


Stochastic stability of particle swarm optimisation

Adam Erskine¹ · Thomas Joyce¹ ·
J. Michael Herrmann¹ 

Received: 3 October 2016 / Accepted: 25 October 2017
© The Author(s) 2017. This article is an open access publication

Abstract Particle swarm optimisation (PSO) is a metaheuristic algorithm used to find good solutions in a wide range of optimisation problems. The success of metaheuristic approaches is often dependent on the tuning of the control parameters. As the algorithm includes stochastic elements that effect the behaviour of the system, it may be studied using the framework of random dynamical systems (RDS). In PSO, the swarm dynamics are quasi-linear, which enables an analytical treatment of their stability. Our analysis shows that the region of stability extends beyond those predicted by earlier approximate approaches. Simulations provide empirical backing for our analysis and show that the best performance is achieved in the asymptotic case where the parameters are selected near the margin of instability predicted by the RDS approach.

Keywords Particle swarm optimisation · Criticality · Random dynamical systems · Random matrix products · Parameter selection

1 Particle swarm optimisation

Particle swarm optimisation (PSO) (Kennedy and Eberhart 1995) is a metaheuristic algorithm which is widely used in search and optimisation tasks. It aims at locating solutions to problems that may be characterised by high dimensionality, heterogeneity, the presence of many suboptimal solutions, and the absence of gradient information. An optimal solution is a global minimum of a given cost function (or, depending on problem, a global maximum) the domain of which is explored by a swarm of particles. In many problems, where PSO is applied, also solutions with near-optimal costs can be considered as good.

The number of particles N is quite low in most applications, usually amounting to a few dozens. Each particle represents a potential solution and shares knowledge about the currently

✉ J. Michael Herrmann
michael.herrmann@ed.ac.uk

¹ Institute for Perception, Action and Behaviour, School of Informatics, The University of Edinburgh, Edinburgh, Scotland, UK

known overall best solution (*global best*) and also retains a memory of the best solution it previously has encountered itself (*personal best*).

The success of the PSO algorithm in locating good solutions depends on the dynamics of the particles in the search space of the problem. In contrast, for example, to many evolution strategies, it is not straightforward to interpret the particle swarm as following a landscape defined by the cost function. In PSO, the particles follow an intrinsic dynamics that does not even indirectly obtain any gradient information. They interact with each other by receiving an update of the position of the best solution currently known, if this position changes. The particles, after random initialisation, obey a linear dynamics of the following form (Kennedy and Eberhart 1995).

$$\begin{aligned} \mathbf{v}_{i,t+1} &= \omega \mathbf{v}_{i,t} + \alpha_1 \mathbf{R}_1 (\mathbf{p}_i - \mathbf{x}_{i,t}) + \alpha_2 \mathbf{R}_2 (\mathbf{g} - \mathbf{x}_{i,t}) \\ \mathbf{x}_{i,t+1} &= \mathbf{x}_{i,t} + \mathbf{v}_{i,t+1} \end{aligned} \quad (1)$$

Here $\mathbf{x}_{i,t}$ and $\mathbf{v}_{i,t}$, for $i = 1, \dots, N, t = 0, 1, 2, \dots$, represent, respectively, the d -dimensional position in the search space and the velocity vector of the i -th particle in a swarm of N particles at time t .

The velocity update contains an inertia term parameterised by ω and includes attractive terms that are analogous to forces towards the personal best location \mathbf{p}_i and towards the current best location among all particles \mathbf{g} , which are parameterised by α_1 and α_2 , respectively. The symbols \mathbf{R}_1 and \mathbf{R}_2 denote diagonal matrices whose nonzero entries are uniformly distributed in the unit interval. This implements a component-wise multiplication of the difference vectors with a random vector and is known to introduce a bias into the algorithm as particles show a tendency to stay near the axes of the problem space (Janson and Middendorf 2007; Spears et al. 2010).

In order to operate as an optimiser, the algorithm uses a cost function $F : \mathbb{R}^d \rightarrow \mathbb{R}$ that is bounded from below. Without loss of generality, we will assume that $F(\mathbf{x}^*) = 0$ at an optimal solution \mathbf{x}^* . The position of each particle is used to represent a solution of the optimisation problem that consists in the minimisation of the function F over a d -dimensional problem space that also forms the position space of the particles. The cost function is evaluated for the state of each particle at each time step. If $F(\mathbf{x}_{i,t})$ is better than $F(\mathbf{p}_i)$, then the personal best \mathbf{p}_i is replaced by $\mathbf{x}_{i,t}$. Similarly, if one of the particles arrives at a state with a cost less than $F(\mathbf{g})$, then \mathbf{g} is replaced in all particles by the position of the particle that has discovered the new best solution. If its velocity is nonzero, a particle will depart even from the current best location, but it still has a chance to return guided by the attractive terms in the dynamics (1).

Thus, in order for PSO to work effectively, the particle dynamics (1) is combined with a switching dynamics that is generated by the updates of \mathbf{g} or \mathbf{p}_i . In this way, the focus of the search dynamics is moved to a new location, in the neighbourhood of which better solutions can be expected. Whether this expectation is actually justified depends on the problem: the particles should rather look for better solutions in nearby places for some problems, whilst in other cases improvements are possible only when more distant regions of the search space are reached. Without prior knowledge about the size of the search space, it may seem reasonable not to restrict the particle dynamics from searching across all distances, i.e. to show a scale-free dynamics, but we will see that even trivial settings of the algorithm such as runtime limits may affect the optimal search strategy.

The particle dynamics depends on the parametrisation, i.e. on the values of ω , α_1 and α_2 , used in Eq. (1). To obtain the best result, we need to select parameter settings that achieve a balance between the particles exploiting the knowledge of good locations and exploring regions of the problem space further from the current particle positions. Although adaptive

schemes have been introduced with some success (Zhan et al. 2009; Hu et al. 2013; Erskine and Herrmann 2015a), parameter values are often experimentally determined, and poor selection may result in premature convergence of the swarm to poor local minima, or in a divergence of the particles towards regions that are irrelevant for the problem. Specifically, when we talk about convergence we mean that the particles collapse to a single point, or to a very small region, of the search space. Conversely, divergence is the situation in which particles in the swarm explode outwards from the initial area of focus, sampling points arbitrarily far from their starting location. Both convergence and divergence can be problematic, and effectively searching for an optimum is in large part about balancing these two opposing tendencies.

We propose a formulation of the PSO dynamics in terms of a random dynamical system (RDS) which leads to a description of the swarm dynamics. In this approach, we will not have to make simplifying assumptions on the stochasticity of the algorithm; instead, we present a stochastically exact formulation, which shows that the range of stable parameters is in fact larger than previously estimated. Our approach will also enable us to explain differences and similarities between theoretical and empirical results for PSO.

In the next section, we will consider an illustrative simulation of a particle swarm and move on to a standard matrix formulation of the swarm dynamics (1), see (Trelea 2003), in order to describe some of the existing analytical work on PSO. In Sect. 3, we will argue for a formulation of PSO as a random dynamical system (RDS) which will enable us to derive a novel exact characterisation of the dynamics of a one-particle system. In Sect. 4, we will compare the theoretical predictions with multi-particle simulations on a representative set of benchmark functions. In Sect. 5, we will discuss the assumptions we have made in order to obtain the analytical solution in Sect. 3 based on the empirical evidence for our approach.

2 Swarm dynamics

2.1 Empirical properties

We will now consider an empirical example to motivate our approach to parameter selection. We will consider only two parameters, namely inertia ω and a single parameter α governing the total strength of the attractive terms, i.e. $\alpha = \alpha_1 + \alpha_2$ and $\alpha_1 = \alpha_2$.

Choosing the parameters α_1 and α_2 different from each other does have an effect on the behaviour of PSO. In the present paper, we will not consider the general case of $\alpha_1 \neq \alpha_2$. For constant $\alpha = \alpha_1 + \alpha_2$, the effect of varying the relative weight of the two attractive terms does not seem to be big in most cases. It certainly deserves a more detailed study, but is beyond the scope of this paper.

Figure 1 shows how the parameters ω and α influence the performance of the PSO algorithm on the Rastrigin function, which is a typical benchmark problem (Liang et al. 2013). The (x, y) -plane represents the position in parameter space, and the vertical position shows the average value of the solution found by the algorithm. For positive ω values between zero and one, there appears to be a broadly banana curve-shaped portion of the parameter space that results in the best performances. For negative ω values, the best parameter pairs obey a nearly linear relationship. Parameter pairs that perform well occupy the regions in the valley of this surface plot.

Large (absolute) values of both α and ω are found to cause the particles to diverge leading to results far from optimality, whilst at small values for both parameters the particles tend to converge to a local optimum which sometimes is acceptable.

