

Comparison of linear and classical velocity update rules in particle swarm optimization: Notes on scale and frame invariance

Daniel N. Wilke¹, Schalk Kok^{1,*†} and Albert A. Groenwold²

¹*Structural Optimisation Research Group, Department of Mechanical Engineering, University of Pretoria, Pretoria 0002, South Africa*

²*Department of Mechanical Engineering, University of Stellenbosch, Private Bag X1, Matieland 7602, South Africa*

SUMMARY

In this paper we investigate whether the particle swarm optimization (PSO) algorithm is invariant of the scale and frame (i.e. translation and rotation) in which an objective function is posed. To do so, we study the linear and classical velocity update rules. We will show that the *linear* velocity update rule is scale and frame *invariant*, but that the *classical* velocity update rule lacks *rotational invariance*.

It is known that the linear velocity update rule lacks diversity, resulting in particle trajectories that collapse to line searches. In contrast, the classical velocity update rule maintains diverse (space filling) particle trajectories.

To illustrate that diversity and invariance are not necessarily exclusive, we propose a new velocity update rule. This update rule, which is just one example of many possible formulations, is rotationally invariant and at the same time directionally diverse. This is achieved through consistent perturbation of the search directions.

We quantify the (in)variance and performance of the three different implementations using a popular test set. The test problems are evaluated in both unrotated and arbitrarily rotated reference frames.
Copyright © 2006 John Wiley & Sons, Ltd.

Received 1 November 2005; Revised 28 June 2006; Accepted 13 July 2006

KEY WORDS: particle swarm optimization; scale; translation; rotation; reference frame; invariance; line search; trajectory collapse

1. INTRODUCTION

In a previous paper [1], the authors studied the directional diversity of the linear [2] and the classical [3, 4] velocity update rules. It was shown that, for the linear velocity update rule, particle trajectories

*Correspondence to: Schalk Kok, Structural Optimisation Research Group, Department of Mechanical and Aeronautical Engineering, University of Pretoria, Pretoria 0002, South Africa.

†E-mail: schalk.kok@up.ac.za

Contract/grant sponsor: National Research Foundation of South Africa; contract/grant number: 2059962

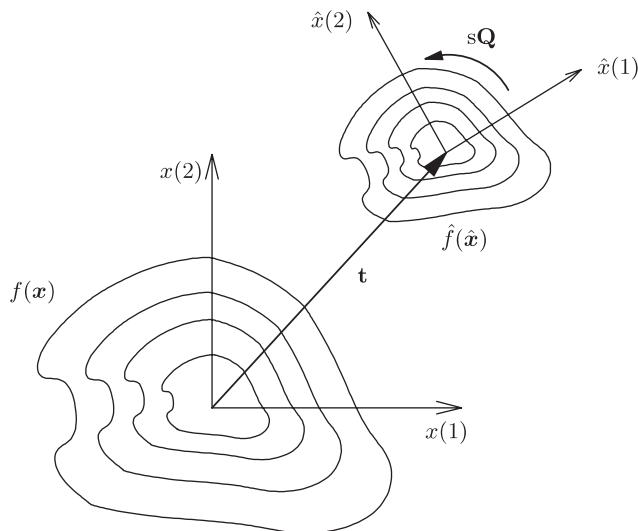


Figure 1. Function value contour plot. Two possible interpretations are that f and \hat{f} are distinct functions in the same reference frame, or that f is a single function expressed in two reference frames \mathbf{x} and $\hat{\mathbf{x}}$. s , \mathbf{t} and \mathbf{Q} denote scaling, translation and rotation, respectively.

collapse to ineffective line searches, while in the classical velocity update rule, directionally diverse search trajectories are maintained. In addition, the benefit of ensuring that directional diversity is present throughout an optimization run was demonstrated.

In this study, we now investigate whether the two velocity update rules under consideration are dependent of the scale and frame in which an optimization problem is defined. As an example, Figure 1 depicts two reference frames \mathbf{x} and $\hat{\mathbf{x}}$, related through a scale factor s , translation by a vector \mathbf{t} , and rotation by a proper orthogonal matrix \mathbf{Q} , i.e. $\hat{\mathbf{x}} = \mathbf{t} + s\mathbf{Q}\mathbf{x}$. The same function is expressed in each of these reference frames as $\hat{f}(\hat{\mathbf{x}}) = \hat{f}(\mathbf{t} + s\mathbf{Q}\mathbf{x})$. An alternative but equivalent interpretation of Figure 1 is that f and \hat{f} are two distinct functions described in the same reference frame \mathbf{x} , i.e. $f(\mathbf{x}) = \hat{f}(\mathbf{t} + s\mathbf{Q}\mathbf{x})$.

Let us now consider ‘invariance’ in the context of optimization algorithms. ‘Scale invariance’ implies algorithm performance that is independent of uniform scaling of all variables. ‘Frame invariance’ on the other hand implies algorithm performance that is independent of frame translation and rotation.

Frame indifference (or objectivity), is well known in classical mechanics [5], where physical laws dictate that this principle must hold. However, no corresponding law in optimization theory requires that an optimization algorithm must be frame invariant. It might therefore seem that frame indifference is merely an aesthetic requirement.

However, arguments in favour of frame invariant optimization algorithms do exist. Due to the lack of a physical justification, the arguments are necessarily based on a user’s perspective. If a particular optimization algorithm is frame dependent, it follows that there exists a specific choice of reference frame in which the problem can be solved easier (i.e. requiring less iterations) or better (i.e. achieving a lower cost function value) as compared to some other reference frame. In general,

the performance difference in different reference frames cannot be quantified, and depends amongst others on the optimization problem and algorithm specifics. Since *a priori* knowledge of the optimal reference frame for a particular problem is seldom available, this places an additional burden on the analyst, which now has to consider solving the problem in a number of reference frames. (An exception being the specialized class of separable functions.) If the algorithm's frame dependency is severe, the analyst requires some external procedure to take the algorithm's frame dependency into account. A conceptual procedure is to recast the optimization problem to simultaneously solve for the reference frame and solution, but this renders the problem ill posed.

An alternative phrasing of the above argument as follows. A frame-dependent algorithm implies a bias towards some particular reference frame (or frames). This in turn implies a bias towards some subclass of problems. If a problem is well suited to be solved in a particular reference frame, a frame-dependent algorithm that is *aligned* with this specific reference frame will perform well. This, however, implies that frame-*dependent* optimization algorithms are *specialist* algorithms, tuned to perform well on a special subclass of problems. In contrast, frame *invariant* optimization algorithms are *general* algorithms, applicable to a larger class of problems.

But, how do we know if a problem is well suited to be solved in a particular reference frame? In general, we do not. Therefore, the choice of a frame-dependent algorithm to solve a particular problem makes the *tacit assumption* that the problem's reference frame is well matched with the algorithm's reference frame bias. If the tacit assumption holds, good performance is expected, since the correct assumption implies additional problem information. Again, an example of practical relevance is the class of decomposable optimization problems, where an n -dimensional problem is simply the sum of n one-dimensional problems. An algorithm that independently searches along the co-ordinate axes, (which renders such an algorithm frame dependent), exploits this specific function characteristic. It is, therefore, expected that such an algorithm will be superior to its frame invariant counterpart.

To summarize, frame invariance of optimization algorithms is not a strict requirement. Frame invariance does, however, provide a useful classification of optimization algorithms. A frame invariant algorithm requires less *a priori* knowledge (or tacit assumptions) of the optimization problem, as compared to a frame variant algorithm. This implies that a frame invariant algorithm will have either inferior or superior performance as compared to a frame variant algorithm, depending on the validity of the assumed information.

Even though frame invariance is not required, some classes of optimization algorithms do satisfy this principle. In classical gradient based optimization [6], the gradient vector (or some conjugate direction to the gradient) indicates some direction of improvement, even if this direction is not optimal. The gradient vector of any scalar function satisfies the transformation rules for changing both scale and reference frame. This (usually) renders classical gradient-based optimization algorithms scale and reference frame invariant.

However, in so-called modern (stochastic) optimization procedures, like the PSO algorithm, few algorithms satisfy the requirement of reference frame invariance.

In this study, we will demonstrate that enforcement of reference frame invariance in the PSO may reduce directional diversity, which in turn aggravates the probability of premature convergence. We will, therefore, relax the requirements of reference frame invariance to only require that the mean objective function value over a large number of runs is invariant of the reference frame. We will define any algorithm which satisfies this relaxed requirement, to be *reference frame invariant* in a *stochastic sense*, which we consider a sufficient requirement for reference frame invariance of stochastic algorithms.

Previously, Salomon [7, 8] has pointed out that in stochastic optimization, most popular test functions are decomposable. For decomposable functions, the design variables are independent, i.e. the function consists of a summation of terms where each term contains only one design variable. For the genetic algorithm (GA), Salomon demonstrated the decrease in performance of the GA for a number of decomposable test functions, when these are arbitrary rotated w.r.t. the reference frame. Note that the rotated functions in general become non-decomposable, since they contain terms in the summation that may contain products of design variables. Incidentally: Solomon showed that the GAs performance at low mutation rates is vastly superior for decomposable functions, compared to the performance for non-decomposable functions.

As said, in this study we will consider whether the linear [2] and classical [3, 4] velocity update rules are invariant under scale, translation and rotation. To do so, we study the ‘basic’ PSO, i.e. we do not implement any heuristics such as maximum velocity restriction, dynamic inertia adjustment, position restriction or local neighbourhoods, since this will confuse the issue.

We will show that the linear velocity update rule is *invariant* under all of scale, translation and rotation, but that the classical velocity update rule is *invariant* under scale and translation only, and *variant* under rotation.

To illustrate that diversity and invariance are not necessarily exclusive, we propose an example of a velocity update rule that is both diverse and stochastically reference frame invariant. Our goal is not to propose yet another competitive and/or superior PSO variant, but merely to illustrate that formulations that are both diverse and invariant do exist. The new velocity update rule makes use of stochastically invariant perturbations of the search directions to generate diverse (space filling) particle search trajectories. The proposed formulation is also strictly invariant under scale and translation.

Our paper is constructed as follows: we present our methodology of investigation in Section 2. In Section 3, we present a brief overview of the PSO. We then study the linear and classical velocity update rules accordingly, in Sections 4 and 5, respectively. Our new proposed velocity update rule, denoted the *diverse rotationally invariant PSO* (DRI PSO), is then presented and studied in Section 6. In Section 7, we present numerical results for some popular test functions, in both the unrotated and rotated reference frames. We then present some further comments on the classical and the DRI velocity update rules, in Sections 8 and 9, respectively, whereafter we briefly discuss some popular heuristics in the context of reference frame invariance in Section 10. Finally, we conclude our study in Section 11.

