

Particle Swarm Optimization with an Oscillating Inertia Weight

Kyriakos Kentzoglanakis

School of Computing
University of Portsmouth
UK

kyriakos.kentzoglanakis@port.ac.uk

Matthew Poole

School of Computing
University of Portsmouth
UK

matthew.poole@port.ac.uk

ABSTRACT

In this paper, we propose an alternative strategy of adapting the inertia weight parameter during the course of particle swarm optimization. Three oscillating inertia weight functions of time are presented and their influence on the performance and success of the algorithm is investigated by optimizing six standard benchmark functions. Instances of the algorithm that make use of inertia weight adaptation schemes from the literature are also tested for comparison purposes. The results show that the proposed functions are competitive and, in some cases, outperform the established inertia weight functions, in terms of consistency and speed of convergence.

Conference Track

Ant Colony Optimization and Swarm Intelligence

Categories and Subject Descriptors

G.1.6 [Numerical Analysis]: Optimization—*Global Optimization*; I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods and Search—*Heuristic methods*

General Terms

Algorithms, Performance, Experimentation

Keywords

particle swarm optimization, inertia weight

1. INTRODUCTION

Early simulations of movement of individual entities in animal societies, such as simulated bird flocks [18, 9] and fish schools [10], demonstrated that collective behaviour and self-organization in such systems emerge from local interactions between these simple entities, rather than from centralized control.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'09, July 8–12, 2009, Montréal Québec, Canada.
Copyright 2009 ACM 978-1-60558-505-5/09/07 ...\$5.00.

Particle swarm optimization (PSO) is a stochastic, global optimization technique that exploits this empirical observation, for the purposes of problem solving [12]. As such, the progression of the swarm towards finding better solutions is not expressed algorithmically but, rather, emerges as a result of simple particle interactions.

Despite the fact that what can be considered as a general, standard PSO version [2] is, essentially, parameter free, there have been numerous efforts to investigate the behaviour of the algorithm under different parametric conditions. Two of the most studied aspects of PSO are the swarm communication topology and the inertia weight parameter.

The swarm's communication topology defines neighborhoods, where information concerning discovered solutions is shared among particles and is used to calculate their subsequent positions in the search space.

The inertia weight parameter, introduced in [19], defines the impact of each particle's previous velocity to the current one and, according to Poli et al., it can be interpreted as the fluidity of the medium in which particles move [17].

The inertia weight effectively controls the scope of the search [20], what is frequently referred to as the balance between searching for an optimal region (exploration) and searching for an optimal point within the region (exploitation) [13].

In this paper, we investigate the impact of a non-monotonic, time varying inertia weight parameter to the objective of guiding the search process, so as to reduce the likelihood of premature convergence by controlling the dynamics of the balance between global and local search, exploration and exploitation.

Two sinusoidal inertia weight functions of time are proposed; one with a constant amplitude and one with a linearly decreasing amplitude. Also the usage of a step function with the same frequency characteristics as the sinusoidal functions is explored.

The performance of the proposed inertia weight functions of time is assessed by optimizing six well-known benchmark functions. Results are also provided using standard methods of adapting the inertia weight, for the purpose of comparison.

Similar approaches have been implemented in the context of simulated annealing [14], a stochastic global optimization technique inspired by the physical process of annealing, i.e. the heating and subsequent cooling of solid materials with the objective of coercing them into a high order, low energy state. The behaviour of the algorithm depends on the temperature parameter, which typically decreases over

Description	Name	Function
Constant ω (value from [19])	const	$\omega(t) = 0.7298$
Linearly decreasing ω [20]	lin dec	$\omega(t) = \omega_{min} + (\omega_{max} - \omega_{min})\frac{S-t}{S}$
Nonlinearly decreasing ω [3]	nonlin dec	$\omega(t) = \omega_{min} + (\omega_{max} - \omega_{min})\left[\frac{S-t}{S}\right]^n$ with $n = 1.2$
Linearly increasing ω [22]	lin inc	$\omega(t) = \omega_{max} - (\omega_{max} - \omega_{min})\frac{S-t}{S}$
Random ω [7]	rnd	$\omega(t) = U(\omega_{min}, \omega_{max})$

where $U(a, b)$ a random number drawn from a uniform distribution in $[a, b]$.