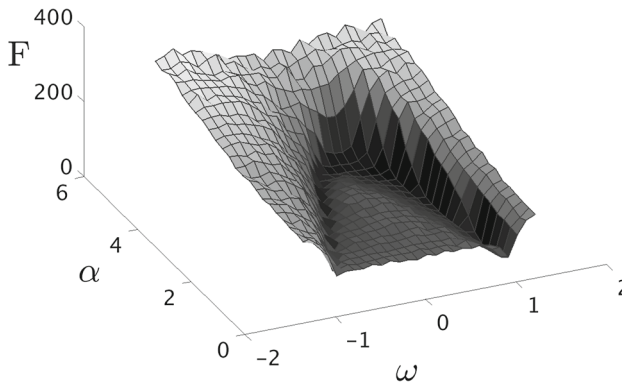


Fig. 1 Typical PSO performance as a function of the parameters ω and α . Here a 25 particle swarm was run for pairs of ω and α values ($\alpha_1 = \alpha_2 = \alpha/2$). The cost function here was the 10-dimensional non-continuous rotated Rastrigin function (Liang et al. 2013). Each parameter pair was repeated 100 times, and the minimal costs found in each run after 2000 iterations were averaged

For other cost functions (Liang et al. 2013), similar relationships are observed in numerical tests (see Sect. 4). It should be noted, however, that for simple cost functions, such as the *sphere* function (single well potential), parameter combinations with small ω and small α also usually lead to good results. With different objective functions, this picture may emerge sooner or later and not all locations in the valley are equally good. Likewise, the difference between the foot of the valley and locations up its slopes is not the same across all functions. However, the general pattern is manifest across many cost functions. We utilise cost functions used in the CEC2013 competition (Liang et al. 2013). All of these functions show a similar pattern, see Appendix A in (Erskine 2016).

Locating the best parameter pairs for specific problems often involves a degree of trial and error. We can explore PSO capabilities by repeatedly executing a simple variant of the algorithm for each parameter pair in the (α, ω) -space. With many repetitions and a suitably large number of fitness evaluations for each run, results can be averaged to even out variations in performance seen in individual executions.

Earlier analytic work (Trelea 2003; Chen et al. 2003; Zhan et al. 2011; Yue et al. 2012; Serani et al. 2015; Bonyadi and Michalewicz 2016, 2017) does not appear to suggest that a curved relationship similar to the valley in Fig. 1 should exist between parameters. In studies like (Martínez and Gonzalo 2008), the discrepancy between numerical simulations and theory becomes evident. Resolving this discrepancy is one of the goals of this paper. A preliminary explanation based on eigenvalues is provided by Jiang et al. (2007) and by Cleghorn and Engelbrecht (2015a,b). We follow a more general approach that is based on an explicit calculation of Lyapunov exponents, in order to obtain a more realistic and precise description, see the discussion below.

2.2 Matrix formulation

In order to analyse the behaviour of the PSO algorithm, it is convenient to use a matrix formulation by inserting the velocity explicitly in the second line of Eq. (1). If we consider particle states $\mathbf{z} = (\mathbf{v}, \mathbf{x})^T$ we can rearrange the update rules. In Eq. (2), we stack the updates for a single particle:

$$\begin{pmatrix} \mathbf{v}_{t+1} \\ \mathbf{x}_{t+1} \end{pmatrix} = \begin{pmatrix} \omega \mathbf{v}_t + \alpha_1 \mathbf{R}_1 (\mathbf{p} - \mathbf{x}_t) + \alpha_2 \mathbf{R}_2 (\mathbf{g} - \mathbf{x}_t) \\ \mathbf{x}_t + \omega \mathbf{v}_t + \alpha_1 \mathbf{R}_1 (\mathbf{p} - \mathbf{x}_t) + \alpha_2 \mathbf{R}_2 (\mathbf{g} - \mathbf{x}_t) \end{pmatrix} \quad (2)$$

In Eq. (3), we separate the state dependent parts from those elements that are independent of velocity or position:

$$\begin{pmatrix} \mathbf{v}_{t+1} \\ \mathbf{x}_{t+1} \end{pmatrix} = \begin{pmatrix} \omega \mathbf{I}_d & -\alpha_1 \mathbf{R}_1 - \alpha_2 \mathbf{R}_2 \\ \omega \mathbf{I}_d & \mathbf{I}_d - \alpha_1 \mathbf{R}_1 - \alpha_2 \mathbf{R}_2 \end{pmatrix} \begin{pmatrix} \mathbf{v}_t \\ \mathbf{x}_t \end{pmatrix} + \alpha_1 \mathbf{R}_1 \begin{pmatrix} \mathbf{p} \\ \mathbf{p} \end{pmatrix} + \alpha_2 \mathbf{R}_2 \begin{pmatrix} \mathbf{g} \\ \mathbf{g} \end{pmatrix} \quad (3)$$

In terms of our state variable $\mathbf{z} = (\mathbf{v}, \mathbf{x})^T$, this yields the following equation:

$$\mathbf{z}_{t+1} = \mathbf{M} \mathbf{z}_t + \alpha_1 \mathbf{R}_1 (\mathbf{p}, \mathbf{p})^T + \alpha_2 \mathbf{R}_2 (\mathbf{g}, \mathbf{g})^T \quad (4)$$

with $\mathbf{z} = (\mathbf{v}, \mathbf{x})^T$ and

$$\mathbf{M} = \begin{pmatrix} \omega \mathbf{I}_d & -\alpha_1 \mathbf{R}_1 - \alpha_2 \mathbf{R}_2 \\ \omega \mathbf{I}_d & \mathbf{I}_d - \alpha_1 \mathbf{R}_1 - \alpha_2 \mathbf{R}_2 \end{pmatrix} \quad (5)$$

where \mathbf{I}_d is the unit matrix in d dimensions. Note that the two occurrences of \mathbf{R}_1 in Eq. (5) refer to the same realisation of the random variable. Similarly, the two \mathbf{R}_2 's are the same realisation, but different from \mathbf{R}_1 . Since the second and third terms on the right in Eq. (4) are constant most of the time, the analysis of the algorithm can focus on the properties of the matrix \mathbf{M} . The approaches that we discuss in Sect. 2.3 are based on the analysis of \mathbf{M} . In Sect. 3, we propose a different approach: instead of using a deterministic variant of the algorithm or restricting the analysis to the expectation and variance values of the update matrix, we will analyse the long-term behaviour of the swarm considering the stationary probability distribution of the particles. The analysis will take place in the phase space that is composed of the position and velocity coordinates of the particle, i.e. the space of the \mathbf{z} vectors, see Eq. (4), which is subject to a stochastic dynamics that we will study in terms of the infinite product of the stochastic matrix \mathbf{M} .

2.3 Analytical results

An early exploration of the PSO dynamics by Kennedy (1998) considered a single particle in a one-dimensional problem space where the personal and global best locations were taken to be the same. The random components were replaced by their averages such that, apart from random initialisation, the algorithm was deterministic. Varying the parameters was shown to result in a range of periodic motions and divergent behaviour for the case of $\alpha_1 + \alpha_2 \geq 4$. The addition of the random vectors was seen as beneficial as it adds noise to the deterministic search.

Control of velocity, not requiring the enforcement of an arbitrary maximum value as in (Kennedy 1998), is derived in an analytical manner by Clerc and Kennedy (2002). Here, eigenvalues derived from the dynamic matrix of a simplified version of the PSO algorithm are used to imply various search behaviours. Thus, again the $\alpha_1 + \alpha_2 \geq 4$ case is expected to diverge. For $\alpha_1 + \alpha_2 < 4$, various cyclic and quasi-cyclic motions are shown to exist for a non-random version of the algorithm.

Also Trelea (2003) considered a single particle in a one-dimensional problem space, using a deterministic version of PSO, setting $\mathbf{R}_1 = \mathbf{R}_2 = 0.5$. The eigenvalues of the system were determined as functions of ω and a combined α , which leads to three conditions: the particle is shown to converge when $\omega < 1$, $\alpha > 0$ and $2\omega - \alpha + 2 > 0$. Harmonic oscillations occur for $\omega^2 + \alpha^2 - 2\omega\alpha - 2\omega - 2\alpha + 1 < 0$ and a zigzag motion for $\omega < 0$ and $\omega - \alpha + 1 < 0$. As with the preceding papers, the discussion of the random numbers in the algorithm views them purely as enhancing the search capabilities by adding a “drunken walk” to the particle

motions. Their replacement by expectation values was thus believed to simplify the analysis with no loss of generality.

A weakness in these early papers stems from the treatment of the stochastic elements. Rather than replacing the R_1 and R_2 vectors by 0.5, the dynamic behaviour can be explored by considering their expectation values and variances. An early work taking this approach produced a predicted best performance region in the parameter space similar to the curved valley of best values that is seen empirically (Jiang et al. 2007). The authors explicitly consider the convergence of means and variances of the stochastic update matrix. The curve they predict marks the locus of (ω, α) -pairs they believed guaranteed swarm convergence, i.e. parameter values within this curve result in convergence. We will refer to this curve as the Jiang curve (Jiang et al. 2007), although matching curves have also been found in other studies (Poli 2009) or have been derived from a weaker stagnation assumption (Liu 2014). An extensive recent review of such analyses is provided by Bonyadi and Michalewicz (2017).

