (pp. 48–56).

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

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

Yang, C., Gao, W., Liu, N., & Song, C. (2015). Low-discrepancy sequence initialized particle swarm optimization algorithm with high-order nonlinear time-varying inertia weight. *Applied Soft Computing*, *29*, 386–394.