Table 1: The inertia weight update functions $\omega_i(t)$ from the literature, that were also tested in the experiments. PSO iterations are denoted by t and the total number of iterations for which the inertia weight is allowed to vary, by S .

time, as is the case with the inertia weight in PSO. However, non-monotonic, time-varying cooling schedules have been proposed and evaluated [8, 16, 15], as well as a sinusoidal temperature update function [1], demonstrating improved results compared to standard, decreasing cooling schedules.

The next section gives a summary of the PSO algorithm and reviews existing inertia weight update strategies. In section 3, the rationale behind our approach is presented and three oscillating inertia weight functions are proposed and explained. The experimental framework is laid out in section 4 and the results are discussed in section 5, before a conclusion is reached in section 6.

2. BACKGROUND

A swarm is a collection of candidate solutions (particles) to a well-defined problem, that are represented as points in the N -dimensional Euclidean space. The quality (fitness) of a particle is determined by evaluating its position using the problem-specific objective function.

Each particle i is characterized by a position vector \vec{x}_i , a velocity vector \vec{v}_i and a vector \vec{p}_i that serves as memory of the best position in terms of fitness that the particle has, thus far, encountered.

Particles interact by communicating their best discovered positions, to other particles within a neighborhood. In what is referred to as a global best (gbest) topology, this neighborhood comprises the entire swarm population, while in a local best (lbest) topology, the swarm forms a ring lattice and each particle communicates with its K adjacent neighbours.

Swarms connected using the gbest topology tend to converge faster than those using the lbest topology, but the latter tends to provide more flexibility in terms of escaping local optima and avoiding premature convergence [11]. In general, particle connectivity greatly affects the performance of PSO, since it defines the way information flows within the swarm. However the consensus seems to be that no topology can be clearly characterised as being optimal for all problems.

Particles move in the search space by stochastically updating their velocity and position vectors, attracted by their own best position \vec{p}_i , as well as their neighborhood's best

position \vec{p}_g , according to the equations:

$$\vec{v}_{t+1} = \omega\vec{v}_t + \vec{U}(0, \phi_1) \otimes (\vec{p}_i - \vec{x}_t) + \vec{U}(0, \phi_2) \otimes (\vec{p}_g - \vec{x}_t) \quad (1)$$

$$\vec{x}_{t+1} = \vec{x}_t + \vec{v}_{t+1} \quad (2)$$

where ω is the inertia weight parameter and ϕ_1 and ϕ_2 are the particle's acceleration coefficients that control the magnitude of stochastic attraction towards \vec{p}_i and \vec{p}_g respectively. Vector $\vec{U}(0, \phi_i)$ contains random numbers drawn from a uniform distribution in $[0, \phi_i]$. The operator \otimes denotes element-wise multiplication.

In an attempt to elucidate the process of convergence and eliminate the need for particle velocity constraints, Clerc and Kennedy [5] proposed and analysed the use of a constriction coefficient χ , instead of ω , which multiplies all three components of the velocity update equation:

$$\vec{v}_{t+1} = \chi \left(\vec{v}_t + \vec{U}(0, \phi_1)(\vec{p}_i - \vec{x}_t) + \vec{U}(0, \phi_2)(\vec{p}_g - \vec{x}_t) \right) \quad (3)$$

$$\chi = \frac{2}{\phi - 2 + \sqrt{\phi^2 - 4\phi}} \quad (4)$$

where $\phi = \phi_1 + \phi_2$, with $\phi > 4$. Setting $\phi_1 = \phi_2 = 2.05$, the value $\chi = 0.7298$ is derived.

It is worth noting that the constricted version is equivalent to the inertia weight version, since equations 1 and 3 can be transformed to one another by mapping ω to χ and ϕ_i to $\chi\phi_i$ [6]. Therefore, the corresponding values for the inertia weight version, as expressed by equation 1, are $\omega = 0.7298$ and $\phi_1 = \phi_2 = 2.05 \times 0.7298 = 1.496$.

A widely used strategy for the inertia weight ω is to maintain a constant value over the algorithm's iterations, as derived from Clerc and Kennedy's analysis. However, Shi and Eberhart [20] argued that the inertia weight effectively controls the balance between exploration and exploitation in searching and, therefore, it is reasonable for this balance to be adjusted during the course of optimization.