We show in this paper that the random factors \mathbf{R}_1 and \mathbf{R}_2 in fact add a further level of complexity to the dynamics of the swarm which affects the behaviour of the algorithm in a non-trivial way. Essentially, it is necessary to consider both the stationary distribution of particles in the state space of the system and the properties of the infinite product of the stochastic update matrix. This leads to a locus of critical parameters that differs from previous analyses. This locus lies outside the Jiang curve (Jiang et al. 2007). We should note that the Jiang curve implies convergence for parameter pairs within the curve, but it does not cover the entire convergent region in the parameter space. Our analytical solution of the stability problem for the swarm dynamics explains why parameter settings derived from the deterministic approaches are not in line with what is observed in practical tests. For this purpose, we formulate the PSO algorithm as a random dynamical system and present an analytical solution for the swarm dynamics in a simplified, but representative case.

In our analysis, we start by considering the one-dimensional single-particle case and provide an analytical expression for the Lyapunov exponent, which we then solve numerically for a range of α, ω pairs, demonstrating that the critical parameters (i.e. those where the Lyapunov exponent is 0 and the swarm is expected to neither expand nor contract in the limit) lie in a banana-like curve on the plane. The hypothesis that PSO performs best when its behaviour is critical is then evaluated numerically, showing a good match between the predicted curve and the optimum parameters found experimentally. This is reassuring regarding the concern that the simplifications made in order to compute the Lyapunov exponent are overly restrictive and don't generalise to real PSO dynamics. In addition to the experimental evidence, we also consider the effects of relaxing various assumptions made to derive the analytic expression. In particular, we consider the effect of restricting analysis to one-dimensional systems, and we explore the case where a particle's personal best is not the global best, showing that this does not affect the convergence results, even though it can influence the short-term dynamics of the swarm. With this relaxation, our analysis is applicable to multi-particle swarms during the updates in which no improvements are made. Improvements tend to reduce in frequency over the runtime of the algorithm, and thus, our analysis fits the behaviour of the algorithm better as the run progresses. In other words, our analysis does not directly address the *switching dynamics*, i.e. the effect on the overall swarm dynamics of the personal or global best changing during the run. Although the various experimental results align well with the predictions, suggesting that this simplification does not significantly affect the results, we also explore the potential effects of the switching dynamics in the discussion.

3 PSO as a random dynamical system

3.1 Dynamics for a single particle

In this section, we study the dynamics of the particle swarm in the single-particle case as it was done by Kennedy (1998); Trelea (2003). This can be justified because the particles interact only via the global best position so that, while \mathbf{g} , see Eq. (1), is unchanged, single particles exhibit qualitatively the same dynamics as in the swarm. For the one-particle case, we necessarily have $\mathbf{p} = \mathbf{g}$ (see Sect. 4.3 for the general case $\mathbf{p} \neq \mathbf{g}$ which describes the independent dynamics between updates of the global and personal best of the individual particles in the multi-particle case) and shift invariance allows us to set both to zero, which leads us to the following stochastic map formulation of the PSO dynamics, see Eq. (4),

$$\mathbf{z}_{t+1} = \mathbf{M}\mathbf{z}_t \quad (6)$$

Extending earlier approaches, we explicitly consider the randomness of the dynamics, i.e. instead of averages over \mathbf{R}_1 and \mathbf{R}_2 , we consider a random dynamical system. Each iteration of the algorithm generates a new matrix of the form of Eq. (5) with freshly drawn random values for \mathbf{R}_1 and \mathbf{R}_2 . The evolution of the swarm is thus determined by the multiplicative effect of these matrices. To explore this, as above, we can replace the α_1 and α_2 with a new parameter $\alpha = \alpha_1 + \alpha_2$, with $\alpha_1, \alpha_2 \geq 0$ and $\alpha > 0$. We rewrite the update matrix (5) as a new dynamic matrix \mathbf{M} which is chosen from the set

$$\mathcal{M}_{\alpha, \omega} = \left\{ \begin{pmatrix} \omega \mathbf{I}_d & -\alpha \mathbf{R} \\ \omega \mathbf{I}_d & \mathbf{I}_d - \alpha \mathbf{R} \end{pmatrix}, \mathbf{R}_{ij} = 0 \text{ for } i \neq j \text{ and } \mathbf{R}_{ii} \in [0, 1] \right\} \quad (7)$$

Here \mathbf{R} in both rows is the same realisation of a random diagonal matrix that combines the effects of \mathbf{R}_1 and \mathbf{R}_2 . In order to determine \mathbf{R} , consider that the diagonal elements of \mathbf{R}_1 and \mathbf{R}_2 are uniformly distributed in $[0, 1]$. Thus, the diagonal elements of $\alpha_1 \mathbf{R}_1$ are uniformly distributed in $[0, \alpha_1]$, and the diagonal elements of $\alpha_2 \mathbf{R}_2$ are uniformly distributed in $[0, \alpha_2]$. The distribution of the sum of the random vectors $\mathbf{R}_{ii} = \frac{\alpha_1}{\alpha} \mathbf{R}_{1,ii} + \frac{\alpha_2}{\alpha} \mathbf{R}_{2,ii}$ is the convolution of the two probability distributions, namely

$$P_{\alpha_1, \alpha_2}(s) = \begin{cases} \frac{\alpha^2 s}{\alpha_1 \alpha_2} & \text{if } 0 \leq s \leq \min \left\{ \frac{\alpha_1}{\alpha}, \frac{\alpha_2}{\alpha} \right\} \\ \frac{\alpha}{\max\{\alpha_1, \alpha_2\}} & \text{if } \min \left\{ \frac{\alpha_1}{\alpha}, \frac{\alpha_2}{\alpha} \right\} < s \leq \max \left\{ \frac{\alpha_1}{\alpha}, \frac{\alpha_2}{\alpha} \right\} \\ \frac{\alpha^2 (1-s)}{\alpha_1 \alpha_2} & \text{if } \max \left\{ \frac{\alpha_1}{\alpha}, \frac{\alpha_2}{\alpha} \right\} < s \leq 1 \end{cases} \quad (8)$$

if the variable $s \in [0, 1]$ and $P_{\alpha_1, \alpha_2}(s) = 0$ otherwise, where $\alpha = \alpha_1 + \alpha_2$ and $\alpha_1, \alpha_2 \geq 0$. $P_{\alpha_1, \alpha_2}(s)$ has a tent shape for $\alpha_1 = \alpha_2$ and a box shape in the limits of either $\alpha_1 \rightarrow 0$ or $\alpha_2 \rightarrow 0$. Thus, the selection of particular values for the α_1 and α_2 parameters will determine the distribution for the random multiplier \mathbf{R}_{ii} in Eq. (7) and, thus, the distribution of the matrices in the set \mathcal{M} .

We expect that the multi-particle PSO is well represented by the simplified version for $\alpha_2 \gg \alpha_1$ or $\alpha_1 \gg \alpha_2$, the latter case being irrelevant in practice. For $\alpha_1 \approx \alpha_2$, deviations from the theory may occur because in the multi-particle case \mathbf{p} and \mathbf{g} will be different for most particles. We will discuss this as well as the effects of the switching of the dynamics at discovery of better solutions in Sect. 5.3.

3.2 Stochastic stability

We aim at describing the behaviour of the PSO swarm by considering whether it is convergent, divergent or stochastically stable. For this purpose, we will consider the maximal Lyapunov exponent λ of the system as a way to describe the stability properties of the dynamics. If one considers two particles initially positioned arbitrarily close to each other and subject to the dynamics of the PSO update rules, then we may observe whether the particles tend to move apart or move closer together as the algorithm runs. For two points that at time $t = 0$ are a distance δZ_0 apart, we denote the future separation of these points by δZ_t :

$$\delta Z_t = e^{\lambda t} \delta Z_0 \quad (9)$$

For swarms where particles are converging with probability 1, the largest Lyapunov exponent λ is less than zero. For divergent swarms, λ is greater than zero.

As the system state is updated in a linear manner, we can consider the effect of these updates on particles on a unit circle in the state space of the system, which is formed by a combination of the position and velocity coordinates. Each update moves the particles inwards or outwards depending on both where the particles are within the state space and the particular stochastic matrix drawn for the update. The iterated behaviour of the swarm is thus determined by the product of stochastic matrices drawn from the set $\mathcal{M}_{\alpha, \omega}$. We need to consider how such products behave on average. Such products have been studied for several decades (Furstenberg and Kesten 1960) and have found applications in physics, biology, and economics. Based on our discussion of criticality in Sect. 1, we aim at determining the α and ω pairs that neither cause the swarm to converge nor lead to an escape of the particles from the search domain, i.e. pairs that maintain a marginally stable dynamics which is characterised by a Lyapunov exponent $\lambda = 0$. The analysis below shows how to calculate the Lyapunov exponent for the simplified version of the system given in Eq. (1). Alternatively, one may use a numerical approach such as the resampled Monte Carlo method (Vanneste 2010).