2. BACKGROUND

Let us again consider Figure 1, which is understood to depict two related functions f and \hat{f} in the same Cartesian reference frame. The two functions are related through an arbitrary scale factor $s \in \mathbb{R}$, translation by an arbitrary vector $\mathbf{t} \in \mathbb{R}^n$, and rotation by an arbitrary proper orthogonal matrix \mathbf{Q} (henceforth denoted $\mathbf{Q} \in \text{Orth}^+$). Recall that a matrix \mathbf{Q} is orthogonal if and only if $\mathbf{Q}^T \mathbf{Q} = \mathbf{Q} \mathbf{Q}^T = \mathbf{I}$. A proper orthogonal matrix \mathbf{Q} has the additional property $\det(\mathbf{Q}) = +1$. By definition,

$$f(\mathbf{x}) = \hat{f}(\hat{\mathbf{x}}) \quad (1)$$

where \mathbf{x} and $\hat{\mathbf{x}}$ represent two design vectors. We denote any arbitrary scaled, translated and rotated quantity by a superscript caret (^).

It follows from vector analysis (e.g. see [9]) that the relation between the two vectors \mathbf{x} and $\hat{\mathbf{x}}$ is given by

$$\hat{\mathbf{x}} = s\mathbf{Q}\mathbf{x} + \mathbf{t} \quad (2)$$

for any $s \in \mathbb{R}$, $\mathbf{t} \in \mathbb{R}^n$ and $\mathbf{Q} \in \text{Orth}^+$. Substituting Equation (2) into Equation (1) gives

$$f(\mathbf{x}) = \hat{f}(s\mathbf{Q}\mathbf{x} + \mathbf{t}) \quad (3)$$

Although Equation (2) depicts a specific sequence of scaling, translation and rotation of \mathbf{x} , it is noted that it is in fact completely general, e.g.

$$\begin{aligned} \hat{\mathbf{x}} &= s\mathbf{Q}(\mathbf{x} + \mathbf{t}) \\ &= s\mathbf{Q}\mathbf{x} + s\mathbf{Q}\mathbf{t} \\ &= s\mathbf{Q}\mathbf{x} + \mathbf{c} \end{aligned} \quad (4)$$

where $\mathbf{c} = s\mathbf{Q}\mathbf{t}$ now represents the translation vector.

Let us consider an additional vector transformation rule. Consider two arbitrary vectors \mathbf{x}_1 and \mathbf{x}_2 and the difference vector $\mathbf{v} = \mathbf{x}_1 - \mathbf{x}_2$. In addition, consider related vectors $\hat{\mathbf{x}}_1$, $\hat{\mathbf{x}}_2$ and $\hat{\mathbf{v}} = \hat{\mathbf{x}}_1 - \hat{\mathbf{x}}_2$. The relation between \mathbf{v} and $\hat{\mathbf{v}}$ is

$$\begin{aligned} \hat{\mathbf{v}} &= \hat{\mathbf{x}}_1 - \hat{\mathbf{x}}_2 \\ &= (s\mathbf{Q}\mathbf{x}_1 + \mathbf{t}) - (s\mathbf{Q}\mathbf{x}_2 + \mathbf{t}) \\ &= s\mathbf{Q}(\mathbf{x}_1 - \mathbf{x}_2) \\ &= s\mathbf{Q}\mathbf{v} \end{aligned} \quad (5)$$

for any $s \in \mathbb{R}$ and $\mathbf{Q} \in \text{Orth}^+$.

Any optimization algorithm is scale, translation and rotation invariant *if and only if* the transformation rules given by Equations (2) and (5) are satisfied *for all* $\mathbf{Q} \in \text{Orth}^+$; the same optimal results are then obtained irrespective of the scale and reference frame used.

For a stochastic optimization procedure, we may choose to satisfy Equations (2) and (5) such that the algorithm's performance is invariant of scale, translation and rotation in a *stochastic sense*. We will refer to this as 'stochastic frame invariance'.

In this study, we consider bounded, unconstrained functions f and \hat{f} , respectively, defined over domains \mathbf{D} and $\hat{\mathbf{D}}$. $\mathbf{x} \in \mathbf{D}$ and $\hat{\mathbf{x}} \in \hat{\mathbf{D}}$ each represents a set of allowable design vectors, which are related through $\hat{\mathbf{D}} = s\mathbf{Q}\mathbf{D} + \mathbf{t}$.

3. BASIC FORMULATION OF THE PSO

Consider a swarm of p particles in an n -dimensional search space. The position vector \mathbf{x}_k^i of each particle i is updated by

$$\mathbf{x}_{k+1}^i = \mathbf{x}_k^i + \mathbf{v}_{k+1}^i \quad (6)$$

where k is a unit pseudotime increment (iteration number). \mathbf{v}_{k+1}^i represents the velocity vector that is obtained from the velocity rule, given by

$$\mathbf{v}_{k+1}^i = w\mathbf{v}_k^i + \mathbf{v}_k^i \quad (7)$$

where the inertia factor w is a real number, typically between 0.4 and 0.9 [10–12], and we define \mathbf{v}_k^i as the stochastic ‘velocity’ vector.

4. LINEAR VELOCITY UPDATE RULE

The stochastic vector \mathbf{v}_k^i for the linear velocity update rule [2] is given by

$$\mathbf{v}_k^i = c_1 r_{1k}^i (\mathbf{p}_k^i - \mathbf{x}_k^i) + c_2 r_{2k}^i (\mathbf{p}_k^g - \mathbf{x}_k^i) \quad (8)$$

Similarly, the transformed (scaled, translated and rotated) stochastic vector $\hat{\mathbf{v}}_k^i$ is given by

$$\begin{aligned} \hat{\mathbf{v}}_k^i &= c_1 r_{1k}^i (\hat{\mathbf{p}}_k^i - \hat{\mathbf{x}}_k^i) + c_2 r_{2k}^i (\hat{\mathbf{p}}_k^g - \hat{\mathbf{x}}_k^i) \\ &= c_1 r_{1k}^i (s\mathbf{Q}\mathbf{p}_k^i + \mathbf{t} - s\mathbf{Q}\mathbf{x}_k^i - \mathbf{t}) + c_2 r_{2k}^i (s\mathbf{Q}\mathbf{p}_k^g + \mathbf{t} - s\mathbf{Q}\mathbf{x}_k^i - \mathbf{t}) \\ &= s\mathbf{Q}c_1 r_{1k}^i (\mathbf{p}_k^i - \mathbf{x}_k^i) + s\mathbf{Q}c_2 r_{2k}^i (\mathbf{p}_k^g - \mathbf{x}_k^i) \\ &= s\mathbf{Q}(c_1 r_{1k}^i (\mathbf{p}_k^i - \mathbf{x}_k^i) + c_2 r_{2k}^i (\mathbf{p}_k^g - \mathbf{x}_k^i)) \\ &= s\mathbf{Q}\mathbf{v}_k^i \end{aligned} \quad (9)$$

Equation (9) makes use of both Equations (2) and (8). The position update rule for \mathbf{x}_{k+1}^i is

$$\mathbf{x}_{k+1}^i = \mathbf{x}_k^i + w\mathbf{v}_k^i + \mathbf{v}_k^i \quad (10)$$

and for $\hat{\mathbf{x}}_{k+1}^i$, it is

$$\begin{aligned} \hat{\mathbf{x}}_{k+1}^i &= \hat{\mathbf{x}}_k^i + w\hat{\mathbf{v}}_k^i + \hat{\mathbf{v}}_k^i \\ &= s\mathbf{Q}\mathbf{x}_k^i + \mathbf{t} + s\mathbf{Q}w\mathbf{v}_k^i + s\mathbf{Q}\mathbf{v}_k^i \\ &= s\mathbf{Q}(\mathbf{x}_k^i + w\mathbf{v}_k^i + \mathbf{v}_k^i) + \mathbf{t} \\ &= s\mathbf{Q}\mathbf{x}_{k+1}^i + \mathbf{t} \end{aligned} \quad (11)$$

where now we used Equations (2) and (10). The construction of the stochastic vector and position update rules of the linear velocity update rule given in Equations (9) and (11), respectively, satisfies the respective transformation rules given in Equations (5) and (2), for all $\mathbf{Q} \in \text{Orth}^+$. Hence, the linear velocity update rule is scale, translation and rotation invariant.

The intrinsic properties of a vector are its magnitude and direction; these exist independent of a reference frame [9]. In the linear velocity update rule, only the vector magnitudes (which are invariant) are randomly scaled. Consequently, the linear velocity update rule demonstrates ‘magnitudal diversity’, but lacks ‘directional diversity’ [1].

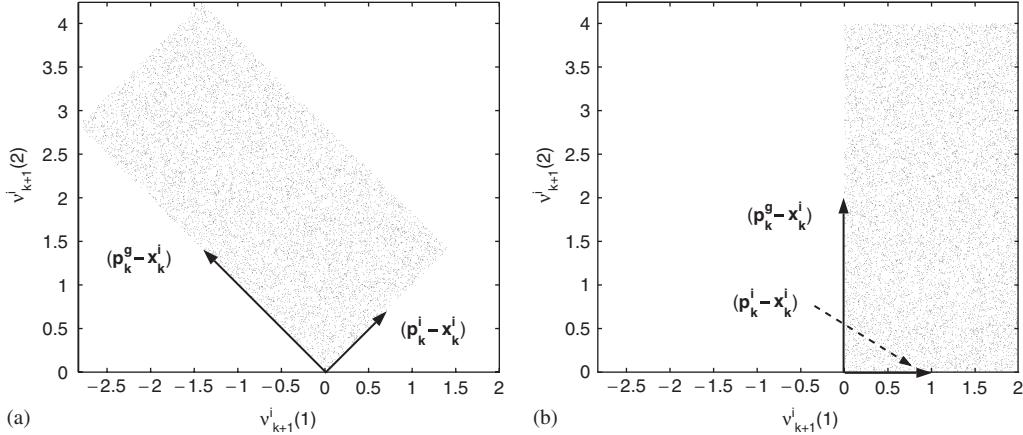


Figure 2. Linear PSO: scatter plot of 10^4 possible stochastic vectors v_k^i , generated using Monte Carlo simulations, with: (a) $(p_k^i - x_k^i) = [\frac{1}{\sqrt{2}} \frac{1}{\sqrt{2}}]$ and $(p_k^g - x_k^i) = [-\sqrt{2} \sqrt{2}]$; and (b) $(p_k^i - x_k^i) = [1 \ 0]$ and $(p_k^g - x_k^i) = [0 \ 2]$. Each point represents the end point of a stochastic vector with origin at $[0 \ 0]$.

4.1. Linear PSO: graphical demonstration of the instantaneous search domain

In a previous paper [1], the authors defined the instantaneous search domain χ_k^i of a given particle, which is composed of a deterministic contribution $x_k^i + wv_k^i$ and a stochastic contribution v_k^i . The stochastic domain is bounded, and has an associated probability distribution.

Further to our analytical investigation, and in order to illustrate the (stochastic) rotational invariance or lack thereof for the stochastic domain v_k^i , we use Monte Carlo simulations [13] to construct scatter plots. (Similar visualization techniques have been used in past research [14].)