For this reason, they proposed the use of a linearly decreasing time-varying inertia weight, where a relatively large value of ω promotes search space exploration during the early stages of the algorithm, by encouraging particles to cover wider search areas. As the value of ω decreases, the search is guided from this initial, exploratory phase, towards

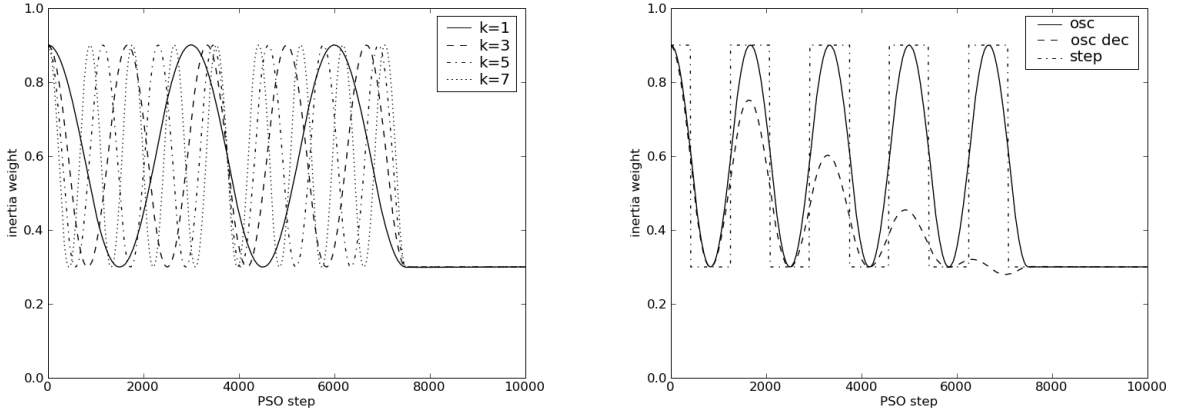


Figure 1: On the left, the behaviour of equation 5 for different values of k . On the right, the behaviour of the three oscillating inertia weight functions of time, as detailed in equations 5 (osc), 8 (osc dec) and 9 (step), for $k = 3$. For both figures, $\omega_{min} = 0.3$, $\omega_{max} = 0.9$ and $S = 10000$.

the refinement of discovered solutions (local search) near the end.

The same authors also showed that a PSO algorithm with a random inertia weight component was able to track a nonlinear dynamic system [7]. In optimizing such systems, an informed decision can not be made on whether exploration or exploitation is preferable at any given time. As such, a random inertia weight, uniformly distributed within a proper range, makes this decision stochastic.

A somewhat different interpretation suggests that the value of the inertia weight is not associated with the scope but rather with the stability of the search [22]. A linearly increasing time-varying inertia weight, in this context, serves to guide the algorithm from an initial turbulent state (low inertia weight) towards stability (high inertia weight) at later stages.

Other approaches to adjusting the inertia weight include a nonlinearly decreasing ω function [3] and a fuzzy system for dynamic adaptation of ω [21].

3. APPROACH

The concept of an oscillating inertia weight implements a strategy whereby a wave of exploration (global search), followed by a wave of exploitation (local search), form a cycle which is repeated during the optimization process. This way, the swarm periodically transitions from exploratory to exploitative states of search.

This temporal behaviour can be implemented by means of an inertia weight function $\omega(t)$, where t denotes PSO iterations, that oscillates between an upper bound ω_{max} and a lower bound ω_{min} .

In this paper, we propose and investigate the behaviour of three such functions: a sinusoidal function with a constant amplitude, a sinusoidal function with a linearly decreasing amplitude and a step function with the same frequency characteristics as the sinusoidal functions.

The first of the three, a sinusoidal with a constant amplitude is expressed by the following equation:

$$\omega(t) = \frac{\omega_{min} + \omega_{max}}{2} + \frac{\omega_{max} - \omega_{min}}{2} \cos\left(\frac{2\pi t}{T}\right) \quad (5)$$

As implied in the previous section, a typical requirement is that the algorithm starts at ω_{max} with high exploration capabilities and ends at ω_{min} with high exploitation tendencies, hence such a wave function should complete $\frac{3}{2} + k$ cycles within a PSO run, where $k \in \mathbb{N}$ is a parameter that controls the frequency of oscillation.

Consequently, the period T of oscillation, i.e. the number of PSO iterations within which the inertia weight function completes a full cycle, is given by:

$$T = \frac{2S_1}{3 + 2k} \quad (6)$$

where S_1 is the number of iterations for which the inertia weight is allowed to oscillate. We set $S_1 = 3S/4$ iterations, where S is the total number of iterations for a single PSO run. During the final $S_2 = S/4$ iterations, the inertia weight value is kept constant at $\omega(t) = \omega_{min}$, in order to promote the refinement of discovered solutions. This behaviour is displayed in figure 1 (on the left).