Whilst none of the particles of a stable swarm discovers any new personal (or global) best solutions, its dynamical properties are determined by an infinite product of matrices from the set $\mathcal{M}_{\alpha, \omega}$ given by Eq. (7). This provides a convenient way to explicitly model the stochasticity of the swarm dynamics such that we can claim that the performance of PSO is determined by the stability properties of the random dynamical system described by Eq. (6).

Since Eq. (6) is linear, the analysis can be restricted to vectors on the unit sphere in the (\mathbf{v}, \mathbf{x}) space, i.e. to unit vectors

$$\mathbf{a} = (\mathbf{x}, \mathbf{v})^\top / \|(\mathbf{x}, \mathbf{v})^\top\| \quad (10)$$

where $\|\cdot\|$ denotes the Euclidean norm. In order to determine whether the swarm is on average expanding, contracting or stable, we assess at each iteration whether particles move from the unit circle: outward for divergence; inward for convergence. Unless the set of matrices shares the same eigenvectors (which is not the case here), standard stability analysis in terms of eigenvalues is not applicable. Instead, we will use tools from the theory of random matrix products in order to decide whether the set of matrices is stochastically contractive, i.e. whether the swarm does converge in the infinite limit given the distribution of matrices generated for given parameter values. Figure 2 visualises the process. A particle in the PSO swarm may be shown on a unit circle in a state space formed by the combination of position and velocity coordinates. At each iteration, the update rules given by Eq. (1) move the particle elsewhere in this space. If the particle moves outward from the unit circle, then it is contributing to expansion of the swarm (as shown in the figure). If it moves within the circle,

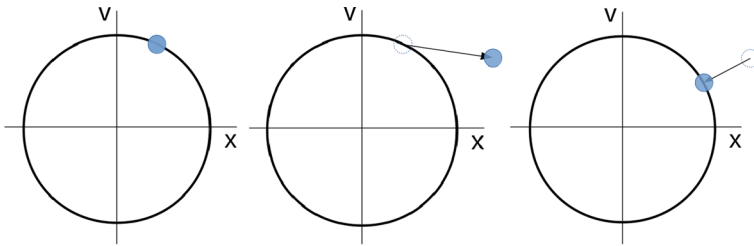


Fig. 2 PSO update and projection onto unit circle process. Here we visualise the process of a single particle in one dimension. (left) A single particle is situated on the unit circle in this state space. (centre) Applying PSO update rules results in the particle moving to a new state which is not on the unit circle. If most particles move out of the circle, the swarm is expanding; if inward, then it is contracting. (right) The particle (whether outward or inward moving) is re-projected back to the unit circle ready for the next update. The process thus results in the particle remaining a single unit away from the state space origin, but its position on the unit circle may change at each update

then it is contributing to the contraction of the swarm. To assess the behaviour of the whole swarm, we integrate over these individual moves. At the end of the iteration, we project the particle back to the unit circle. As the system update is linear, the scale of the system is not relevant. A matrix update that moves a particle 10% out of the circle or 50% in from the circle does so whatever the size of the circle. All updates are thus made relative to the unit circle for convenience.

The resultant effect of the update shown in Fig. 2 is to shift the position of the particle on the unit circle. We need to consider also the location of the particle in the state space when an update is applied. One way is to think of having multiple particles spread through the state space, i.e. positioned around the unit circle. These particles, that started elsewhere, may have reached other locations. However, the dynamics of this system essentially treats the particles individually. Only when a new personal best improves upon the swarm's global best, does one particle influence the others. In general, such improvements are rare, and as the algorithm runs they tend to occur less and less often. The stability of the system may thus be explored by only considering a single particle.

Any given update of the particle coordinates in the state space will result in some particles moving outward, or inward, from the unit circle. The likelihood of these moves is dependent on where on the unit circle a particle lies prior to the update being applied. In order to assess the whole effect, we therefore need to determine the distribution of particles on the unit circle under the dynamics of the update rules and the specific parameter values. Whilst updates are stochastic, some portions of the state space act somewhat like attractor regions, i.e. particles in these portions of the unit circle are likely to remain there and only rare matrix updates will appreciably shift them elsewhere. Figure 3 visualises this for a single iteration of a small swarm. The particles in the top right are moved outward during the shown update. The re-projection will return them to a location close to their previous place. The remaining particles move inward, by a larger amount. At this iteration, the sum of these moves may suggest the swarm is contracting a little. However, the particle in the bottom right will be re-projected to join the main group of particles. If this upper left location has an overall tendency to be expansive, then on subsequent iterations more and more particles may be contributing more divergent behaviour to the overall swarm.

We can estimate the stationary distribution of particles on the unit circle, $v_{\alpha,\omega}(\mathbf{a})$, by the following process. A number of particles, \mathcal{N}_p , are placed around our unit circle. For each particle, we create a set of new particle locations by applying the update Eq. (6) \mathcal{N}_u times. Each

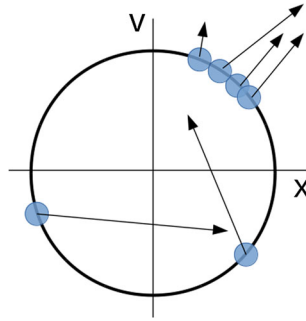


Fig. 3 Here we visualise a number of particles on the unit circle. To assess whether the whole swarm is expanding or contracting, we need to consider the effects of both the location on the unit circle on which each particle lies, and the dynamics of the PSO algorithm as applied to a particle at that location. Here the PSO update rules are moving the lower particles closer to the origin, so are contractile for those particles. However, the upper particles are moving outward representing a divergent move. Summing the sizes and directions of these moves gives an indication of the effect on the swarm of this iteration. However, when the moved particles are re-projected onto the unit circle, there will now be five particles in the top right area

update is re-projected onto the unit circle (as per Eq. (10)). This gives us $\mathcal{N}_{\text{new}} = \mathcal{N}_p \mathcal{N}_u$ new particles. We can then randomly sample \mathcal{N}_p of these and repeat the process. Different PSO parameter values result in different stochastic update matrices ($\mathcal{M}_{\alpha, \omega}$). These yield different stationary distributions of the particles. Figure 4 shows a number of these. Equation (12) expresses this process for a state space of dimension d and represents the definition of the stationary distribution.

The properties of the asymptotic dynamics can be described based on a double Lebesgue integral over the unit sphere S^{2d-1} and the set $\mathcal{M}_{\alpha, \omega}$ (Khas'minskii 1967). As in Lyapunov exponents, the effect of the dynamics is measured in logarithmic units in order to account for multiplicative action:

$$\lambda(\alpha, \omega) = \int d\nu_{\alpha, \omega}(\mathbf{a}) \int dP_{\alpha, \omega}(\mathbf{M}) \log \|\mathbf{M}\mathbf{a}\| \quad (11)$$

If $\lambda(\alpha, \omega)$ is negative, the algorithm will converge to \mathbf{p} with probability 1, while for positive λ arbitrarily large fluctuations are possible. While the measure for the inner integral of Eq. (11) is given by Eq. (8), we have to determine the stationary distribution $\nu_{\alpha, \omega}$ (called invariant measure by Khas'minskii (1967)) on the unit sphere for the outer integral. The stationary distribution $\nu_{\alpha, \omega}$ tells us where (or, rather, in which sector) the particles are likely to be in the state space, and is given by the solution of the integral equation:

$$\nu_{\alpha, \omega}(\mathbf{a}) = \int d\nu_{\alpha, \omega}(\mathbf{b}) \int dP_{\alpha, \omega}(\mathbf{M}) \delta(\mathbf{a}, \mathbf{M}\mathbf{b} / \|\mathbf{M}\mathbf{b}\|), \quad \mathbf{a}, \mathbf{b} \in S^{2d-1} \quad (12)$$

which represents the stationarity of $\nu_{\alpha, \omega}$, i.e. the fact that under the action of the matrices from $\mathcal{M}_{\alpha, \omega}$ the distribution of particles over sectors remains unchanged. For particles on the unit circle, we consider their moves expressed by the δ -function on a given update. This is integrated over the distribution of stochastic matrices $dP_{\alpha, \omega}(\mathbf{M})$.

