Degrees of Stochasticity in Particle Swarm Optimization

Oldewage, E.T. · Engelbrecht, A.P. · Cleghorn, C.W.

Abstract This paper illustrates the importance of independent, component-wise stochastic scaling values, from both a theoretical and empirical perspective. It is shown that a swarm employing scalar stochasticity in the particle update equation is unable to express every point in the search space if the problem dimensionality is sufficiently large in comparison to the swarm size. The theoretical result is emphasized by an empirical experiment which shows that a swarm using scalar stochasticity performs significantly worse when the optimum is not in the span of its initial positions. It is also demonstrated that even when the problem dimensionality allows a scalar swarm to reach the optimum, a swarm with component-wise stochasticity significantly outperforms the scalar swarm. This result is extended by considering different degrees of stochasticity, in which groups of components share the same stochastic scalar. It is demonstrated on a large range of benchmark functions that swarms with dimensional coupling (including scalar swarms in the most extreme case) perform significantly worse than a swarm with component-wise stochasticity. The paper also shows that, contrary to previous results in the field, a swarm with component-wise stochasticity is not biased towards the subspace within which it is initialized. The misconception is shown to have arisen in previous literature due to overzealous normalization when measuring swarm movement, which is corrected in this paper.

Oldewage, E.T.
E-mail: etv21@cam.ac.uk
Department of Engineering, University of Cambridge, Cambridge, UK

Engelbrecht, A.P.
E-mail: engel@sun.ac.za
Department of Industrial Engineering, Stellenbosch University, Stellenbosch, South Africa
Division of Computer Science, Stellenbosch University, Stellenbosch, South Africa

Cleghorn, C.W.
E-mail: ccleghorn@cs.up.ac.za
Department of Computer Science, University of Pretoria, Pretoria, South Africa

Keywords Particle swarm optimization · Stochastic scaling · Component-wise scaling · Dimensional coupling

1 Introduction

The particle swarm optimization (PSO) algorithm employs stochasticity as an important mechanism to avoid premature convergence to local optima. The stochasticity should (usually) be applied in every dimension (i.e. component-wise) to ensure independence between position updates in each dimension. However, it is still a prevalent mistake for scalar stochastic values to be used instead (Feng et al. 2019; Fu et al. 2018; Han and Liu 2014; Li et al. 2019; Pandey et al. 2010; Sun et al. 2019; Yoshida et al. 2000; Zahara et al. 2009), which unnecessarily restricts the swarm's movement and degrades performance.

This paper investigates the effect of using scalar stochasticity, both theoretically and empirically. The paper begins by introducing the PSO algorithm and briefly discussing the importance of component-wise stochasticity in Section 2. Section 3 provides theoretical results to formalize the restriction on the swarm's movement caused by scalar stochasticity. It is shown that there is a problem of "reachability", i.e. a swarm with scalar stochasticity will not be able to reach the optimum if the problem dimensionality is higher than the size of the swarm.

Section 4 examines the empirical effects of reachability by comparing the performance of a swarm with scalar stochasticity to a swarm with component-wise stochasticity for a wide range of benchmark functions. The section analyzes the behaviour of the swarms using a measure from existing literature, which compares the magnitude of particle steps inside the subspace within which they are initialized, to step sizes in the orthogonal subspace. A correction to the normalization of the measure is proposed and using the modified measure, it is shown empirically that a vector swarm searches inside and outside the initial subspace equally, without bias towards either space.

Section 5 considers an intermediate approach, between scalar and component-wise stochasticity, where coupling is induced among groups of dimensions by using the same stochastic scaling value within a group. The behaviour of swarms with induced coupling is discussed and their performance is compared to vector and scalar swarms. Section 6 concludes the paper.