The second proposed function is a sinusoidal with a linearly decreasing amplitude, so that the swarm's transition between global and local search becomes subtler over the course of optimization. Effectively, ω_{max} in equation 5, becomes a linearly decreasing function ω'_{max} of time:

$$\omega'_{max}(t) = \left(1 - \frac{t}{S}\right)\omega_{max} \quad (7)$$

and the value of inertia weight now oscillates between a static ω_{min} and a dynamic $\omega'_{max}(t)$, according to the equation:

$$\omega(t) = \frac{\omega_{min} + \omega'_{max}(t)}{2} + \frac{\omega'_{max}(t) - \omega_{min}}{2} \cos\left(\frac{2\pi t}{T}\right) \quad (8)$$

The third proposed inertia weight function is a step function with the same frequency characteristics as in the sinusoidal case. In this case, exploration and exploitation tendencies are discretely modelled, according to the equation:

$$\omega(t) = \begin{cases} \omega_{max} & \text{if } \cos\left(\frac{2\pi t}{T}\right) \geq 0 \\ \omega_{min} & \text{otherwise} \end{cases} \quad (9)$$

Name	Function	Feasible Bounds	Initialization	D	Goal
sphere	$f_1 = \sum_{i=1}^D x_i^2$	$(-100, 100)^D$	$(50, 100)^D$	30	0.01
rosenbrock	$f_2 = \sum_{i=1}^{D-1} \{100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\}$	$(-30, 30)^D$	$(15, 30)^D$	30	100
rastrigin	$f_3 = \sum_{i=1}^D \{x_i^2 - 10 \cos(2\pi x_i) + 10\}$	$(-5.12, 5.12)^D$	$(2.56, 5.12)^D$	30	100
griewank	$f_4 = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos(\frac{x_i}{\sqrt{i}}) + 1$	$(-600, 600)^D$	$(300, 600)^D$	30	0.05
schaffer's f6	$f_5 = 0.5 + \frac{(\sin(\sqrt{x^2+y^2})^2 - 0.5)}{(1+0.001(x^2+y^2))^2}$	$(-100, 100)^2$	$(50, 100)^2$	2	10^{-5}
schwefel 1.2	$f_6 = \sum_{i=1}^D (\sum_{j=1}^i x_j)^2$	$(-100, 100)^D$	$(50, 100)^D$	30	100

Table 2: The six benchmarks functions that were used in the tests.

The temporal behaviour of the three proposed functions is depicted in figure 1 (on the right).

4. METHODS

The conducted tests investigate the influence of the proposed oscillating inertia weight functions, described by equations 5 (osc), 8 (osc dec) and 9 (step) to PSO. For comparison purposes, five inertia weight functions from the literature, summarized in table 1, were included in the tests. These tests consist of optimizing six standard benchmark functions, presented in table 2.

For the experiments, the swarm population size was set to 20 particles, particle positions were assymmetrically initialized and PSO was allowed to run for $S = 10000$ iterations. For the first $S_1 = 3S/4 = 7500$ iterations, the inertia weight was allowed to vary according to the equation under consideration, while during the final $S_2 = S/4 = 2500$ iterations, the inertia weight was kept constant at its last assumed value (except for the random inertia weight function where the value ω_{min} was used for this final stage).

The empirical values $\omega_{min} = 0.3$ and $\omega_{max} = 0.9$ were chosen as the range of the inertia weight. The acceleration coefficients were set to $\phi_1 = \phi_2 = 1.496$, following Clerc and Kennedy's recommendation [5].

Enforcement of x_{max} was also performed, according to which if a particle position vector component exceeds the objective function's feasible bounds, it is brought back to the exceeded boundary and the corresponding velocity vector component is set to zero.

Two different sets of experiments were conducted, each varying the swarm topology in use. The first set uses a static local best topology, considered as standard [2], according to which the swarm forms a ring lattice and each particle communicates with its $K = 2$ adjacent particles.

The second set uses a dynamic random neighborhood selection strategy, whereby at each iteration, a particle randomly chooses K other particles to form its information neighborhood. For the results presented here, the value $K = 3$ is used.

Table 3 summarizes the results, which are averaged across 50 independent PSO runs. The table records the average optimization success rate according to the function goals defined in table 2, as well as the average number of iterations needed to satisfy these goals. Parameter k of equation 6 has

been set to $k = 7$ for the three oscillatory inertia weight functions (osc, osc dec and step).