Obviously, if the particles are more likely to reside in some region on the unit circle, then this region should have a stronger influence on the stability, see Eq. (11). The existence of the invariant measure requires the dynamics to be ergodic, which is ensured if at least some elements of $\mathcal{M}_{\alpha, \omega}$ have complex eigenvalues, which is the case for $\omega^2 + \alpha^2/4 - \omega\alpha - 2\omega - \alpha + 1 < 0$ (see above, (Trelea 2003)). This condition excludes a small region

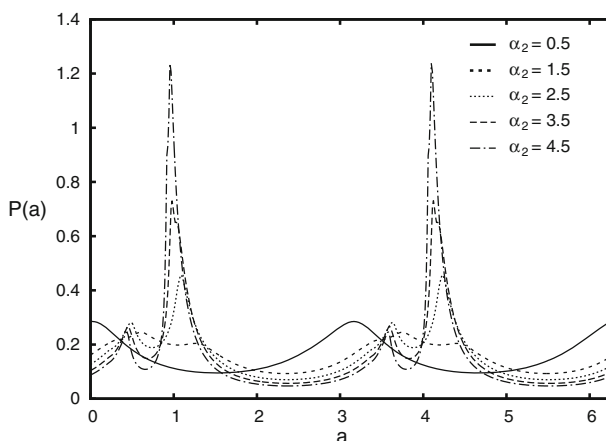


Fig. 4 Stationary distribution $v_{\alpha,\omega}(\mathbf{a})$ on the unit circle ($\mathbf{a} \in [0, 2\pi)$) in the (x, v) plane for a one-particle system of Eq. (6) for $\omega = 0.7$ and $\alpha = \alpha_2 = 0.5$ (solid), 1.5 (short dashed), 2.5 (dotted), 3.5 (wide dashed), 4.5 (dash dotted)

in the parameters space at small values of ω . If ergodicity is not guaranteed, it is possible that some distributions on the unit circle do not converge towards the stationary distribution by iterating Eq. (12). In the present case, small ω and large α cause cyclic (“zigzagging”) behaviour which prevents convergence if the initial distribution was not symmetric. We can easily prevent this by starting from a homogenous initial distribution which guarantees that the effect of the “zigzagging” is balanced and will thus not affect the stationary distribution. We should also remark that, although such problems are theoretically possible, we have not been able to reproduce them in the simulations.

The shape of the stationary distribution depends on the parameters α and ω and differs strongly from a homogenous distribution, see Fig. 4 for a few examples in the case $d = 1$. It can be noted that the strong inhomogeneity of the stationary distribution is in line with previous demonstrations of particles having a bias towards becoming aligned to the axes (Spears et al. 2010).

3.3 Critical swarm conditions

Critical parameters are obtained from Eq. (11) by the relation

$$\lambda(\alpha, \omega) = 0 \quad (13)$$

A similar goal was aimed at by Erskine and Herrmann (2015a), where, however, an adaptive scheme rather than an analytical approach was invoked in order to identify critical parameters. Solving Eq. (13) is difficult in higher dimensions, so we rely on the linearity of the system when considering the ($d = 1$)-case as representative. Figure 5 represents solutions of Eq. (13). Alternatively one may use a numerical approach such as the resampled Monte Carlo method (Vanneste 2010). The figure shows two curves that correspond to different values of the parameters α_1 and α_2 . Parameter pairs that lie inside a given curve give rise to swarms whose largest Lyapunov exponent is less than zero. These swarms are therefore stable in the sense that they will converge or become static. For parameter pairs outside each curve, the swarm generated will have a largest Lyapunov exponent greater than zero and will tend to explode.

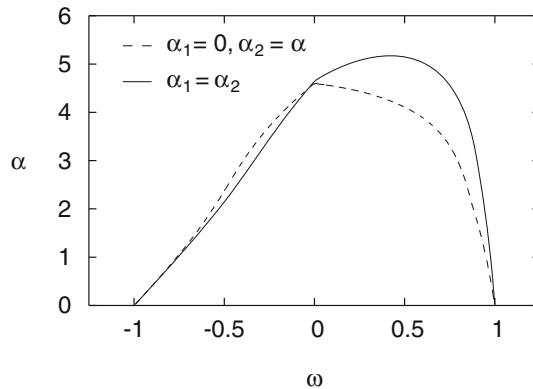


Fig. 5 Solution of Eq. (13) representing a single particle in one dimension with a fixed best value at $\mathbf{g} = \mathbf{p} = 0$. The solid curve is for $\alpha_1 = \alpha_2$, the dashed curve is for $\alpha = \alpha_2$, $\alpha_1 = 0$. Except for the regions near $\omega = \pm 1$, where numerical instabilities can occur in the integration, simulations produce an indistinguishable curve: in the simulation, we tracked the probability of a particle to either reach a small region (10^{-12}) near the origin or to escape beyond a radius of 10^{12} after starting from a random location on the unit circle. Along the curve, both probabilities are equal

Parameter pairs that yield swarms with Lyapunov exponents equal to zero are therefore stable in the infinite limit. However, they can make deviations into either divergent or convergent behaviours for extended periods of time. We should note that this arises from the infinite product of our stochastic matrices and is true for both one or many particles.

PSO experiments use finite iteration counts, so any individual trial may yield a set of random matrices whose product may generate a behaviour that for the limited nature of a single experiment differs from this theoretical approach. This means the theory developed here is expected to apply for generic cases. The set of parameter pairs on the curve result in a critical swarm, whose deviations are not described by either convergence or divergence.

The solid curve in Fig. 5 represents the solution for $d = 1$, $\alpha = \alpha_1 + \alpha_2$ and $\alpha_1 = \alpha_2$. The dashed curve is the solution for $d = 1$, $\alpha = \alpha_2$ and $\alpha_1 = 0$.

Inside the contour (Fig. 5), $\lambda(\alpha, \omega)$ is negative, meaning that the state will converge with probability 1. Along the contour and in the outside region, large state fluctuations are possible. Interesting parameter values are expected near the curve where, due to a coexistence of stable and unstable dynamics (induced by different sequences of random matrices), a theoretically optimal combination of exploration and exploitation is possible. For specific problems, however, deviations from the critical curve can be expected to be beneficial. In order to solve a practical problem, there is usually a finite number of fitness evaluations for the algorithm. In that case, it is beneficial to allow the swarm to converge at some point. Thus, the swarm being somewhat subcritical to allow such a convergence is desirable. The degree of subcriticality will depend on the number of fitness evaluations and may also be related to the nature of the problem space being explored.

Our result for $\alpha_1 = \alpha_2$ is also given as a black curve in Fig. 6 (top), where it is compared with the curve (dotted) of stability predicted by Jiang et al. (2007) and given as a polynomial by Cleghorn and Engelbrecht (2014). Experimentally, it is shown below that the outer curve, rather than the Jiang curve, represents the limit parameter pairs that lead to convergent swarms.

It is interesting that Clerc (2006b) presents a relationship between ω and α that is very similar to Fig. 5. It is also worth mentioning that Clerc's interpretation of the PSO dynamics in

terms of optimality near the *edge of chaos* is the same as the one supported here. Nevertheless, the curve in Fig. 4 of (Clerc 2006b) does not match the solution of Eq. (13), as can be seen at the value $\beta = \frac{1}{2}$ in Fig. 7 (note that Clerc's parameter c is scaled by a factor of $\frac{1}{2}$ relative to α). This difference is caused by the approximation of a Lyapunov exponent by the norm of the average dynamic matrix which is in general not exact unless the eigenvectors coincide. In addition, the result was fitted using the mean values of the dynamical coefficients and includes another approximation. The clear advantage of Clerc's approach is that explicit values can be obtained for the parameters in some cases, while the analytical result of Eq. (13) only permits a numerical solution. As the simulation results by Clerc (2006b) are not conclusive, it is difficult to distinguish the applicability of our solution from Clerc's approach to practical problems.

4 Optimisation of benchmark functions

4.1 Experimental setup

Metaheuristic algorithms are often tested in competitions against benchmark functions designed to represent problems with different characteristics. The 28 functions found in (Liang et al. 2013), for example, contain a mix of unimodal, basic multimodal and composite functions. The domains of the functions in this test set are all defined to be $[-100, 100]^d$ where d is the dimensionality of the problem. Particles are initialised uniformly randomly within the domain of the functions. We use 10-dimensional problems throughout. It may be interesting to consider higher dimensionalities, but $d = 10$ seems sufficient in the sense that it is very unlikely that a very good solution is found already at initialisation. Our implementation of PSO performs no spatial or velocity clamping. In all trials, a swarm of 25 particles is used. For the competition, 50,000 fitness evaluations were allowed which corresponds to 2000 iterations with 25 particles. In some cases, we consider also other iteration numbers (20, 200, 20,000) for comparison. Results are averaged over 100 trials. This protocol is carried out for pairs of $\omega \in [-1.1, 1.1]$ and $\alpha \in [0, 6]$. This experimental procedure is repeated for all 28 functions. The averaged solution cost as a function of the two parameters shows curved valleys similar to that in Fig. 1 for all problems. For each function, we obtain different best values along (or near) the theoretical curve given by Eq. (13). There appears to be no generally preferable location within the valley.