These are conducted for different values of p_k^g , p_k^i and x_k^i . We construct scatter plots to define the domain of possible stochastic vectors v_k^i by generating 10^4 instances of v_k^i . We select $c_1 = c_2 = 2$ in all our investigations.

First, we conduct the study for non-parallel cognitive and social vectors $c_1(p_k^i - x_k^i)$ and $c_2(p_k^g - x_k^i)$. In Figure 2(a), the vectors $(p_k^i - x_k^i)$ and $(p_k^g - x_k^i)$ are given by $[\frac{1}{\sqrt{2}} \frac{1}{\sqrt{2}}]$ and $[-\sqrt{2} \sqrt{2}]$, respectively. A scatter plot yields the plane \mathcal{P}_k^i , with c_1 and c_2 merely scaling \mathcal{P}_k^i .

We then construct a scatter plot after rotating the vectors $(p_k^g - x_k^i)$ and $(p_k^i - x_k^i)$ 45° clockwise, as depicted in Figure 2(b). Hence $(p_k^i - x_k^i)$ and $(p_k^g - x_k^i)$ are given by $[1 \ 0]$ and $[0 \ 2]$, respectively. From Figure 2(b), it is clear that the domain remains a bounded plane \mathcal{P}_k^i , which is merely rotated 45° clockwise.

It also follows from random variable theory that the probability distribution over the domain \mathcal{P}_k^i is uniform [15], as illustrated in Figures 2(a) and (b).

Secondly, we conduct a similar study for parallel cognitive and social vectors $c_1(p_k^i - x_k^i)$ and $c_2(p_k^g - x_k^i)$, as depicted in Figure 3. In Figure 3(a), the parallel vectors $(p_k^i - x_k^i)$ and $(p_k^g - x_k^i)$ are given by $[\frac{1}{\sqrt{2}} \frac{1}{\sqrt{2}}]$ and $[\sqrt{2} \sqrt{2}]$, respectively. The domain is a bounded line \mathcal{L}_k^i with c_1 and c_2 merely scaling the length of \mathcal{L}_k^i .

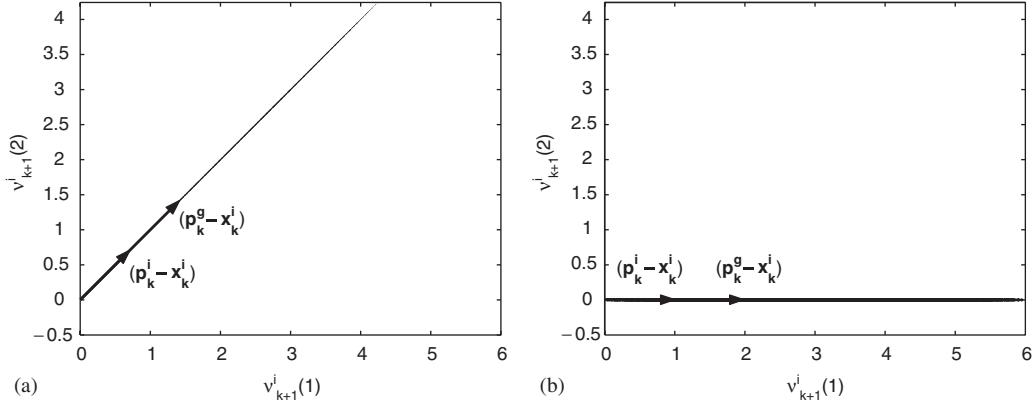


Figure 3. Linear PSO: scatter plot of 10^4 possible stochastic vectors \mathbf{v}_k^i , generated using Monte Carlo simulations, with: (a) $(\mathbf{p}_k^i - \mathbf{x}_k^i) = [\frac{1}{\sqrt{2}} \frac{1}{\sqrt{2}}]$ and $(\mathbf{p}_k^g - \mathbf{x}_k^i) = [\sqrt{2} \sqrt{2}]$; and (b) $(\mathbf{p}_k^i - \mathbf{x}_k^i) = [1 \ 0]$ and $(\mathbf{p}_k^g - \mathbf{x}_k^i) = [2 \ 0]$.

We again construct a scatter plot after rotating $(\mathbf{p}_k^g - \mathbf{x}_k^i)$ and $(\mathbf{p}_k^i - \mathbf{x}_k^i)$ 45° clockwise, as depicted in Figure 3(b). Now, $(\mathbf{p}_k^i - \mathbf{x}_k^i)$ and $(\mathbf{p}_k^g - \mathbf{x}_k^i)$ are given by $[1 \ 0]$ and $[2 \ 0]$, respectively. As shown in Figure 3 (b), the bounded line \mathcal{L}_k^i is merely rotated.

It follows from random variable theory [15] that the probability distribution over the bounded line is trapezoidal, for different vector lengths $(\mathbf{p}_k^i - \mathbf{x}_k^i)$ and $(\mathbf{p}_k^g - \mathbf{x}_k^i)$. It is triangular for the same vector lengths $(\mathbf{p}_k^i - \mathbf{x}_k^i)$ and $(\mathbf{p}_k^g - \mathbf{x}_k^i)$.

As discussed earlier and graphically demonstrated here, the linear velocity update rule is rotationally variant. A rotation of the vectors \mathbf{p}_k^i , \mathbf{p}_k^g and \mathbf{x}_k^i merely results in a rotation of the stochastic domains, \mathcal{P}_k^i and \mathcal{L}_k^i . This follows since we only scale the magnitude of the cognitive and social vectors in the linear velocity update rule. As a result the linear velocity update rule is reference frame *invariant*.

5. CLASSICAL VELOCITY UPDATE RULE

For the classical velocity update rule [3, 4] the stochastic vector \mathbf{v}_k^i is

$$\mathbf{v}_k^i = c_1 \Phi_{1k}^i (\mathbf{p}_k^i - \mathbf{x}_k^i) + c_2 \Phi_{2k}^i (\mathbf{p}_k^g - \mathbf{x}_k^i) \quad (12)$$

Similarly, the transformed stochastic vector $\hat{\mathbf{v}}_k^i$ is given by

$$\begin{aligned} \hat{\mathbf{v}}_k^i &= c_1 \hat{\Phi}_{1k}^i (\hat{\mathbf{p}}_k^i - \hat{\mathbf{x}}_k^i) + c_2 \hat{\Phi}_{2k}^i (\hat{\mathbf{p}}_k^g - \hat{\mathbf{x}}_k^i) \\ &= c_1 \hat{\Phi}_{1k}^i (s \mathbf{Q} \mathbf{p}_k^i + \mathbf{t} - s \mathbf{Q} \mathbf{x}_k^i - \mathbf{t}) + c_2 \hat{\Phi}_{2k}^i (s \mathbf{Q} \mathbf{p}_k^g + \mathbf{t} - s \mathbf{Q} \mathbf{x}_k^i - \mathbf{t}) \\ &= sc_1 \hat{\Phi}_{1k}^i \mathbf{Q} (\mathbf{p}_k^i - \mathbf{x}_k^i) + sc_2 \hat{\Phi}_{2k}^i \mathbf{Q} (\mathbf{p}_k^g - \mathbf{x}_k^i) \end{aligned} \quad (13)$$

Since $\hat{\mathbf{v}}_k^i$ is the summation of two difference vectors it follows from Equation (5) that the required transformation between $\hat{\mathbf{v}}_k^i$ and \mathbf{v}_k^i is given by

$$\begin{aligned}\hat{\mathbf{v}}_k^i &= s\mathbf{Q}\mathbf{v}_k^i \\ &= s\mathbf{Q}(c_1\Phi_{1k}^i(\mathbf{p}_k^i - \mathbf{x}_k^i) + c_2\Phi_{2k}^i(\mathbf{p}_k^g - \mathbf{x}_k^i)) \\ &= sc_1\mathbf{Q}\Phi_{1k}^i(\mathbf{p}_k^i - \mathbf{x}_k^i) + sc_2\mathbf{Q}\Phi_{2k}^i(\mathbf{p}_k^g - \mathbf{x}_k^i)\end{aligned}\quad (14)$$

Comparing the actual transformation in Equation (13) to the required transformation in Equation (14), satisfaction of scale and frame invariance requires

$$\hat{\Phi}_{lk}^i \mathbf{Q} = \mathbf{Q}\Phi_{lk}^i \implies \hat{\Phi}_{lk}^i = \mathbf{Q}\Phi_{lk}^i \mathbf{Q}^T \quad \text{for } l = 1, 2 \quad \text{and} \quad \forall \mathbf{Q} \in \text{Orth}^+ \quad (15)$$

Since the relation between $\hat{\Phi}_{lk}^i$ and Φ_{lk}^i has to hold $\forall \mathbf{Q} \in \text{Orth}^+$, the solution to Equation (15) is

$$\hat{\Phi}_{lk}^i = \Phi_{lk}^i = a_l^i \mathbf{I} \quad \text{for } l = 1, 2 \quad (16)$$

a_1^i and a_2^i are any real scalar values. If we choose these values to be uniform random scalars, the classical velocity update rule reduces to the linear velocity update rule.

But, $\hat{\Phi}_{lk}^i$ and Φ_{lk}^i are random matrices, generated independently. Therefore, frame invariance is not satisfied. By scaling the components in the classical velocity update rule, both the vector magnitudes and vector directions are stochastically varied, thereby introducing both magnitude and directional diversity. The property of frame invariance is, however, sacrificed.

Accepting that the classical velocity update rule is rotationally variant, we now substitute $\mathbf{Q} = \mathbf{I}$ for the remainder of the investigation, to evaluate only the scale and translational (in)variance of the classical PSO. For this case we assume that the random matrices $\hat{\Phi}_{lk}^i$ and Φ_{lk}^i are generated to be equal. The position update rule \mathbf{x}_{k+1}^i is now given by

$$\mathbf{x}_{k+1}^i = \mathbf{x}_k^i + w\mathbf{x}_k^i + \mathbf{v}_k^i \quad (17)$$

Similarly, the transformed position update rule for $\hat{\mathbf{x}}_{k+1}^i$ is given by

$$\begin{aligned}\hat{\mathbf{x}}_{k+1}^i &= \hat{\mathbf{x}}_k^i + w\hat{\mathbf{v}}_k^i + \hat{\mathbf{v}}_k^i \\ &= s\mathbf{x}_k^i + \mathbf{t} + sw\mathbf{v}_k^i + s(c_1\Phi_{1k}^i(\mathbf{p}_k^i - \mathbf{x}_k^i) + c_2\Phi_{2k}^i(\mathbf{p}_k^g - \mathbf{x}_k^i)) \\ &= s(\mathbf{x}_k^i + w\mathbf{v}_k^i + c_1\Phi_{1k}^i(\mathbf{p}_k^i - \mathbf{x}_k^i) + c_2\Phi_{2k}^i(\mathbf{p}_k^g - \mathbf{x}_k^i)) + \mathbf{t} \\ &= s\mathbf{x}_{k+1}^i + \mathbf{t}\end{aligned}\quad (18)$$

Equation (18) shows that the classical velocity update rule is strictly *invariant* of scale and translation, since it satisfies Equation (2) exactly.