Ring topology				
	const	lin dec	nonlin dec	lin inc
f_1	100% (700)	100% (620)	100% (610)	100% (560)
f_2	98% (900)	96% (880)	100% (825)	100% (1060)
f_3	12% (2440)	2% (480)	2% (770)	0% (-)
f_4	100% (685)	100% (630)	100% (610)	30% (500)
f_5	92% (1225)	98% (740)	98% (900)	20% (8645)
f_6	52% (2855)	68% (2230)	80% (2085)	64% (3130)
Random neighborhood topology				
	const	lin dec	nonlin dec	lin inc
f_1	96% (1600)	100% (1070)	98% (1000)	100% (1480)
f_2	96% (2240)	98% (1360)	100% (1200)	96% (2225)
f_3	72% (3055)	78% (1660)	80% (1445)	2% (250)
f_4	94% (1680)	96% (1050)	98% (1020)	10% (1400)
f_5	100% (410)	100% (365)	100% (410)	58% (310)
f_6	22% (8800)	82% (3340)	84% (3170)	52% (2475)
	osc	osc dec	step	rnd
f_1	100% (565)	100% (570)	74% (2100)	100% (360)
f_2	96% (1120)	98% (1140)	54% (4710)	98% (480)
f_3	96% (530)	96% (505)	100% (415)	46% (830)
f_4	72% (565)	72% (495)	20% (1630)	88% (350)
f_5	100% (535)	100% (440)	100% (555)	92% (325)
f_6	80% (2330)	80% (2185)	80% (3935)	76% (1730)

Table 3: Average optimization success rates across 50 PSO runs, for the function goals defined in table 2. The average number of iterations required to reach the optimization goal is shown in parentheses. For the results shown at the top panel, a local best (ring) topology is used. For the results shown at the bottom panel, a random neighborhood topology of size 3 is used. The frequency parameter k from equation 6 is set to $k = 7$ for the oscillating functions.