4.2 Empirical results

All parameter pairs are evaluated using the average over the performance on all benchmark functions (Liang et al. 2013). The 5% best parameter pairs are shown in Fig. 6 for different numbers of fitness evaluations. For more fitness evaluations, the best locations move out from the origin as we would expect. For 2000 iterations per run, the best performing locations appear to agree well with the Jiang curve (Jiang et al. 2007). It is known that some problem functions return good results even when parameters are well inside the stable line. Simple functions (e.g. *sphere*) benefit from early swarm convergence. Thus, our average performance may mask the full effects. Figure 6 also shows an example of the best performing parameter for 2000 iterations on a single function. The *sphere* function shows many locations beyond the Jiang curve for which good results are obtained.

In Fig. 8, detailed explorations of two functions are shown. For these, we set $\omega = 0.55$, while α is varied with a much finer granularity between 2 and 6. In total, 2000 repetitions

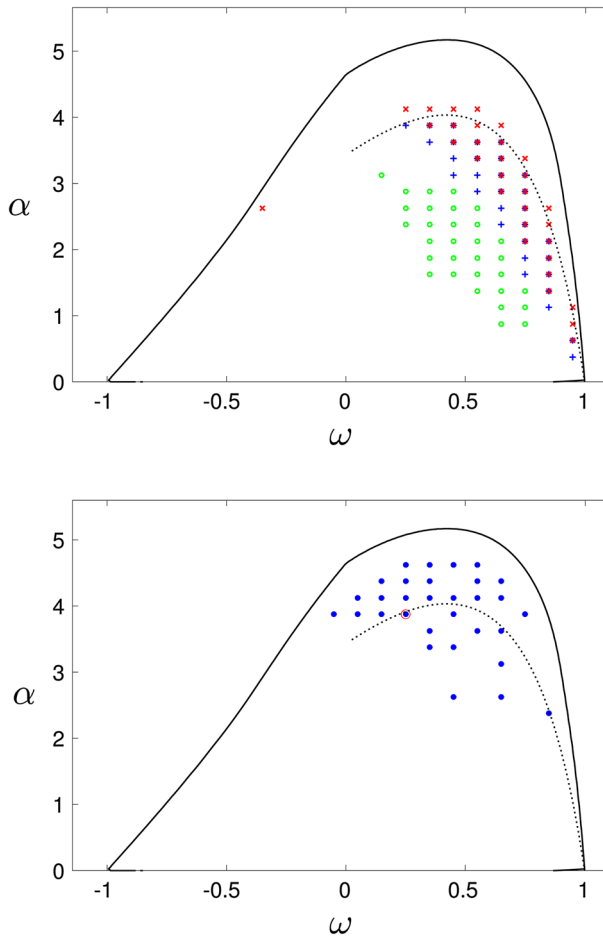


Fig. 6 Empirical cost function value results. (top) 5% best parameter pair locations across all 28 functions (Liang et al. 2013) plotted against our curve (solid) and the Jiang curve (dotted). The numbers of iterations are 20 (circles), 200 (crosses), 2000 (x's). (bottom) Dots represent the 5% best parameter pair locations for function 21 plotted against our curve (solid) and the Jiang curve (dotted). The encircled dot indicates the best location over all parameter pairs

of the algorithm are performed for each parameter pair. The curves shown are for increasing number of iterations (20, 200, 2000, 20,000). Vertical lines mark where the two predicted stable loci sit on these parameter space slices.

The best results lie outside the Jiang curve for these functions. Our predicted stable limit appears to be consistent with these results. In other words, if the solution is to be found in a short time, a more stable dynamics is preferable, because the particles can settle in a nearby optimum at smaller fluctuations. If more time is available, then parameter pairs more close to the critical curve lead to an increased search range which obviously allows the swarm to explore better solutions. Similarly, we expect that in larger search spaces (e.g. relative to the width of the initial distribution of the particles) parameters near the critical line will lead to better results.

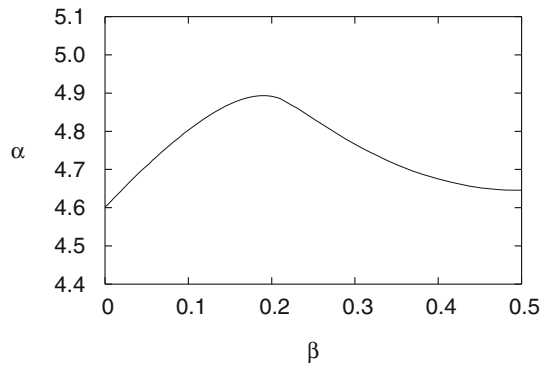


Fig. 7 For a general choice of α_1 and α_2 , the critical curve lies mostly between the two curves ($\alpha_1 = 0$, $\alpha_2 = \alpha$ and $\alpha_1 = \alpha_2 = \alpha/2$) shown in Fig. 5. This is, however, not always the case. Here the critical value of α is shown for $\alpha_1 = \beta\alpha$, $\alpha_2 = (1 - \beta)\alpha$ at an inertia value $\omega = 0$, which shows that intermediate splits between the personal and global best tend to be slightly more stable than the boundary cases ($\beta = 0$ and $\beta = 0.5$). Here the idealised case without a fitness function is considered such that the curve continues symmetrically beyond $\beta = 0.5$. In realistic cases, the weight of the personal best may be more important in the exploratory phase, while the weight of global best is crucial for the final phase of the optimisation

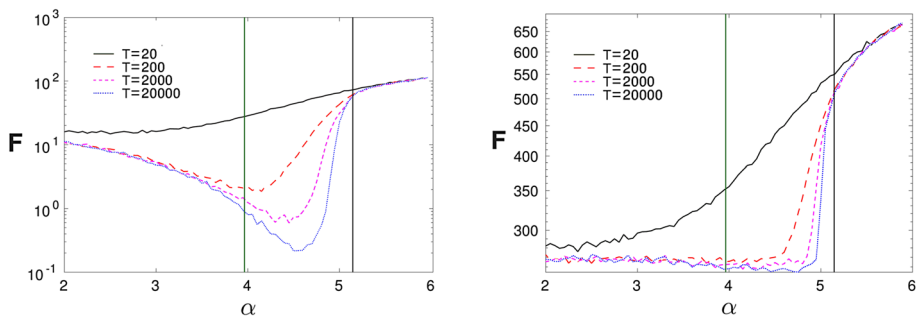


Fig. 8 Detailed empirical cost function value results. Detailed PSO performance along the ($\omega = 0.55$)-slice of parameter space for functions 7 (left) and 21 (right) (Liang et al. 2013). Different numbers of iterations are shown: 20 (solid), 200 (wide dashed), 2000 (short dashed) and 20,000 (dotted). Vertical lines show our limit (vertical line near $\alpha = 5$) and the Jiang limit (Jiang et al. 2007) (vertical line near $\alpha = 4$)

4.3 Personal best versus global best

A numerical scan of the (α_1, α_2) -plane shows a valley of good fitness values, which, for a small fixed positive ω , is roughly linear and described by the relation $\alpha_1 + \alpha_2 = \text{const}$; i.e. only the joint parameter $\alpha = \alpha_1 + \alpha_2$ matters. For large ω , and accordingly small predicted optimal α values, the valley is less straight. This may be because the effect of the known solutions is relatively weak, so the interaction of the two components becomes more important. In other words, if the movement of the particles is mainly due to inertia, then the relation between the global and local best is non-trivial, while at low inertia the particles can adjust their \mathbf{p} vectors quickly towards the \mathbf{g} vector so that both terms become interchangeable.

Due to linearity, the particle swarm update rule of Eq. (1) is subject to a scaling invariance that was also used in Eq. (10). We now consider the consequences of linearity for the case where personal best and global best differ, i.e. $\mathbf{p} \neq \mathbf{g}$. For an interval where \mathbf{p}_i and \mathbf{g} remain unchanged, the particle i with personal best \mathbf{p}_i will behave like a particle in a swarm where