This paper is an extension of the work by Oldewage et al. (2018). The paper includes additional results regarding the roaming behaviour of scalar and vector swarms. The extended paper supplements the discussion of component-wise and scalar stochasticity by considering an intermediate degree of stochasticity, where stochastic scalars are shared among groups of dimensions. The extension also introduces an important correction to the measure of particle steps within a given subspace from existing literature. The corrected measure leads to results that are entirely different from the observations by Chen et al. (2015) regarding the exploration ability of swarms with component-wise scaling. The measure is applicable to other population-based algorithms, so the correction

is of interest to the general field of population-based optimization. The extension also includes an appendix with additional proofs for the propositions presented in Section 3.

2 Background

This section briefly discusses the PSO algorithm and introduces relevant concepts regarding the importance of component-wise stochasticity.

PSO is a stochastic, population-based optimization algorithm that does not require gradient information and may thus be applied to black box optimization problems (Eberhart and Kennedy 1995). A swarm consists of a number of particles. The position of a particle in the search space represents a potential solution to the optimization problem. The particle moves through the search space for a number of iterations, using local information (the best position encountered by the particle thus far, called the personal best position) and global information (the best position encountered by all the particles within the given particle's logical neighbourhood, called the global or local best position, depending on how the neighbourhood is defined). This paper considers the global best topology (Eberhart and Kennedy 1995), but the findings presented are applicable to arbitrary topologies. Each particle i's position is updated at iteration t according to:

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1} \tag{1}$$

where $\mathbf{x}_i^{t+1}$ denotes the position of particle i at iteration $t+1$ and $\mathbf{v}_i^{t+1}$ denotes its velocity at iteration $t+1$. The particle's initial position, $\mathbf{x}_i^0$, is usually drawn from a uniform distribution over the search space boundaries (in every dimension). PSO with inertia weight, as introduced by Shi and Eberhart (1998), updates particle i's velocity at iteration t in every dimension j as below:

$$v_{ij}^{t+1} = wv_{ij}^t + c_1 r_{1j}(y_{ij}^t - x_{ij}^t) + c_2 r_{2j}(\hat{y}_{ij} - x_{ij}^t) \tag{2}$$

where v_{ij}^{t+1} denotes particle i's velocity in dimension j, w denotes the inertia weight, and c_1 and c_2 denote the cognitive and social acceleration constants respectively. The components r_{1j}, $r_{2j} \sim U(0,1)$ are random numbers sampled between 0 and 1 at every iteration. y_{ij}^t denotes the personal best position of particle i in the jth dimension and $\hat{y}_{ij}$ denotes the jth dimension of the best position found by all the particles in the neighbourhood of particle i. Particle neighbourhoods are usually defined by logical indexing of the swarm. When neighbourhoods are strict subsets of the entire swarm, the algorithm is referred to as a local best PSO. If every particle's neighbourhood consists of the entire swarm, the algorithm is referred to as a global best PSO and $\hat{\mathbf{y}} = \hat{\mathbf{y}}_i$ is called the global best position. Except where otherwise specified, this paper considers a global best PSO.

The stochastic scaling components, $\mathbf{r}_1$ and $\mathbf{r}_2$, can also be expressed as:

$$\mathbf{v}_i^{t+1} = w\mathbf{v}_i^t + c_1\mathbf{R}_1(\mathbf{y}_i^t - x_{ij}^t) + c_2\mathbf{R}_2(\hat{\mathbf{y}} - \mathbf{x}_i^t) \tag{3}$$

where $\mathbf{R}_1$ and $\mathbf{R}_2$ are diagonal matrices, with $\mathbf{r}_1$ and $\mathbf{r}_2$ forming their diagonals.

This paper emphasizes that if $\mathbf{r}_1$ and $\mathbf{r}_2$ are instead random scalars, then the swarm's movement becomes entirely linear. The limitation of linear movement is illustrated by Parsopoulos and Vrahatis (2010, p. 30) which plots a particle's potential future positions when using scalar or vector stochastic scaling. Every position investigated by the swarm will necessarily be a linear combination of the swarm's initial positions, velocities, and personal best positions. If the swarm size is too small relative to the problem dimensionality, then the swarm can only reach a subspace within the larger search space (as is proved in Section 3). Since the swarm is typically initialised randomly, there is a possibility that the optimum can not be expressed as a linear combination of the swarm's initial positions, i.e. the swarm will never be able to find the optimum.