In addition to scale invariance of the classical velocity update rule, Schutte *et al.* [16] showed that the classical velocity update rule is even invariant of independent component scaling. Note that this, however, necessarily comes at the cost of *sacrificing* rotational invariance.

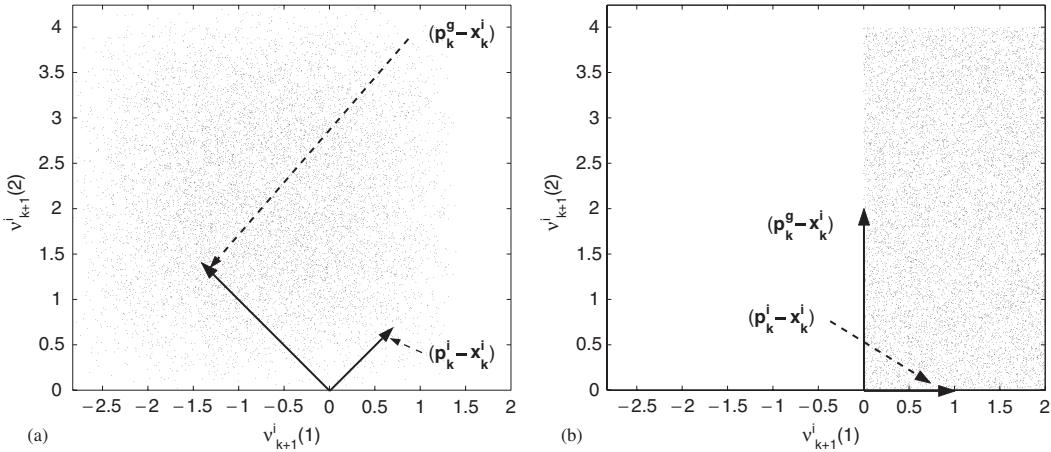


Figure 4. Classical PSO: scatter plot of 10^4 possible stochastic vectors v_k^i , generated using Monte Carlo simulations with: (a) $(\mathbf{p}_k^i - \mathbf{x}_k^i) = [\frac{1}{\sqrt{2}} \frac{1}{\sqrt{2}}]$ and $(\mathbf{p}_k^g - \mathbf{x}_k^i) = [-\sqrt{2} \sqrt{2}]$; and (b) $(\mathbf{p}_k^i - \mathbf{x}_k^i) = [1 0]$ and $(\mathbf{p}_k^g - \mathbf{x}_k^i) = [0 2]$.

While the classical velocity update rule is rotational *variant*, we may argue that it is possible to approximate the rotation invariance requirement $c_l \Phi_{lk}^i = \mathbf{I}$, $l = 1, 2$ in a stochastic sense, such that the algorithm's performance is 'approximately invariant'. We can allow for a stochastic deviation from $c_l \Phi_{lk}^i = \mathbf{I}$, $l = 1, 2$ within a 'small' range around \mathbf{I} , with some associated distribution. We will do so in Section 8 to come.

5.1. Classical PSO: graphical demonstration of the instantaneous search domain

We now illustrate the rotational *variance* of the classical velocity update rule using Monte Carlo simulations, similar to those we have performed in Section 4.1.

As before, we conduct the study for non-parallel cognitive and social vectors $c_1(\mathbf{p}_k^i - \mathbf{x}_k^i)$ and $c_2(\mathbf{p}_k^g - \mathbf{x}_k^i)$. Figure 4(a) depicts that the domain is an n -dimensional space \mathcal{S}_k^i (with $n = 2$ in this case), with c_1 and c_2 merely scaling \mathcal{S}_k^i . It is also clear that the probability distribution over \mathcal{S}_k^i is non-uniform.

The scatter plot after rotating the vectors $(\mathbf{p}_k^g - \mathbf{x}_k^i)$ and $(\mathbf{p}_k^i - \mathbf{x}_k^i)$ 45° clockwise is depicted in Figure 4(b). It is clear that the *domain changes* after rotation of the vectors. However, the domain remains an n -dimensional space \mathcal{S}_k^i , but the size of, and the probability distribution over, the domain depends on the orientation w.r.t. the Cartesian co-ordinate axis.

We now again conduct the study for parallel cognitive and social vectors $c_1(\mathbf{p}_k^i - \mathbf{x}_k^i)$ and $c_2(\mathbf{p}_k^g - \mathbf{x}_k^i)$, as depicted in Figure 5(a). The domain is still generalized to n -dimensional space \mathcal{S}_k^i with c_1 and c_2 merely scaling the size of \mathcal{S}_k^i .

The scatter plot after rotating $(\mathbf{p}_k^g - \mathbf{x}_k^i)$ and $(\mathbf{p}_k^i - \mathbf{x}_k^i)$ 45° clockwise is depicted in Figure 5(b). It is clear that the domain changes significantly after rotation of the vectors. In fact, the domain collapses to a bounded line \mathcal{L}_k^i , since both vectors are parallel to one of the Cartesian basis vectors.

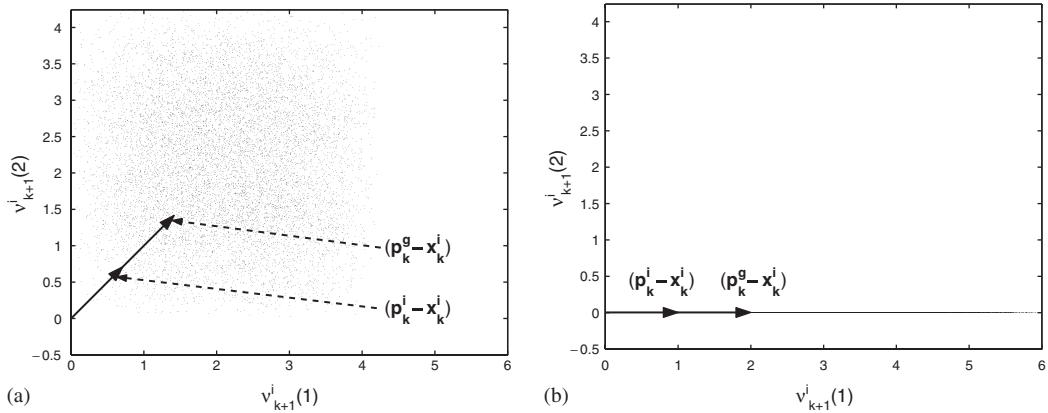


Figure 5. Classical PSO: scatter plot of 10^4 possible stochastic vectors v_k^i , generated using Monte Carlo simulations with: (a) $(\mathbf{p}_k^i - \mathbf{x}_k^i) = [\frac{1}{\sqrt{2}} \frac{1}{\sqrt{2}}]$ and $(\mathbf{p}_k^g - \mathbf{x}_k^i) = [\sqrt{2} \sqrt{2}]$; and (b) $(\mathbf{p}_k^i - \mathbf{x}_k^i) = [1 \ 0]$ and $(\mathbf{p}_k^g - \mathbf{x}_k^i) = [2 \ 0]$.

As discussed earlier and graphically demonstrated here, the classical velocity update rule is rotation *variant*. A rotation of the vectors $(\mathbf{p}_k^i - \mathbf{x}_k^i)$ and $(\mathbf{p}_k^g - \mathbf{x}_k^i)$ results in the size of, and the probability distribution over, the stochastic domain to change. This follows since the classical velocity update rule scales the *components* of the cognitive and social vectors. Since the components of a vector are reference frame variant, the classical velocity update rule is also reference frame variant.

However, the advantage of the classical velocity update rule is that the particle trajectories remain space filling in n -dimensional space as the authors have shown in a previous paper [1]. The result is that directional diversity in particle trajectories is maintained.

6. DIVERSE ROTATIONALLY INVARIANT (DRI) VELOCITY UPDATE RULE

As discussed in Section 4, the linear velocity update rule is rotationally *invariant*, although the particle trajectories collapse to lines. The advantage of directionally diverse (n -dimensional) particle search trajectories was quantified in a previous paper [1]. On the other hand, the classical velocity update rule allows for particles to have directionally diverse search trajectories, but unfortunately this comes at the cost of rotational *variance*.

To illustrate that diversity and invariance are not necessarily exclusive, we propose an example of a velocity update rule that is both diverse and stochastically reference frame invariant. Our goal is not to propose yet another competitive and/or superior PSO variant, but merely to illustrate that formulations that are both diverse and invariant do exist.

Based on the linear velocity update rule, our directionally diverse velocity update rule is denoted the DRI velocity update rule. We randomly scale the vector magnitudes of $(\mathbf{p}_k^i - \mathbf{x}_k^i)$ and $(\mathbf{p}_k^g - \mathbf{x}_k^i)$. In addition, we impose ‘small’ perturbations of the vector directions of $(\mathbf{p}_k^i - \mathbf{x}_k^i)$ and $(\mathbf{p}_k^g - \mathbf{x}_k^i)$. We perturb the vector directions by multiplying each of the above vectors with an independent

random rotation matrix \mathbf{R} . The random rotation matrices are constructed anew for each particle i and for every iteration k , i.e.

$$\mathbf{v}_k^i = c_1 r_{1k}^i \mathbf{R}_{1k}^i (\mathbf{p}_k^i - \mathbf{x}_k^i) + c_2 r_{2k}^i \mathbf{R}_{2k}^i (\mathbf{p}_k^g - \mathbf{x}_k^i) \quad (19)$$

with each \mathbf{R}_{lk}^i , $l = 1, 2$, a random rotation matrix of dimension $n \times n$. (A computationally viable alternative to these rotation matrices is discussed in Section 9.1.)

Similarly, the transformed stochastic vector $\hat{\mathbf{v}}_k^i$ is given by

$$\hat{\mathbf{v}}_k^i = c_1 r_{1k}^i \hat{\mathbf{R}}_{1k}^i (\hat{\mathbf{p}}_k^i - \hat{\mathbf{x}}_k^i) + c_2 r_{2k}^i \hat{\mathbf{R}}_{2k}^i (\hat{\mathbf{p}}_k^g - \hat{\mathbf{x}}_k^i) \quad (20)$$

Comparing Equations (19) and (20) to Equations (12) and (13), we notice that \mathbf{R}_{lk}^i simply replaces Φ_{lk}^i . The scale and frame invariance analysis, therefore, proceeds exactly as in Section 5, with the final requirement for invariance

$$\hat{\mathbf{R}}_{lk}^i = \mathbf{R}_{lk}^i = a_l^i \mathbf{I} \quad \text{for } l = 1, 2 \quad (21)$$

Exactly as before, a strict enforcement of rotational invariance reduces the DRI velocity update rule to the linear velocity update rule.

However, since we deal with a stochastic algorithm, we may choose to satisfy Equation (21) in an average sense only. In order to satisfy $\mathbf{R}_{lk}^i = \mathbf{I}$ in a stochastic sense, we will show numerically that it is sufficient to require that the rotations are ‘small’.