5. DISCUSSION

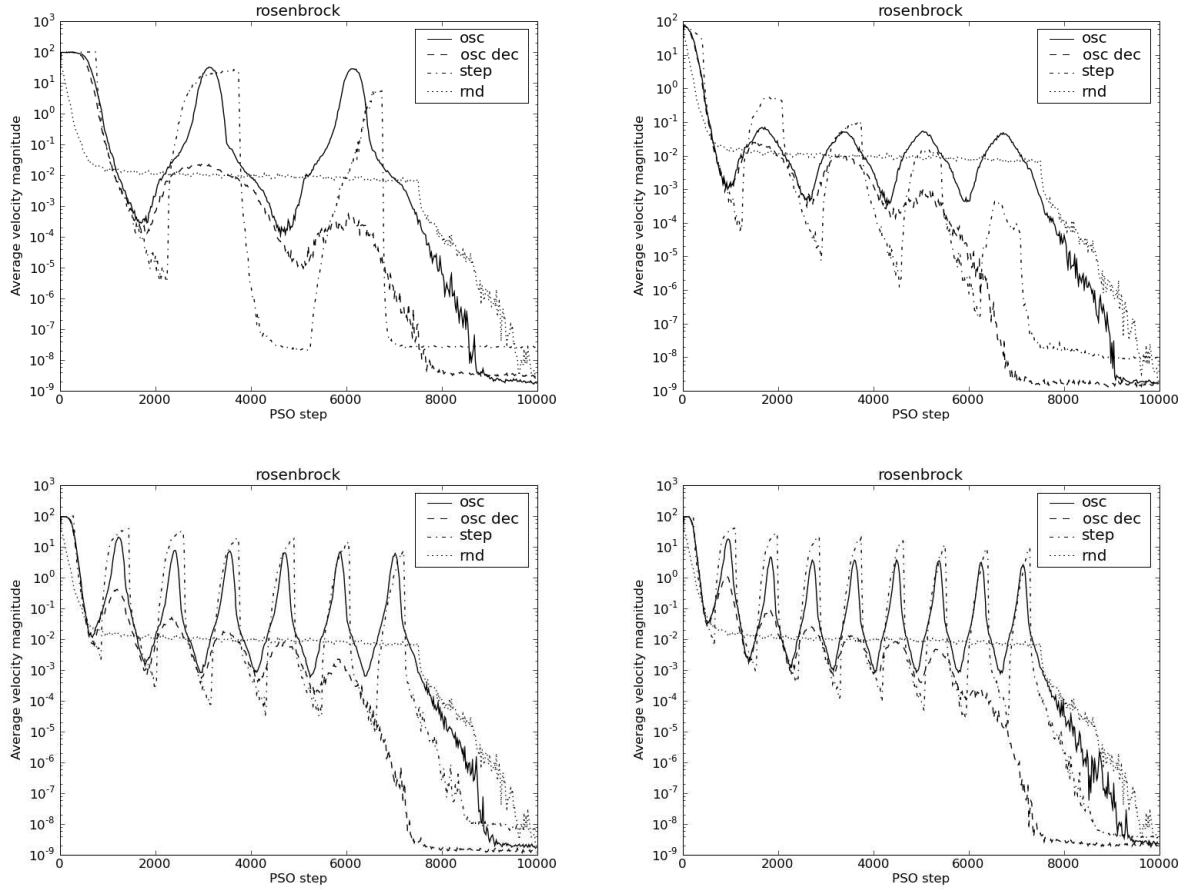


Figure 2: Examples that showcase the effects of different non-monotonic inertia weight functions on the average particle velocity vector magnitude, for the rosenbrock function. From left to right and from top to bottom, parameter k from equation 6 is set to 1, 3, 5 and 7 respectively.

The results indicate that the performance of the oscillating functions is generally competitive to the inertia weight functions from the literature that were also tested. Among the latter, the constant (const) and decreasing (lin dec and nonlin dec) functions achieve satisfactory success rates with good convergence speeds, especially when the ring topology is in use (with the exception of the rastrigin function). Among the decreasing functions, the nonlinear version is slightly better than the linear one. The behaviour of the linearly increasing function is somewhat erratic, while the random inertia weight appears to be more successful when the random neighborhood topology is deployed.

Among the oscillating functions, the performance of osc and osc dec are quite comparable and competitive to the decreasing functions. The step function is less consistent in its success rates but performs particularly well on the rastrigin problem, unlike all other inertia weight functions, especially when using the ring topology.

The switch from the ring topology to the random neighborhood has the effect of generally enhancing the consistency of optimization success rates across all benchmark functions. The success rates for the rastrigin function in particular are significantly increased, compared to those using the ring topology.

Considering the trade-offs between optimization success rates, speed of convergence and consistency of performance across all benchmark functions, the sinusoidal inertia weight function with a constant amplitude and the nonlinearly decreasing function produce the best set of results, when using a random neighborhood topology, with the sinusoidal function converging rather faster.

A comparison between the number of iterations required to satisfy the set goals (shown in parentheses in table 3) and the inertia weight oscillation cycles in figure 1 reveals that, in many cases, these goals are achieved before the oscillatory function completes a full cycle. Indeed, in such cases, an oscillation half-cycle corresponds to a rapid nonlinear inertia weight decrease (exploitation), which appears to be enough to satisfy the required objective function goal.

The influence of the repeated oscillation cycles to the average best particle fitness is displayed in figure 3, for all benchmark functions. In addition to the oscillating functions, the average best particle fitness using the nonlinearly decreasing function has also been plotted for comparison purposes.

The inertia weight adaptation strategy is one the factors influencing the momentum of particles as they fly in the search space attracted by their own and their neighborhood's best solutions. In this context, an oscillating inertia weight