There are variants of PSO from literature that use different sampling distributions for $\mathbf{r}_1$ and $\mathbf{r}_2$, such as double-exponential (Krohling and dos Santos Coelho 2006b), normal (Krohling and dos Santos Coelho 2006a), and normal-absolute (Krohling and dos Santos Coelho 2006a). The use of different distributions affects the dynamics of the swarm and induce different convergence regions for the swarm's parameters (Garca-Gonzalo and Fernndez-Martnez 2014).

The majority of literature considering the movement of PSO assumes that the particles move independently in each dimension, thereby requiring that $\mathbf{r}_1$ and $\mathbf{r}_2$ be vectors of independent random numbers. Apart from being mentioned as a common mistake in some textbooks (Parsopoulos and Vrahatis 2010, p. 30), there is not much literature which explicitly discusses the effect of using scalar random values, or inducing coupling between dimensions, on the space of solutions that can be reached by the swarm.

Paquet and Engelbrecht (2007) introduced a Linear PSO which uses scalar r_1 and r_2 values in order to solve constrained linear optimization problems. If the swarm is initialised so that all positions are within the problem constraints, then scalar r_1 and r_2 values ensures that the swarm can never leave the feasible space, which forms a subspace of the search space. In (van Zyl and Engelbrecht 2016), purposeful dimensional coupling via shared $\mathbf{r}_1$ and $\mathbf{r}_2$ components was suggested to reduce the unwanted roaming exhibited by PSO in high dimensional problem spaces.

Throughout the remainder of the paper, a swarm that uses scalar values for r_1 and r_2 will be called a "scalar swarm". A swarm that uses vectors for $\mathbf{r}_1$ and $\mathbf{r}_2$ which are multiplied with the cognitive and social components in every dimension (or, alternatively, are diagonal matrices) will be called a "vector swarm".

3 Theoretical Results

This section provides a theoretical discussion regarding the consequences of using a scalar PSO. It is proven that if r_1 and r_2 are scalars, then the positions of any particle at any iteration of the search space must be a linear combination of their initial positions, personal best positions, and velocities. The proposition below is for a local best PSO, since global best PSO can be considered as a special case of local best PSO where the neighbourhood is the entire swarm. For the sake of generality, the proposition makes no assumptions regarding the initial particle velocities or personal best positions.

Proposition 1 *For a particle swarm governed by the movement update equations in equations (1) and (2), at any iteration $t \geq 0$, the position $\mathbf{x}_i^t$ of any particle i is in the span of $\mathcal{I}$ where $\mathcal{I} = \{\mathbf{x}_1^0, \mathbf{y}_1^0, \mathbf{v}_1^0, ..., \mathbf{x}_m^0, \mathbf{y}_m^0, \mathbf{v}_m^0\}$.*

Proof Suppose that particle velocities, positions and personal best positions are initialised randomly within the search space. Let the set of all these initial points be given by $\mathcal{I} = \{\mathbf{x}_1^0, \mathbf{y}_1^0, \mathbf{v}_1^0, ..., \mathbf{x}_{n_s}^0, \mathbf{y}_{n_s}^0, \mathbf{v}_{n_s}^0\}$. Assume that all the elements in $\mathcal{I}$ are unique and non-zero. These assumptions are made without loss of generality: the probability of obtaining a zero vector from a uniform initialisation is zero, since the probability of a continuous random variable being a particular constant is zero (due to a single point having zero measure). Similarly, the probability of sampling two equal vectors is zero because the set of such points have zero measure. The proof proceeds by induction.

The position of any particle at $t = 0$ is in $\mathcal{I}$ by the definition of $\mathcal{I}$. Thus, the hypothesis holds for the case $t = 0$.