The remainder of the proof for scale and translation invariance is identical to the proof presented for the linear velocity update rule in Section 4.

As a result, the DRI velocity update rule is strictly scale and translation invariant, and rotationally invariant in a stochastic sense.

6.1. Random rotation matrix construction

Let us consider the construction of a proper orthogonal matrix \mathbf{R} . Numerous methods are available to construct rotation matrices, e.g. see the approach of Salomon [7]. Constructing $n \times n$ matrices using Salomon’s routine is, however, computationally expensive, since $(n-1)(n-2)$ matrix–matrix multiplications are required.

As a computationally economical alternative, we use the exponential map [17]. There are again numerous ways to construct exponential maps. We select the simple series method [17]. The general series expansion of an exponential map is given by

$$\mathbf{R} = \mathbf{I} + \frac{1}{2} \mathbf{W} \mathbf{W} + \frac{1}{6} \mathbf{W} \mathbf{W} \mathbf{W} + \dots \quad (22)$$

where \mathbf{I} is the identity matrix and \mathbf{W} is a skew matrix.

We construct the random skew matrix \mathbf{W} as follows:

$$\mathbf{W} = \frac{\alpha\pi}{180} (\mathbf{A} - \mathbf{A}^T) \quad (23)$$

with \mathbf{A} an $n \times n$ random matrix with each entry a uniform random number $\in [-0.5, 0.5]$, and α a real scaling factor.

Since we are only interested in ‘small’ rotations, we choose to construct the exponential map \mathbf{R}_k^i using only the first two terms of a truncated series method, i.e.

$$\mathbf{R}_k^i = \mathbf{I} + \mathbf{W}_k^i \quad (24)$$

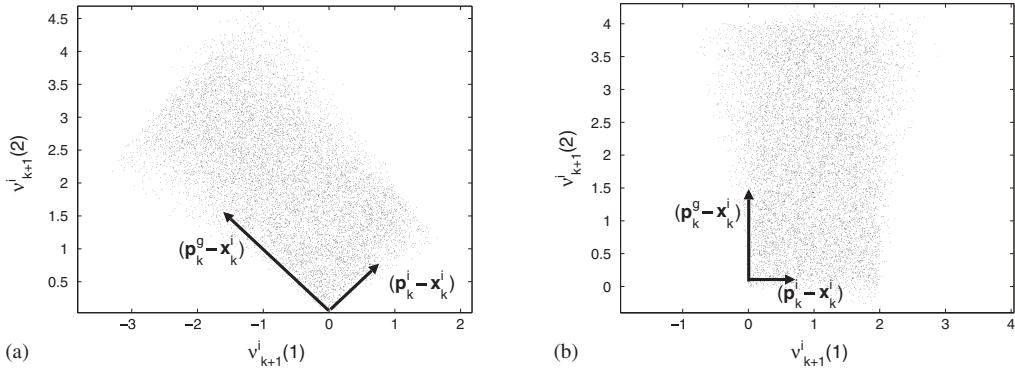


Figure 6. DRI PSO: scatter plot of 10^4 possible stochastic vectors \mathbf{v}_k^i , generated using Monte Carlo simulations, with: (a) $(\mathbf{p}_k^i - \mathbf{x}_k^i) = [\frac{1}{\sqrt{2}} \frac{1}{\sqrt{2}}]$ and $(\mathbf{p}_k^g - \mathbf{x}_k^i) = [-\sqrt{2} \sqrt{2}]$; and (b) $(\mathbf{p}_k^i - \mathbf{x}_k^i) = [1 \ 0]$ and $(\mathbf{p}_k^g - \mathbf{x}_k^i) = [0 \ 2]$.

This is the linear approximation to a rotation matrix, and is valid for small perturbations, since the entries of the higher order terms are close to zero. The advantage of the simplification is that the number of matrix–matrix multiplications in constructing \mathbf{R}_k^i is zero.

6.2. DRI PSO: investigation of the instantaneous search domain

As before, we quantify the stochastic rotation invariance of the DRI PSO using Monte Carlo simulations. In two dimensions, we select $\alpha=15$. (Although this is not ‘small’, this serves to clearly illustrate the proposed concept.)

Again, we conduct the study for non-parallel cognitive and social vectors, as depicted in Figure 6(a). The domain generalizes to n -dimensional space \mathcal{S}_k^i , with c_1 and c_2 scaling \mathcal{S}_k^i .

The scatter plot after rotating the vectors $(\mathbf{p}_k^g - \mathbf{x}_k^i)$ and $(\mathbf{p}_k^i - \mathbf{x}_k^i)$ 45° clockwise is depicted in Figure 6(b). Clearly, the domain remains generalized to n -dimensional space \mathcal{S}_k^i , rotated 45° clockwise. The probability distribution over the domain \mathcal{S}_k^i is non-uniform.

Secondly, we conduct the study for parallel cognitive and social vectors $c_1(\mathbf{p}_k^i - \mathbf{x}_k^i)$ and $c_2(\mathbf{p}_k^g - \mathbf{x}_k^i)$, as depicted in Figure 7(a). Again the domain generalizes to n -dimensional space \mathcal{S}_k^i , with c_1 and c_2 merely scaling the domain.

We now construct a scatter plot after rotating $(\mathbf{p}_k^g - \mathbf{x}_k^i)$ and $(\mathbf{p}_k^i - \mathbf{x}_k^i)$ 45° clockwise, as depicted in Figure 7(b). Evidently, the n -dimensional space \mathcal{S}_k^i is merely rotated, and the probability distribution over the domain is non-uniform.

As discussed earlier and graphically demonstrated here, the DRI PSO is reference frame invariant in a stochastic sense. A rotation of the vectors $(\mathbf{p}_k^i - \mathbf{x}_k^i)$ and $(\mathbf{p}_k^g - \mathbf{x}_k^i)$ merely results in a rotation of the stochastic domain \mathcal{S}_k^i .

The drawback of the linear PSO is that the particle trajectories collapse to line searches, i.e. it lacks directional diversity. This is a result of stochastically scaling only the vector magnitudes, and not the directions as well.

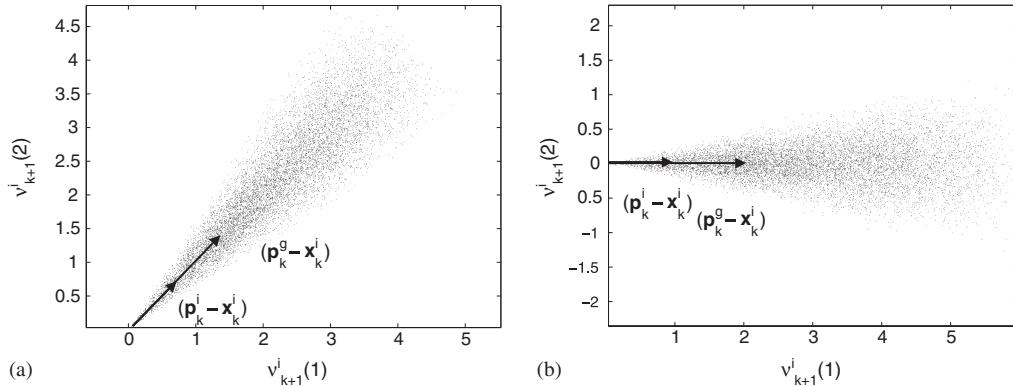


Figure 7. DRI PSO: scatter plot of 10^4 instances of the stochastic vectors \mathbf{v}_k^i , generated using Monte Carlo simulations, with: (a) $(\mathbf{p}_k^i - \mathbf{x}_k^i) = [\frac{1}{\sqrt{2}} \frac{1}{\sqrt{2}}]$ and $(\mathbf{p}_k^g - \mathbf{x}_k^i) = [\sqrt{2} \sqrt{2}]$; and (b) $(\mathbf{p}_k^i - \mathbf{x}_k^i) = [1 \ 0]$ and $(\mathbf{p}_k^g - \mathbf{x}_k^i) = [2 \ 0]$.

The drawbacks of the classical PSO is that it is rotationally variant and that the magnitude and directional diversity are coupled. It is clear from Section 5 that satisfying the requirements of rotation invariance reduces the classical PSO to the linear PSO, i.e. it then lacks directional diversity.

The drawback of both is overcome in the DRI PSO, where in addition to scaling the vector magnitudes, the vector directions are also ‘slightly’ perturbed. We allow ourselves to stochastically vary both magnitudes and directions of $(\mathbf{p}_k^i - \mathbf{x}_k^i)$ and $(\mathbf{p}_k^g - \mathbf{x}_k^i)$, thereby introducing magnitude and directional diversity. Since we independently vary the vector magnitude and vector directions, the magnitude and directional diversity in the DRI PSO are *uncoupled*.

This is in contrast to the linear PSO, where we *only* update the magnitude stochastically, while the direction update is deterministic. Hence the linear PSO has *adequate* diversity in *magnitude*, while it lacks *directional* diversity. (Incidentally, termination occurs when \mathbf{p}_k^i , \mathbf{p}_k^g and \mathbf{x}_k^i converge on the same point in n -dimensional space, combined with $w\mathbf{v}_k^i \rightarrow \mathbf{0}$.)

7. NUMERICAL EXPERIMENTS

We now perform an empirical study to quantify the (lack of) rotational invariance of the three discussed velocity update rules of the PSO. We therefore study a popular test set in the unrotated reference frame $f(\mathbf{x})$, as well as in an arbitrary rotated reference frame $\hat{f}(\mathbf{Q}\mathbf{x})$ [8]. Here, \mathbf{Q} is a random, proper orthogonal transformation matrix, constructed as in [7]. (\mathbf{Q} is not to be confused with \mathbf{R} , introduced in previous sections.) The transformation matrix results in a pure rotation of each test function.

The aim is *not* an exhaustive determination of optimal algorithmic parameters, but merely to quantify rotational *variance* of the classical PSO compared to the linear and DRI PSO, which are rotationally invariant. We do so without additional heuristics such as maximum velocity restriction, position restriction, craziness or dynamic inertia reduction.

We select $c_1 = c_2 = 2$, a swarm size of $p = 20$ particles, initial velocities $\mathbf{v}_0^i = \mathbf{0}$, a simple synchronous updating scheme [18] and for the DRI PSO, we simply select $\alpha = 3$. We do not seek an optimal value for α , but merely wish to illustrate the effects of perturbing the vector directions. Initial positions \mathbf{x}_0^i are generated randomly within the entire search space. Real variables are implemented using *double-precision floating-point* arithmetic according to the *IEEE Standard for Binary Floating-Point Arithmetic*, commonly referred to as ‘IEEE 754’.

We want to investigate the effect of inertia w . To do this, we average the performance over 100 runs (each run terminates after 10 000 iterations), with w kept constant. This procedure is repeated for w between 0 and 1, in 0.1 increments. For each of the 100 independent runs, a new random rotation matrix \mathbf{Q} is constructed, to ensure that there is no bias towards any particular reference frame.