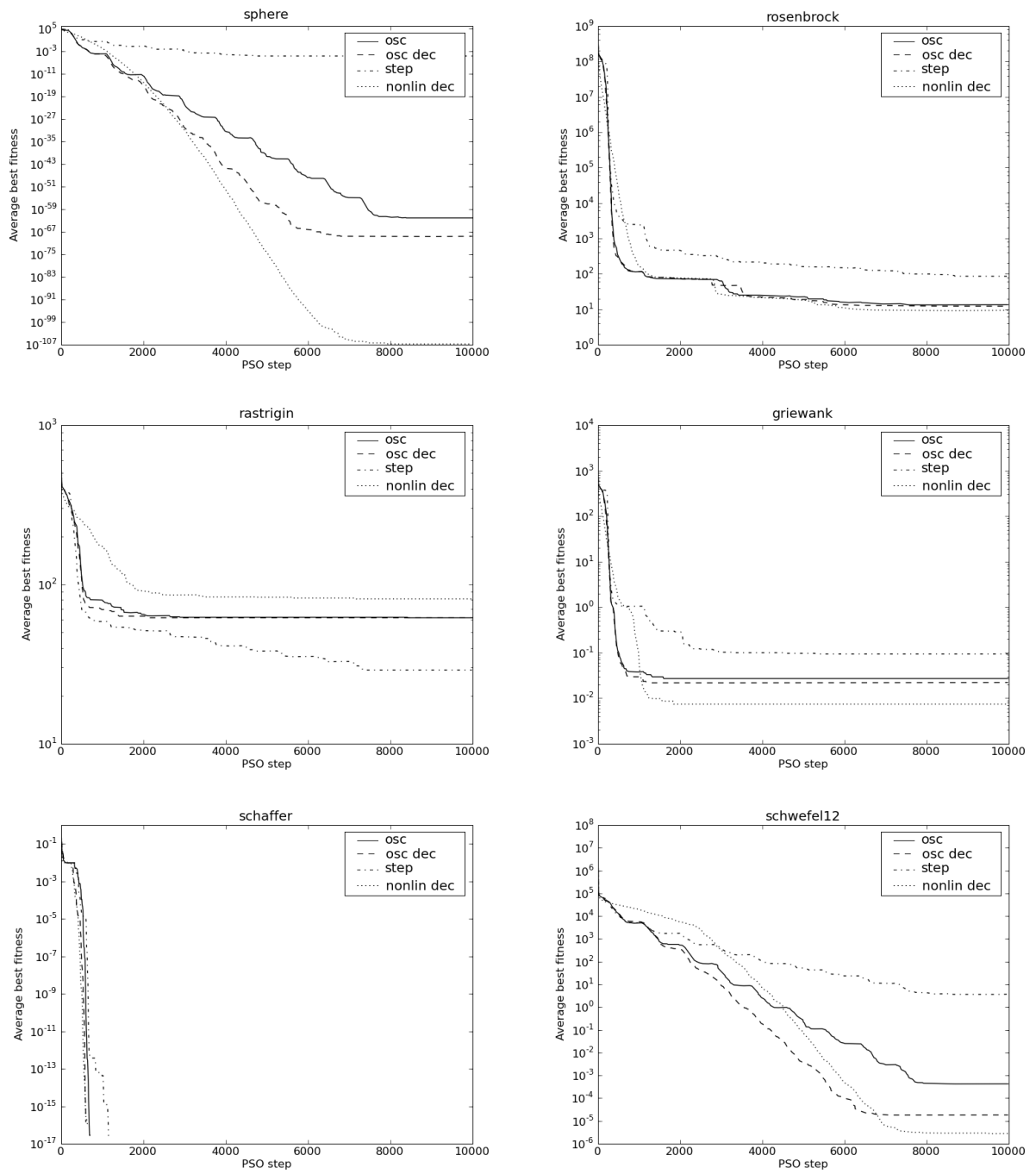


Figure 3: Average best particle fitness across 50 PSO runs for the six benchmark functions, using the random neighborhood topology, for selected inertia weight functions. The frequency parameter k from equation 6 is set to $k = 7$ for the oscillating functions.

serves the purpose of periodically promoting and suspending particle momentum, or driving the swarm towards global and local search, respectively. This accelerate/decelerate effect is demonstrated in figure 2, where average particle velocity vector magnitude cycles can be seen to correspond to inertia weight function cycles.

Different frequency settings of inertia weight oscillation, by adjusting parameter k in equation 6, were tested but no definite conclusions could be drawn. Variations to the value of k did not appear to have a noticeable impact to the results, as presented in table 3. Only the step inertia weight function benefited from a frequency increase and improved its success rates considerably, when the random neighborhood topology was in use.

Naturally, the momentum of particles is also dependent on the structure and possible dynamics of the swarm's topology, i.e. the kind of information communicated between particles within a neighborhood, what Kennedy and Mendes call neighborhood consensus [13]. Inertia weight adaptation strategies can, thus, be viewed as a set of global, secondary filters that scale the results of individual particle interactions in different ways.

Finally, we note that none of the adaptation strategies presented and tested in this paper can be considered as being intelligent, in the sense that they do not dynamically respond to particular conditions that would suggest a shift between a local and a global mode of search, but rather enforce a scheme of a globally predefined balance between the two.

6. CONCLUSION

Inertia weight adaptation during the course of optimization is a significant aspect affecting the behaviour of PSO, both in terms of the ability to satisfy set goals and the speed of convergence.

In this paper, we proposed a set of three oscillating inertia weight functions of time that periodically alternate between global and local search waves, implementing a different kind of balance between the swarm's tendencies towards exploration and exploitation of the search space.