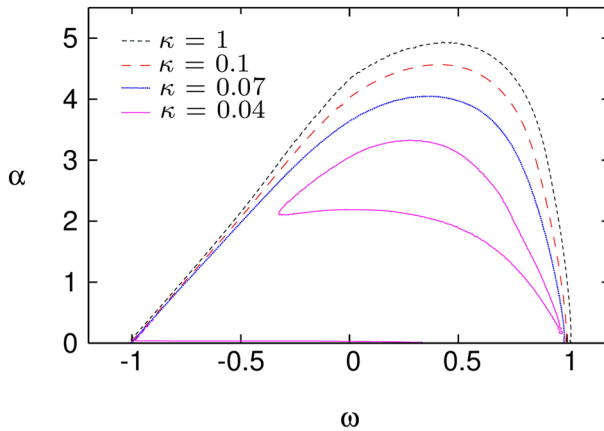


Fig. 9 For $\mathbf{p} \neq \mathbf{g}$, we define neutral stability as the equilibrium between divergence and convergence. Convergence means here that the particle approaches the line connecting \mathbf{p} and \mathbf{g} . Curves are for a one-dimensional problem with $\mathbf{p} = 0.1$ and $\mathbf{g} = 0$ scaled (see Sect. 4.3) by $\kappa = 1$ (short dashed), $\kappa = 0.1$ (wide dashed), $\kappa = 0.07$ (dotted) and $\kappa = 0.04$ (solid). Results are for 200 iterations and averaged over 100,000 repetitions

together with \mathbf{x} and \mathbf{v} , \mathbf{p}_i is also scaled by a factor $\kappa > 0$. The finite-time approximation of the Lyapunov exponent, see Eq. (11),

$$\lambda(t) = \frac{1}{t} \log \langle \|\mathbf{x}_t, \mathbf{v}_t\| \rangle \quad (14)$$

will be changed by an amount of $\frac{1}{t} \log \kappa$ by the scaling. Although this has no effect on the asymptotic behaviour, we will have to expect an effect on the stability of the swarm for finite times which may be relevant for practical applications. For the same parameters, the swarm will be more stable if $\kappa < 1$ and less stable for $\kappa > 1$, provided that the initial conditions are scaled in the same way. Likewise, if $\|\mathbf{p}_i - \mathbf{g}\|$ is increased, then the critical contour will move inwards, see Fig. 9, which is also confirmed by simulations, see Fig. 10. Note that in this figure, the low number of iterations leads to a few erroneous trials for parameter pairs outside the outer contour which have been omitted here. We also do not consider the behaviour near $\alpha = 0$ which is complex, but irrelevant for PSO. The contour that is obtained from Eq. (13) can be seen as the limit $\kappa \rightarrow 0$ so that only an increase of $\|\mathbf{p}_i - \mathbf{g}\|$ is relevant for comparison with the theoretical stability result. When comparing the stability results with numerical simulations for real optimisation problems, we will need to take into account the effects caused by differences between \mathbf{p} and \mathbf{g} in a multi-particle swarm with finite runtimes.

5 Discussion

5.1 Relevance of criticality

Our analytical approach predicts α and ω values that maintain the critical behaviour of the PSO swarm. Together with the *omega*, these values form a closed contour that describes the stability properties of the swarm: outside this contour, the swarm will diverge unless steps are taken to constrain it. Inside, the swarm will eventually converge to a single solution. In

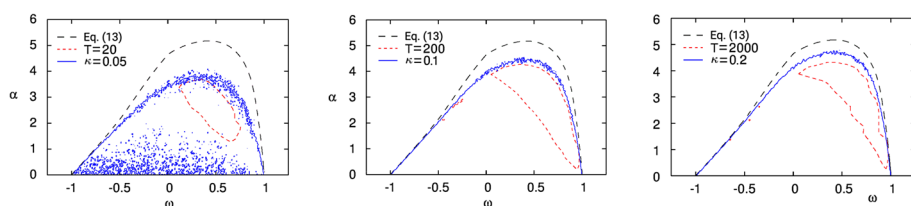


Fig. 10 Average over 28 benchmark functions after 20 (left, $|\mathbf{p} - \mathbf{g}| \approx 2$), 200 (centre, $|\mathbf{p} - \mathbf{g}| \approx 1$) and 2000 (right, $|\mathbf{p} - \mathbf{g}| \approx 0.5$) iterations. The short-dashed curve indicates the region where the top-10% of the best solutions are found. The wide-dashed curve shows asymptotic stability boundary. The solid curve shows a numerical solution of the integral for corresponding scales $\kappa = 0.05, 0.1$ and 0.2 , cf. Eq. (14) and Fig. 9

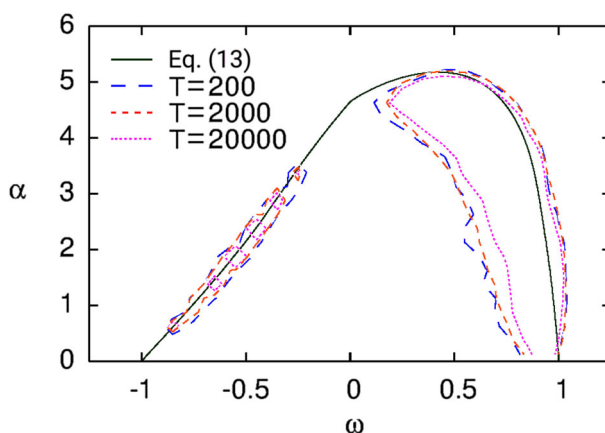


Fig. 11 Distribution of particle fluctuations. Region of lowest log mean square deviation from power law. Shown are results for two particles averaged over 28 benchmark functions in two dimensions after 200 (wide dashed), 2000 (short dashed), and 20,000 (dotted) iterations with, respectively, 1000, 100, and 10 repetitions. Due to the scale-free dynamics in the critical regime, the results for different iteration numbers are not significantly different. Interestingly, the dynamics fails to show signs of critical fluctuations for $\omega \approx 0$, although the quality of the solutions is similar along the critical line for $\alpha \gg 0$, which is shown as a solid line

order to locate a solution precisely within the search space, the swarm needs to converge at some point, so the line represents an upper bound on the exploration-exploitation mix that a swarm manifests. For parameters on the critical line, fluctuations are still arbitrarily large. Therefore, subcritical parameter values can be preferable so that the settling time is of the same order as the scheduled runtime of the algorithm. If, in addition, a typical length scale of the problem is known, then the finite standard deviation of the particle fluctuations in the stable parameter region can be used to decide about the distance of the parameter values from the critical curve. These dynamical quantities can be approximately set, based on the theory presented here, such that a precise control of the behaviour of the algorithm is in principle possible.

The observation of the distribution of empirically optimal parameter values along the critical curve confirms the expectation that critical or near-critical behaviour is the main reason for success of the algorithm. Critical dynamics (see Fig. 11) is a plausible tool in optimisation problems if, apart from certain smoothness assumptions, nothing is known about the cost landscape. The majority of the critical fluctuations will exploit the smoothness

of the cost function by local search, whereas the fat tails of the jump distribution allow the particles to escape from local minima.

5.2 Comparison with existing theory

The critical line in the PSO parameter space has been previously investigated and approximated by various authors (Poli 2009; Kadirkamanathan et al. 2006; Gazi 2012; Cleghorn and Engelbrecht 2014; Poli and Broomhead 2007). Many of these approximations are compared using empirical simulation in Gazi (2012). As Cleghorn and Engelbrecht (2014b) note, the most accurate calculation of the critical line so far is provided by Poli and Broomhead (2007) and by Poli (2009). In contrast, the method we present here uses a convergent approximation approach which does not exclude the effects of higher-order terms. Thus, where our results differ from those previously published (which occurs most for values of ω near zero), we can conclude that the difference is a result of incorporating the effects of these higher-order terms. Further, these higher-order terms do not have noticeable effect for ω values close to ± 1 , and thus, in these regions of the parameter space the two methods coincide.

The critical line we present defines the best parameters for a PSO allowed to run for infinite iterations. As the number of iterations (and the size of the problem space) decrease, the best parameters move inwards, and for around 2000 iterations the line proposed by Poli and Broomhead (2007) and by Poli (2009) provides a good estimate of the outer limit of good parameters. A potential explanation of the good match with the Poli line at lower iteration numbers and a poor match at large iteration numbers is that the small error introduced by ignoring higher-order terms accumulates over time.

The above-mentioned Jiang curve (Jiang et al. 2007) is an explanation in terms of eigenvalues which we are generalising here, i.e. the work presented here can be seen as a Lyapunov condition-based approach to uncovering the phase boundary. Previous work considering the Lyapunov condition has produced rather conservative estimates for the stability region (Gazi 2012; Kadirkamanathan et al. 2006) which is a result of the particular approximation used, while we avoid this by directly calculating the integral in Eq. (11) for the one-particle case.

5.3 Switching dynamics

Equation (4) shows that the discovery of a better solution affects only the constant terms of the linear dynamics of a particle, whereas its dynamical properties are governed by the (linear) parameter matrices. However, in the time step after a particle has found a new solution, the corresponding attractive term in the dynamics is zero, see Eq. (1), so that the particle dynamics slows down compared to the theoretical solution which assumes a finite distance from the best position at all (finite) times. As this affects usually only one particle at a time and because new discoveries tend to become rarer over time, this effect will be small in the asymptotic dynamics, although it could justify the empirical optimality of parameters in the unstable region for some test cases.

The stability of PSO cast as a random dynamical system is determined by the infinite product of its stochastic update matrix. Equation (4) shows that both a particle's personal best, \mathbf{p}_i , and the swarm's global best locations, \mathbf{g} , have a role in the stability of the swarm. When not changing, these terms provide additive components to the iterated updates. In order to achieve stability, the particles must counteract this influence by behaving somewhat subcritically, i.e. the ω and α parameters need to be within the derived critical line. However, as the swarm evolves, new finds become rarer and each \mathbf{p}_i will tend to converge towards \mathbf{g} . Thus, asymptotically, the dynamics will tend towards the theoretical case.