We use the following five test functions that are very popular in PSO research:

- (i) the lesser known variant of the extended Rosenbrock function [1]:

$$f_0(\mathbf{x}) = \sum_{i=1}^{n/2} (100(x_{2i} - x_{2i-1}^2)^2 + (1 - x_{2i-1})^2)$$

- (ii) the Quadric function (unimodal):

$$f_1(\mathbf{x}) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$$

- (iii) the Ackley function (multimodal):

$$f_2(\mathbf{x}) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) \\ - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) + 20 + e$$

- (iv) the generalized Rastrigin function (multimodal):

$$f_3(\mathbf{x}) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$$

- (v) and finally, the generalized Griewank function (multimodal):

$$f_4(\mathbf{x}) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$$

All problems have dimension $n = 30$, and each component of the initial positions are limited between ± 2.048 , ± 100 , ± 30 , ± 5.12 and ± 600 , for the respective problems. The global best objective function value for all these test problems are zero. Except for the Rosenbrock function,

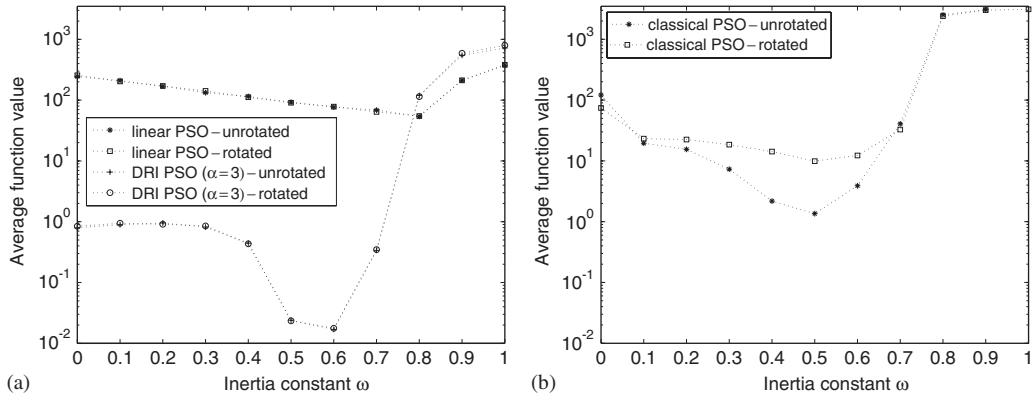


Figure 8. Average function value obtained with: (a) the linear PSO; and DRI PSO; and (b) classical PSO after 10^4 iterations, averaged over 100 runs, for the rotated and unrotated Rosenbrock test function f_0 .

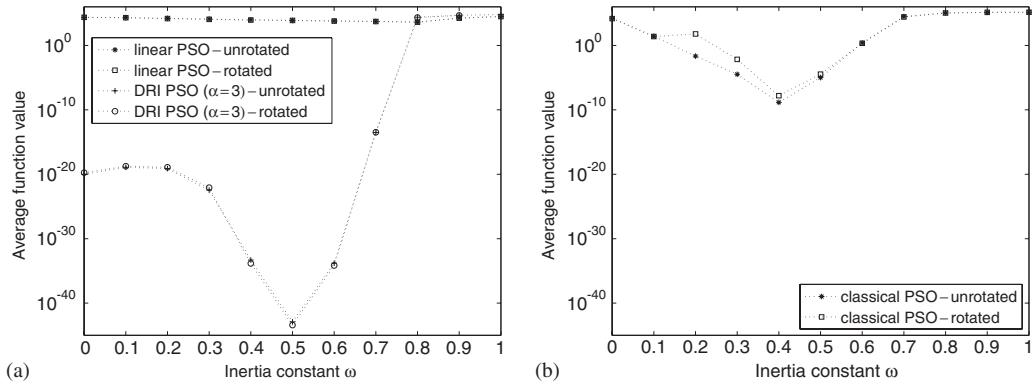


Figure 9. Average function value obtained with: (a) the linear PSO and the DRI PSO; and (b) classical PSO after 10^4 iterations averaged over 100 runs for the rotated and unrotated Quadric test function f_1 .

which has solution vector $\mathbf{x}^* = [1, 1, \dots, 1]^T$, all other test problems have the solution vector $\mathbf{x}^* = [0, 0, \dots, 0]^T$.

Depicted in Figures 8–12 are the mean objective function values after 10 000 iterations averaged over 100 runs for both the unrotated and rotated functions, for the five test problems under consideration.

The *rotational invariance* of the linear PSO and the *stochastic rotational invariance* of the DRI PSO are evident from Figures 8(a), 9(a), 10(a), 11(a) and 12(a). The poor performance of the linear PSO directly results from the particle trajectories collapsing to lines [1]. There is a significant improved performance for all the test functions with the DRI PSO, due to the scaling of the vector magnitudes and perturbation of the vector directions.

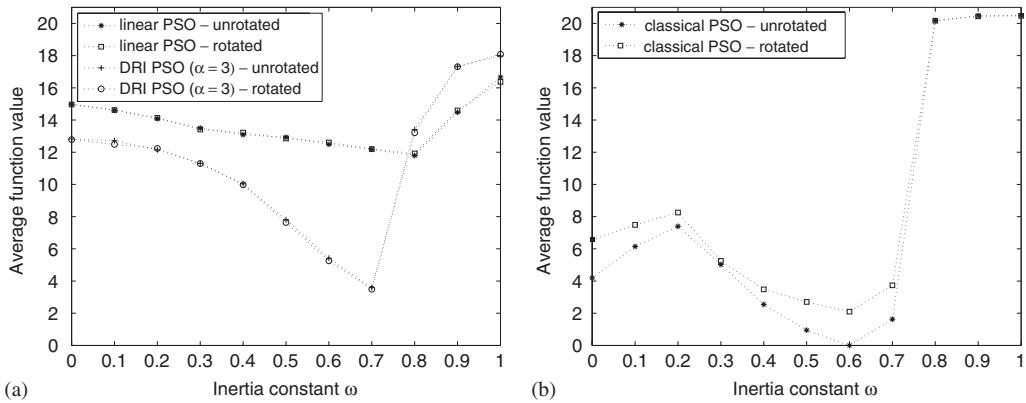


Figure 10. Average function value obtained with: (a) the linear PSO and the DRI PSO; and (b) the classical PSO after 10^4 iterations averaged over 100 runs for the rotated and unrotated Ackley test function f_2 .

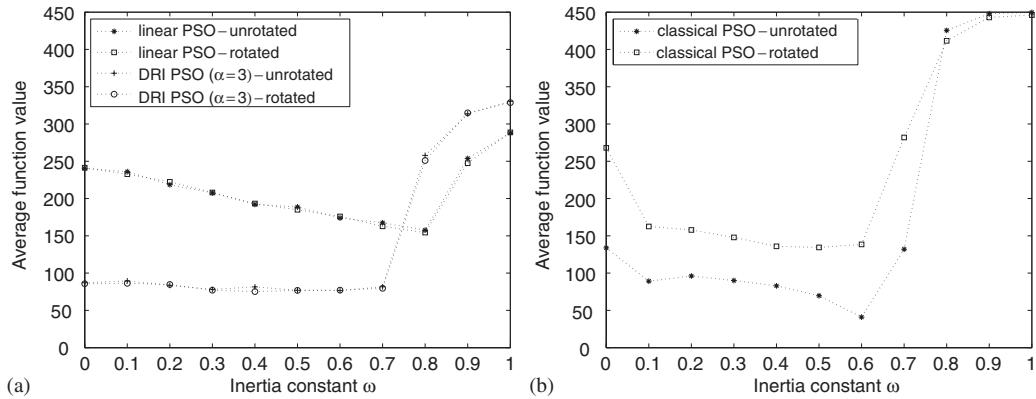


Figure 11. Average function value obtained with: (a) the linear PSO and the DRI PSO; and (b) the classical PSO after 10^4 iterations averaged over 100 runs for the rotated and unrotated Rastrigin test function f_3 .

The *rotational variance* of the classical PSO is evident from Figures 8(b), 9(b), 10(b), 11(b) and 12(b). There is a *severe* performance loss for some of the rotated functions compared to the unrotated functions, in particular, for the Rastrigin test function f_3 .

With the classical PSO, two functions result in similar performance for the rotated and unrotated functions, namely the Quadric function f_1 , and the Griewank function, f_4 . The Quadric and Griewank functions are almost insensitive to rotation. (The Griewank function is a spherical function on which sinusoidal ‘noise’ is imposed. There are thousands of local minima in $\hat{\mathbf{D}}$. Hence this function is artificially indifferent to rotation, since many ‘good’ local minima appear, irrespective of rotation.) Inadvertently, this also suggests that non-spherical unimodal test functions should be used to evaluate rotational invariance. The two functions with a single local minimum, namely the Rosenbrock function f_0 and the Quadric function f_1 , are of further interest, since

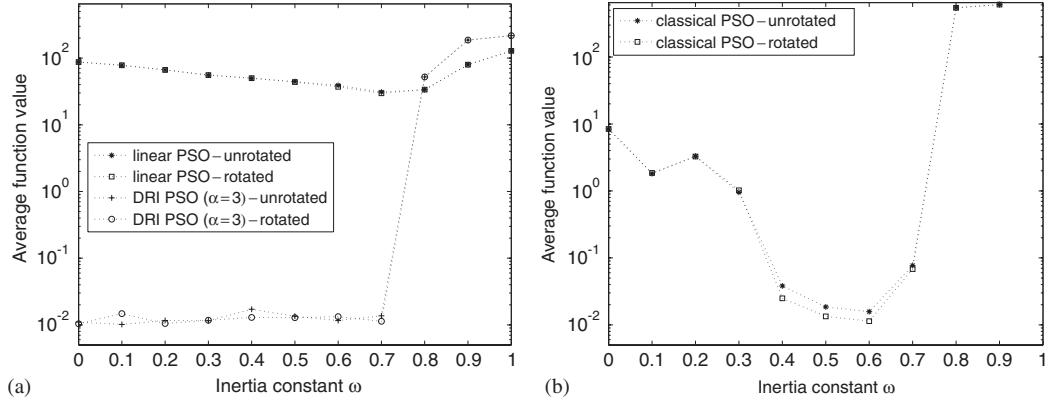


Figure 12. Average function value obtained with: (a) the linear PSO and the DRI PSO; and (b) the classical PSO after 10^4 iterations averaged over 100 runs for the rotated and unrotated Griewank test function f_4 .