The proposed functions, as well as a number of inertia weight functions from the literature, were tested on the optimization of six standard benchmark problems. The oscillating functions were found to be generally competitive and, in some cases, outperformed established inertia weight adaptation schemes. When considering particular experimental settings, such as a random neighborhood communication topology, the sinusoidal functions achieved very good success rates and convergence speeds, consistently across all benchmark functions.

Further work is needed to investigate how and why the swarm responds to different inertia weight frequency settings, as regulated by parameter k in equation 6. In the suite of tests performed for the purposes of this paper, no definite conclusions could be drawn regarding this issue.

Other non-monotonic inertia weight functions, perhaps oscillating between dynamic upper and lower bounds can also be developed with the possible incorporation of heuristics such as gradient estimations [4] or best performance evaluations [21], so that the inertia weight can adapt according to emerging swarm conditions.

As is the case with the swarm's communication topology, there seems to be no one-size-fits-all, optimal strategy of

adapting the inertia weight during the course of optimization and problem-specific characteristics have to be considered before a decision regarding the selection of a such strategy is made.

7. REFERENCES

- [1] A. Bolte and U. W. Thonemann. Optimizing simulated annealing schedules with genetic programming. *European Journal of Operational Research*, 92(2):402–416, 1996.
- [2] D. Bratton and J. Kennedy. Defining a standard for particle swarm optimization. In *IEEE Swarm Intelligence Symposium*, pages 120–127, 2007.
- [3] A. Chatterjee and P. Siarry. Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization. *Computers & Operations Research*, 33(3):859–871, March 2006.
- [4] M. Clerc. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In *Congress on Evolutionary Computation*, volume 3, pages 1951–1957, 1999.
- [5] M. Clerc and J. Kennedy. The particle swarm - explosion, stability and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1):58–73, 2002.
- [6] R. C. Eberhart and Y. Shi. Comparing inertia weights and constriction factors in particle swarm optimization. In *Proceedings of the 2000 Congress on Evolutionary Computation*, volume 1, pages 84–88, 2000.
- [7] R. C. Eberhart and Y. Shi. Tracking and optimizing dynamic systems with particle swarms. In *Proceedings of the 2001 Congress on Evolutionary Computation*, volume 1, pages 94–100, 2001.
- [8] B. Hajek and G. Sasaki. Simulated annealing—to cool or not. *Syst. Control Lett.*, 12(4):443–447, 1989.
- [9] F. Heppner and U. Grenander. A stochastic nonlinear model for coordinated bird flocks. In E. Krasner, editor, *The ubiquity of chaos*, pages 233–238. AAAS Publications, 1990.
- [10] A. Huth and C. Wissel. The simulation of the movement in fish schools. *Journal of Theoretical Biology*, 156(7):365–385, 1992.
- [11] J. Kennedy. Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 3, pages 1931–1938, 1999.
- [12] J. Kennedy and R. Eberhart. Particle swarm optimization. In *IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948, 1995.
- [13] J. Kennedy and R. Mendes. Neighborhood topologies in fully informed and best-of-neighborhood particle swarms. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 36(4):515–519, 2006.
- [14] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
- [15] J. W. Mann and G. D. Smith. A comparison of heuristics for telecommunications traffic routing. In V. J. Rayward-Smith, I. H. Osman, C. R. Reeves, and

G. D. Smith, editors, *Modern Heuristic Search Methods*, pages 235–254. Wiley, 1996.

- [16] I. H. Osman and N. Christofides. Capacitated clustering problems by hybrid simulated annealing and tabu search. *International Transactions in Operational Research*, 1(3):317–336, 1994.
- [17] R. Poli, J. Kennedy, and T. Blackwell. Particle swarm optimization. *Swarm Intelligence*, 1(1):33–57, 2007.
- [18] C. W. Reynolds. Flocks, herds and schools: a distributed behavioral model. *Computer Graphics*, 21(4):25–34, 1987.
- [19] Y. Shi and R. Eberhart. A modified particle swarm optimizer. In *Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence*, pages 69–73, 1998.
- [20] Y. Shi and R. Eberhart. Parameter selection in particle swarm optimization. In *Evolutionary Programming VII*, pages 591–600. Springer, 1998.
- [21] Y. Shi and R. C. Eberhart. Fuzzy adaptive particle swarm optimization. In *Congress on Evolutionary Computation*, volume 1, pages 101–106, 2001.
- [22] Y. L. Zheng, L. H. Ma, L. Y. Zhang, and J. X. Qian. On the convergence analysis and parameter selection in particle swarm optimization. In *International Conference on Machine Learning and Cybernetics*, volume 3, pages 1802–1807, 2003.