The question is, nevertheless, how often these changes occur. A weakly converging swarm can still produce good results if it often discovers better solutions by means of the fluctuations it performs before settling into the current best position. For cost functions that are not ‘deceptive’, i.e. where local optima tend to be near better optima, parameter values far inside the critical contour (see Fig. 5) may give good results, while in other cases more exploration is needed.

6 Conclusion

Particle swarm optimisation is a widely used optimisation metaheuristic. In previous approaches, inherent stochasticity of PSO was handled via simplifications such as the consideration of expectation values or independence assumptions, thus excluding higher-order terms that were, however, shown to be important in the approach presented in this paper. Thus, where our results differ from those previously published, we can conclude that the difference is a result of incorporating the effects of these higher-order terms. It is known that the standard PSO algorithm requires parameter tuning to ensure good performance. However, choosing optimal parameter values for any given problem can be difficult. It is shown here that the system can be modelled as a random dynamical system. Analysis of this system shows that there exists a locus of (ω, α) -pairs that result in the swarm behaving in a critical manner. This plays a role also in other applications of swarm dynamics, for example, the behaviour reported by [Erskine and Herrmann \(2015\)](#) occurred as well in the vicinity of critical parameter settings. Similarly, [Martius and Herrmann \(2010, 2012\)](#) showed that the (self-organised) criticality of the parameter dynamics makes it possible to achieve certain behaviours in a natural way in autonomous robots.

A weakness of the approach presented in this paper is that it addresses only the main parameters, ω and α , while swarm size or parameters regulating confinement of the swarm are not considered, although they are known to have an effect, see, for example, ([Clerc 2012](#)). In addition, we have focused only on the standard PSO ([Kennedy and Eberhart 1995](#)) which is known to include biases ([Clerc 2006a](#); [Spears et al. 2010](#)), that are not necessarily justifiable, and to be outperformed on benchmark sets as well as in practical applications by many of the existing PSO variants. Similar analyses are certainly possible and can be expected to be carried out for some of these variants or even for other metaheuristic algorithms.

Acknowledgements This work was supported by the Engineering and Physical Sciences Research Council (EPSRC), Grant Number EP/K503034/1. The authors are very grateful for the detailed comments from the referees and the constructive support from the editors.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Bonyadi, M. R., & Michalewicz, Z. (2016). Stability analysis of the particle swarm optimization without stagnation assumption. *IEEE Transactions on Evolutionary Computation*, 20(5), 814–819.
- Bonyadi, M. R., & Michalewicz, Z. (2017). Particle swarm optimization for single objective continuous space problems: A review. *Evolutionary Computation*, 25(1), 1–54.

- Chen, J., Pan, F., Cai, T., & Tu, X. (2003). Stability analysis of particle swarm optimization without Lipschitz constraint. *Journal of Control Theory and Applications*, 1(1), 86–90.
- Cleghorn, C. W., & Engelbrecht, A. P. (2014a). A generalized theoretical deterministic particle swarm model. *Swarm Intelligence*, 8(1), 35–59.
- Cleghorn, C. W. & Engelbrecht, A. P. (2014b). Particle swarm convergence: An empirical investigation. In *Proceedings of IEEE congress on evolutionary computation* (pp. 2524–2530). IEEE.
- Cleghorn, C. W., & Engelbrecht, A. (2015a). Fully informed particle swarm optimizer: Convergence analysis. In *Proceedings of 2015 IEEE congress on evolutionary computation (CEC)* (pp. 164–170). IEEE.
- Cleghorn, C. W., & Engelbrecht, A. P. (2015b). Particle swarm variants: Standardized convergence analysis. *Swarm Intelligence*, 9(2–3), 177–203.
- Clerc, M. (2006a). *Confinements and biases in particle swarm optimisation*. Technical Report hal-00122799, Open archive HAL. <http://hal.archives-ouvertes.fr/>.
- Clerc, M. (2006b). *Stagnation analysis in particle swarm optimization or what happens when nothing happens*. Technical Report hal-00122031, Open archive HAL. <http://clerc.maurice.free.fr/ps0>.
- Clerc, M. (2012). *Standard particle swarm optimisation*. Technical Report hal-00764996, Open archive HAL. <http://hal.archives-ouvertes.fr/>.
- Clerc, M., & Kennedy, J. (2002). The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1), 58–73.
- Erskine, A. (2016). *Analysis of behaviours in swarm systems*. Ph.D. thesis, University of Edinburgh, UK. <https://www.era.lib.ed.ac.uk/handle/1842/15897>.
- Erskine, A., & Herrmann, J. M. (2015a). CriPS: Critical particle swarm optimisation. In P. Andrews, L. Caves, R. Doursat, S. Hickinbotham, F. Polack, S. Stepney, T. Taylor, & J. Timmis (Eds.), *Proceedings of the European conference on artificial life* (pp. 207–214). Cambridge: MIT Press.
- Erskine, A., & Herrmann, J. M. (2015b). Cell-division behavior in a heterogeneous swarm environment. *Artificial Life*, 21(4), 481–500.
- Furstenberg, H., & Kesten, H. (1960). Products of random matrices. *The Annals of Mathematical Statistics*, 31(2), 457–469.
- Gazi, V. (2012). Stochastic stability analysis of the particle dynamics in the PSO algorithm. In *Proceedings of IEEE international symposium on intelligent control* (pp. 708–713). IEEE.
- Hu, M., Wu, T.-F., & Weir, J. D. (2013). An adaptive particle swarm optimization with multiple adaptive methods. *IEEE Transactions on Evolutionary Computation*, 17(5), 705–720.
- Janson, S. & Middendorf, M. (2007). On trajectories of particles in PSO. In *Proceedings of swarm intelligence symposium (SIS)* (pp. 150–155). IEEE.
- Jiang, M. J., Luo, Y., & Yang, S. (2007). Stagnation analysis in particle swarm optimization. In *Proceedings of swarm intelligence symposium (SIS)* (pp. 92–99). IEEE.
- Kadirkamanathan, V., Selvarajah, K., & Fleming, P. J. (2006). Stability analysis of the particle dynamics in particle swarm optimizer. *IEEE Transactions on Evolutionary Computation*, 10(3), 245–255.
- Kennedy, J. (1998). The behavior of particles. In V. Porto, N. Saravanan, D. Waagen, & A. E. Eiben (Eds.), *Evolutionary programming VII* (pp. 579–589). Berlin: Springer.
- Kennedy, J. & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of IEEE international conference on neural networks* (Vol. 4, pp. 1942–1948). IEEE.
- Khas'minskii, R. Z. (1967). Necessary and sufficient conditions for the asymptotic stability of linear stochastic systems. *Theory of Probability & Its Applications*, 12(1), 144–147.
- Liang, J. J., Qu, B. Y., Suganthan, P. N., & Hernández-Díaz, A. G. (2013). *Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization*. Technical Report 201212, Computational Intelligence Laboratory, Zhengzhou University, China and Nanyang Technological University, Singapore.
- Liu, Q. (2014). Order-2 stability analysis of particle swarm optimization. *Evolutionary Computation*, 23(2), 187–216.
- Martínez, J. L. F., & Gonzalo, E. G. (2008). The generalized PSO: A new door to PSO evolution. *Journal of Artificial Evolution and Applications*, 861275(15), 2008.
- Martius, G. & Herrmann, J. M. (2010). Taming the beast: Guided self-organization of behavior in autonomous robots. In *Proceedings of international conference on simulation of adaptive behavior* (pp. 50–61). Springer.
- Martius, G., & Herrmann, J. M. (2012). Variants of guided self-organization for robot control. *Theory in Biosciences*, 131(3), 129–137.
- 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). ACM.
- Serani, A., Diez, M., Campana, E. F., Fasano, G., Peri, D., & Iemma, U. (2015). Globally convergent hybridization of particle swarm optimization using line search-based derivative-free techniques. In *Recent advances in swarm intelligence and evolutionary computation* (pp. 25–47). Springer.
- Spears, W. M., Green, D. T., & Spears, D. F. (2010). Biases in particle swarm optimization. *International Journal of Swarm Intelligence Research*, 1(2), 34–57.
- Trelea, I. C. (2003). The particle swarm optimization algorithm: Convergence analysis and parameter selection. *Information Processing Letters*, 85(6), 317–325.
- Vanneste, J. (2010). Estimating generalized Lyapunov exponents for products of random matrices. *Physical Review E*, 81(3), 036701.
- Yue, B., Liu, H., & Abraham, A. (2012). Dynamic trajectory and convergence analysis of swarm algorithm. *Computing and Informatics*, 31(2), 371–392.
- Zhan, Z.-H., Zhang, J., Li, Y., & Chung, H. S.-H. (2009). Adaptive particle swarm optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 39(6), 1362–1381.
- Zhang, W., Jin, Y., Li, X., & Zhang, X. (2011). A simple way for parameter selection of standard particle swarm optimization. In *Proceedings of international conference on artificial intelligence and computational intelligence* (pp. 436–443). Springer.