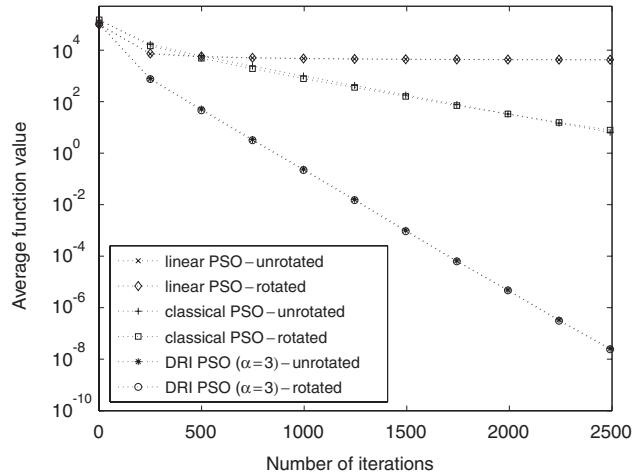


Figure 13. Mean function value history plot averaged over 100 runs for the rotated and unrotated Quadric test function f_1 with the linear PSO (with $w=0.8$), the classical PSO (with $w=0.4$) and the DRI PSO (with $w=0.5$ and $\alpha=3$).

they indicate the ability of an algorithm to search within a local basin. The performance of the DRI PSO is significantly better than the classical PSO for both functions, for both the rotated and unrotated test functions. Also, the classical PSO demonstrates a severe performance loss for the Rosenbrock function, for the rotated function compared to the unrotated function. (Note the scale of the graphs in Figure 8.)

The performance difference between the classical PSO and the DRI PSO for the unimodal Quadric test function f_1 is depicted in Figure 13, which depicts the mean function value convergence history of the linear (with $w=0.8$), the classical (with $w=0.4$) and the DRI PSO (with $w=0.5$ and $\alpha=3$) over 2500 iterations. The values for w are optimal for each algorithm, but no attempt

Table I. Constant inertia factor w at which the best average objective function value is obtained for the unrotated $f_{\text{ave}}^{\text{best}}|_U$ and rotated $f_{\text{ave}}^{\text{best}}|_R$ test functions.

Function	w	$f_{\text{ave}}^{\text{best}} _U$	$f_{\text{ave}}^{\text{best}} _R$
<i>Linear velocity update rule</i>			
f_0	0.8	54.1	54.7
f_1	0.8	409×10^1	412×10^1
f_2	0.8	11.8	11.9
f_3	0.8	158	154
f_4	0.7	30.9	29.7
<i>Classical velocity update rule</i>			
f_0	0.5	1.36	9.91
f_1	0.4	1.44×10^{-9}	1.56×10^{-8}
f_2	0.6	8.61×10^{-15}	2.09
f_3	0.6	40.9	138
f_4	0.6	1.51×10^{-2}	1.13×10^{-2}
<i>DRI velocity update rule</i>			
f_0	0.6	1.66×10^{-2}	1.72×10^{-2}
f_1	0.5	1.27×10^{-43}	3.81×10^{-44}
f_2	0.7	3.58	3.49
f_3	0.5	76.9	76.9
f_4	0.6	1.15×10^{-2}	1.30×10^{-2}

was made to optimize α . Of the three formulations, it is clear that the DRI PSO is computationally the most effective on the Quadric test function.

For the sake of clarity, an overview of the performances of the linear, the classical and the DRI PSO is given in Table I. The table summarizes the best function values obtained, together with the inertia factor at which the best function value is obtained after 10 000 iterations. Results for both the unrotated and rotated test functions are given.

8. COMMENTS ON THE CLASSICAL PSO

For the classical PSO, we now consider some alternatives to the rotational invariance requirement $c_l \Phi_{lk}^i = \mathbf{I}$. In particular, we investigate the effect of different probability distribution functions which have a mean of $c_l \Phi_{lk}^i = \mathbf{I}$, as depicted in Figure 14. Throughout, we use $c_1 = c_2 = 2$.

For each random number in Φ_{lk}^i , generated from the uniform distribution $\in [0 \ 1]$, the probability distribution function (PDF) is depicted in Figure 14(a). This is of course the classical implementation we have considered in the foregoing sections.

We then consider random numbers generated from the bilinear distribution over $\in [0 \ 1]$ depicted in Figure 14(b). This decreases rotational variance, as demonstrated in Figure 15(a) for the unrotated and rotated Rastrigin test function f_3 .

Finally, the random numbers are generated from a uniform distribution over $\in [0.25 \ 0.75]$ (Figure 14(c)). Figure 15(b) illustrates that the classical PSO now becomes ‘almost’ rotationally invariant, as again demonstrated for the unrotated and rotated Rastrigin test function f_3 .

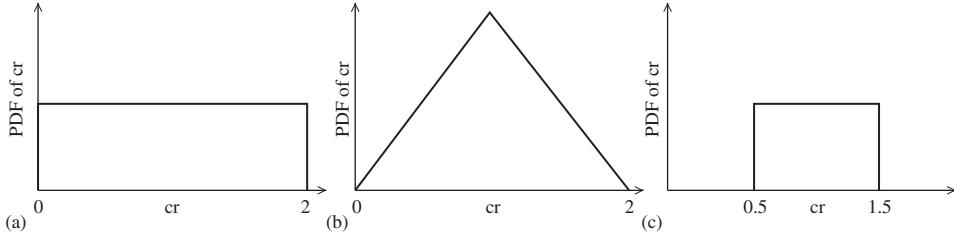


Figure 14. Alternative probability distribution functions (PDF) of cr such that the $\text{mean}(cr) = 1$ for $c = 2$:
 (a) r uniform $\in [0; 1]$; (b) r bilinear $\in [0; 1]$; and (c) r uniform $\in [0.25; 0.75]$.

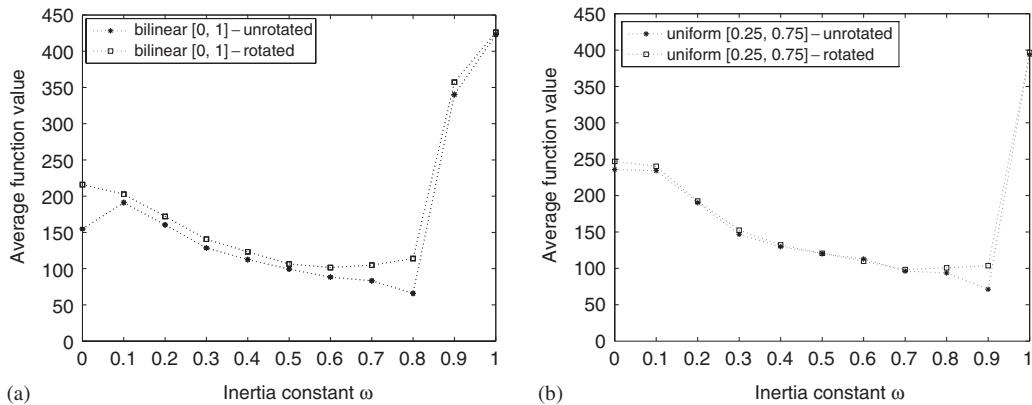


Figure 15. Function value after 10^4 iterations, averaged over 100 runs, obtained using the classical PSO, for the rotated and unrotated Rastrigin test function f_3 , with random numbers generated from a: (a) bilinear distribution between 0 and 1; and (b) uniform distribution between 0.25 and 0.75.

Comparing Figure 11(b) to Figures 15(a) and (b), it is clear that the classical PSO becomes ‘more’ rotationally invariant as the deviation from $c_l \Phi_{lk}^i = \mathbf{I}$ decreases. This, however, comes at the cost of sacrificing both magnitude and directional diversity. For the classical PSO this reduction in diversity is especially pronounced since the magnitude and directional diversity are *coupled*. Therefore, a reduction in the directional diversity inadvertently reduces the magnitude diversity, and *vice versa*.

9. COMMENTS ON THE DRI PSO

9.1. Alternative implementation of the rotation matrices

The implementation of the DRI PSO, as discussed in Section 6, involves the construction of two rotation matrices, and then requires two matrix–vector multiplications per particle, per iteration. Hence the computational expense of the DRI PSO increases drastically in comparison to the classical PSO as the design dimension increases. However, we opted for this route since the

introduction of rotation matrices allowed for a simple proof of (stochastic) rotational invariance. In addition, it nicely illustrates the idea of perturbing the vector search directions.

A computationally economical alternative to the rotation matrices is to directly perturb the direction cosines of $(\mathbf{p}_k^i - \mathbf{x}_k^i)$ and $(\mathbf{p}_k^g - \mathbf{x}_k^i)$: Firstly, for particle i at iteration k , the unit vectors \mathbf{e}_{1k}^i and \mathbf{e}_{2k}^i of, respectively $(\mathbf{p}_k^i - \mathbf{x}_k^i)$ and $(\mathbf{p}_k^g - \mathbf{x}_k^i)$ are computed as

$$\begin{aligned}\mathbf{e}_{1k}^i &= \frac{\mathbf{p}_k^i - \mathbf{x}_k^i}{\|\mathbf{p}_k^i - \mathbf{x}_k^i\|} \\ \mathbf{e}_{2k}^i &= \frac{\mathbf{p}_k^g - \mathbf{x}_k^i}{\|\mathbf{p}_k^g - \mathbf{x}_k^i\|}\end{aligned}\quad (25)$$

where $\|\cdot\|$ indicates the Euclidean vector norm.

We then perturb each j th component of the two computed unit vectors in Equation (25) as follows:

$$\begin{aligned}\varepsilon_{1k}^i(j) &= \cos\left(\beta \frac{\pi}{180} \left(\frac{1}{2} - \varrho_{1k}^i(j)\right) + \arccos(e_{1k}^i(j))\right), \quad j = 1, \dots, n \\ \varepsilon_{2k}^i(j) &= \cos\left(\beta \frac{\pi}{180} \left(\frac{1}{2} - \varrho_{2k}^i(j)\right) + \arccos(e_{2k}^i(j))\right), \quad j = 1, \dots, n\end{aligned}\quad (26)$$

with $\varrho_{1k}^i(j)$ and $\varrho_{2k}^i(j)$ two uniform random scalars between 0 and 1. β defines the limits of the direction cosine perturbations.

The stochastic contribution v_k^i is then given by

$$v_k^i = c_1 r_{1k}^i \|\mathbf{p}_k^i - \mathbf{x}_k^i\| \mathbf{e}_{1k}^i + c_2 r_{2k}^i \|\mathbf{p}_k^g - \mathbf{x}_k^i\| \mathbf{e}_{2k}^i \quad (27)$$

To evaluate this, we constructing a grid of w (between 0 and 1 in increments of 0.1) and β (between 0 and 10, again in increments of 1) for both the rotated and unrotated test functions.

In Figures 16 and 17, we present contour plots for the best function values obtained after 10 000 iterations, averaged over 100 runs for the Rosenbrock f_0 and the Ackley f_2 test functions. Evidently, the methodology is invariant in a stochastic sense, for both examples. Incidentally: ‘good’ values for w and β are of course problem dependent, as observed for both f_0 and f_2 . Furthermore, for the problems studied, similar performance can be obtained by increasing w , accompanied by an appropriate decrease in β , and *vice versa*. For high values of w and β the algorithm is unstable (too much diversity); for low values of w and β the algorithm converges prematurely (the algorithm lacks diversity).

A summary of the best results obtained is presented in Table II, together with the corresponding w and β values. Again the problem dependency of the parameters is evident.

9.2. Alternatives to the DRI PSO

Finally, there are of course numerous methods to introduce directional diversity into the linear PSO, as opposed to our proposed option of independent directional perturbation. (Again, the linear PSO is taken as the departure point, since it is invariant.)

We mention only a single alternative here, namely an increase in the social awareness of the particles. In turn, this may for example be effected by increasing the number of particles

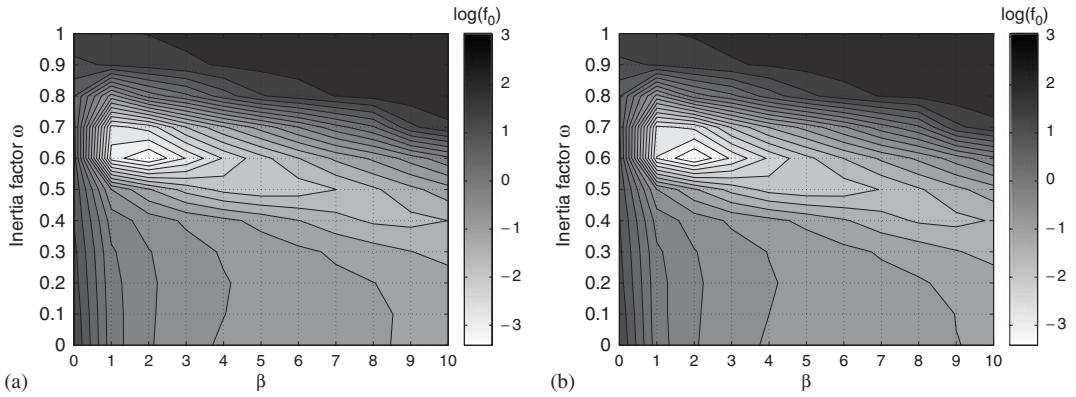


Figure 16. Contour plot of the average function value obtained with the direction cosine perturbation implementation of the DRI PSO after 10^4 iterations averaged over 100 runs. We present different values of β and ω for: (a) the unrotated; and (b) the rotated Rosenbrock test function f_0 .

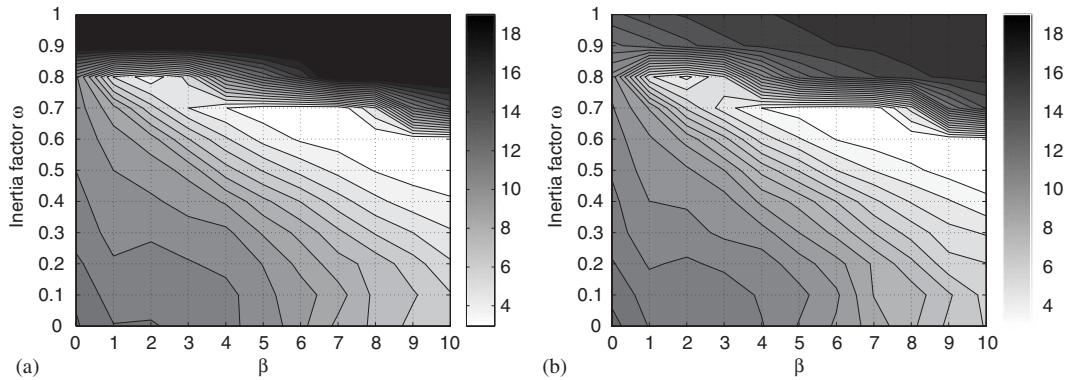


Figure 17. Contour plot of the average function value obtained with the direction cosine perturbation implementation of the DRI PSO after 10^4 iterations averaged over 100 runs. We present different values of β and ω for: (a) the unrotated; and (b) the rotated Ackley test function f_2 .

that contribute to Equation (8) [19, 20]. (One may of course achieve n -dimensional searches, if the number of particles $p \geq n$, unless the particle trajectories are parallel.) As an added benefit, additional information about the objective function is then used in the searches of any particle i .

10. NOTES ON HEURISTICS

In this study, we have considered a basic PSO algorithm, i.e. we have not implemented any of the popular heuristics like maximum velocity limit, craziness, minimum velocity and inertia adjustment. However, if reference frame invariance is desired in an optimization algorithm, the implemented heuristics introduced must also satisfy the requirements of reference frame invariance.

Table II. The best average cost function values $f_{\text{ave}}^{\text{best}}$ for the DRI velocity update rule with constant inertia factor w and direction cosine perturbation β . Results presented are for the unrotated $f_{\text{ave}}^{\text{best}}|_U$ and rotated $f_{\text{ave}}^{\text{best}}|_R$ test problems.

Function	w	β	$f_{\text{ave}}^{\text{best}} _U$	w	β	$f_{\text{ave}}^{\text{best}} _R$
f_0	0.6	2	4.30×10^{-4}	0.6	2	3.91×10^{-4}
f_1	0.7	2	5.28×10^{-21}	0.7	1	5.77×10^{-21}
f_2	0.6	9	2.74	0.6	9	2.71
f_3	0.8	1	64.7	0.8	1	65.1
f_4	0.6	2	1.25×10^{-2}	0.6	2	1.28×10^{-2}

We point out that certain heuristics are inherently reference frame *invariant* e.g. velocity re-initialization, maximum velocity restriction on the *magnitude* level, craziness on the particle level, etc. In turn, certain heuristics are inherently reference frame *variant*, e.g. an implementation of maximum velocity restriction on the velocity vector *components*, maximum position restriction on the position vector *components* and craziness on the *component* level.

When constructing a reference frame invariant PSO algorithm, the velocity update rule and the heuristics should satisfy the transformation rules discussed in Section 2, either exactly or in a stochastic sense.

11. CONCLUSION

We have investigated the *invariance* (or lack thereof) of the linear and the classical velocity update rules, expressed in terms of the scale, translation and rotation of an objective function.

We have presented proof of the scale and translational invariance of both the linear and the classical velocity update rules. In addition, we have presented proof that the linear velocity update rule is *invariant* of rotation, while the classical velocity update rule is rotationally *variant*.

Thereafter, we have proposed the DRI velocity update rule, which is reference frame *invariant* and *diverse*. In the DRI velocity update rule, we independently scale the magnitudes and perturb the directions of both the cognitive and social vectors $c_1(\mathbf{p}_k^i - \mathbf{x}_k^i)$ and $c_2(\mathbf{p}_k^g - \mathbf{x}_k^i)$. (This, however, comes at the cost of an additional scaling factor.) We have presented proof of scale, and translational invariance of the DRI velocity update rule, as well as rotational invariance in a stochastic sense.

In summary: the linear velocity update rule is rotational *invariant*, but demonstrates an overall poor performance, due to the particle trajectories collapsing to lines. This is a direct result of only scaling the magnitude of the cognitive and social vectors $c_1(\mathbf{p}_k^i - \mathbf{x}_k^i)$ and $c_2(\mathbf{p}_k^g - \mathbf{x}_k^i)$.

In turn, the classical velocity update rule lacks rotational *invariance*, which results in severe performance loss for ‘rotated’ functions. Nevertheless, the classical velocity update rule still outperforms the linear velocity update rule for both rotated and unrotated test functions, since the algorithm is diverse, i.e. the particle trajectories do not collapse to lines.

We then demonstrated that the DRI PSO may outperform the classical PSO for test functions with a single local minimum. In addition, the performance of the DRI PSO is comparable to the performance of the classical PSO for the multimodal test functions, with the added advantage of being invariant of the reference frame.

While we have not introduced heuristics, we have pointed out that few heuristics which introduce directional diversity are invariant of reference frame. We note that when formulating an algorithm which is invariant of scale and reference frame, only heuristics that reflects this invariance should be incorporated into such an algorithm. Doubtless, such formulations will be proposed in future.

ACKNOWLEDGEMENTS

The financial assistance of the National Research Foundation (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the authors and are not necessarily to be attributed to the NRF.

REFERENCES

1. Wilke DN, Kok S, Groenwold AA. Comparison of linear and classical velocity update rules in particle swarm optimization: notes on diversity. *International Journal for Numerical Methods in Engineering*, in this issue.
2. Paquet U, Engelbrecht AP. A new particle swarm optimiser for linearly constrained optimisation. *IEEE Congress on Evolutionary Computation* 2003; **1**:227–233.
3. Kennedy J, Eberhart RC. Particle swarm optimization. *IEEE Conference on Neural Networks* 1995; **4**:1942–1948.
4. Eberhart RC, Kennedy J. A new optimizer using particle swarm theory. *Sixth International Symposium on Micro Machine and Human Science* 1995; 39–43.
5. Holzapfel GA. *Nonlinear Solid Mechanics. A Continuum Approach for Engineering*. Wiley: Chichester, 2001.
6. Snyman JA. *Practical Mathematical Optimization. An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms*. Springer: New York, 2005.
7. Salomon R. Re-evaluating genetic algorithm performance under coordinate rotatation of benchmark functions. *BioSystems* 1996; **39**(3):263–278.
8. Salomon R. Some comments on evolutionary algorithm theory. *Evolutionary Computation Journal* 1996; **4**(4): 405–415.
9. Spiegel MR. *Theory and Problems of Vector Analysis*. Schaum Publishing Co.: New York, U.S.A., 1959; 57–59.
10. Shi Y, Eberhart RC. A modified particle swarm optimizer. *IEEE Conference on Evolutionary Computation* 1998; 69–73.
11. Shi Y, Eberhart RC. Parameter selection in particle swarm optimization. *Evolutionary Programming VII* 1998; 591–600.
12. Eberhart RC, Shi Y. Comparing inertia weights and constriction factors in particle swarm optimization. *IEEE Congress on Evolutionary Computation* 2000; **1**:84–88.
13. Metropolis N, Ulam S. The Monte Carlo method. *Journal of the American Statistical Association* 1949; **44**(247):335–341.
14. Secrest BR, Lamont GB. Visualizing particle swarm optimization—Gaussian particle swarm optimization. *IEEE Swarm Intelligence Symposium* 2003; 198–204.
15. Papoulis A. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill/Kogakusha, Ltd.: New York, Tokyo, 1965; 187–199.
16. Schutte JF, Koh BI, Reinbolt JA, Haftka RT, George AD, Fregly BJ. Scale-independent biomechanical optimization. *2003 Summer Bioengineering Conference*, Key Biscayne, FL, U.S.A., June 2003; 55–56.
17. Moler C, Van Loan C. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review* 2003; **45**:3–49.
18. Carlisle A, Dozier G. An off-the-shelf PSO. *Workshop on Particle Swarm Optimization*, Purdue School of Engineering and Technology, IN, U.S.A., 2001.
19. Mendes R, Kennedy J, Neves J. The fully informed particle swarm: simpler, maybe better. *IEEE Transactions on Evolutionary Computation* 2004; **8**:204–210.
20. Baskar S, Suganthan PN. A novel concurrent particle swarm optimization. *IEEE Congress on Evolutionary Computation* 2004; **1**:792–796.