Suppose for all iterations $s \leq t$ that the positions of all the particles are in the span of $\mathcal{I}$. It will now be proved that the positions of all the particles must still be in the span of $\mathcal{I}$ at iteration $t + 1$. The position of any particle i is given by the position update equation:

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1} \tag{4}$$

where $\mathbf{x}_i^t \in \mathit{span}(\mathcal{I})$ by virtue of the inductive assumption. It thus remains to prove that $\mathbf{v}_i^{t+1}$ is in the span of $\mathcal{I}$:

$$\mathbf{v}_i^{t+1} = w\mathbf{v}_i^t + c_1 r_1(\mathbf{y}_i^t - \mathbf{x}_i^t) + c_2 r_2(\hat{\mathbf{y}}_i^t - \mathbf{x}_i^t) \tag{5}$$

$$= w\mathbf{v}_i^t + c_1 r_1 \mathbf{y}_i^t + c_2 r_2 \hat{\mathbf{y}}_i^t - (r_1 c_1 + r_2 c_2)\mathbf{x}_i^t \tag{6}$$

where $\mathbf{x}_i^t \in \mathit{span}(\mathcal{I})$ by the inductive assumption. It remains to prove that $\mathbf{v}_i^t$, $\mathbf{y}_i^t$ and $\hat{\mathbf{y}}_i^t$ are in the span of $\mathcal{I}$. According to the position update equation,

$$\mathbf{x}_i^t = \mathbf{x}_i^{t-1} + \mathbf{v}_i^t \tag{7}$$

$$\implies \mathbf{v}_i^t = \mathbf{x}_i^t - \mathbf{x}_i^{t-1} \tag{8}$$

In other words, $\mathbf{v}_i^t$ is a linear combination of $\mathbf{x}_i^t$ and $\mathbf{x}_i^{t-1}$, both of which are elements in the span of $\mathcal{I}$ by the inductive assumption. Thus, $\mathbf{v}_i^t$ is also in the span of $\mathcal{I}$. The personal best position of any particle i can only be equal to one

of the particle's previous positions. Similarly, the neighbourhood best position must be equal to a previous position of some particle in i's neighbourhood. But the previous positions of all particles are in the span of $\mathcal{I}$ by the inductive assumption. Thus, $\mathbf{y}_i^t$ and $\hat{\mathbf{y}}_i^t$ are in the span of $\mathcal{I}$, as required. The velocity and thus also the position of particle i at iteration $t+1$ is in $span(\mathcal{I})$. □

Proposition 1 implies that, when r_1 and r_2 are random scalar values, the positions of the particles are limited to be in the span of their initial velocities, positions and personal best positions. If either of the assumptions on $\mathcal{I}$ does not hold (e.g. some vectors are multiples of another or some are zero), then the positions of the particles are limited further to being linear combinations of all non-zero, linearly independent initial velocities, positions and personal best positions. The question arises whether any point in the search space can be expressed in terms of such linear combinations. This question is answered by the Proposition below:

Proposition 2 *Suppose $\mathcal{I}$ contains m linearly independent vectors. Let the search space, S, be an n-dimensional hypercube with lower bound L and upper bound U in all dimensions, i.e. $S = [L, U]^n$. If $m < n$ then $span(\mathcal{I}) \cap S \subsetneq S$. Thus $\mathcal{I}$ can only be a spanning set of S if it contains at least n linearly independent elements.*

Proposition 2 follows from the fundamental theorem of invertible matrices as given in (Poole 2011), but not directly since S is not a subspace of $\mathbb{R}^n$. For the sake of completeness, the proof is given in the appendix.

If $\mathcal{I}$ constitutes a spanning set for the search space (i.e. the span of $\mathcal{I}$ is larger than the search space or equal to), then any point in the search space can theoretically be reached by the particles. However, if $\mathcal{I}$ is not a spanning set of the search space (i.e. the span of $\mathcal{I}$ is a strict subspace within the search space), then the particles can not reach every position in the search space. If the global optimum happens to be outside the span of $\mathcal{I}$, then the particles will never be able to find it. Since initial positions and personal best positions are typically generated randomly, there is no guarantee that the optimum for a given problem can be reached by the swarm (when $|\mathcal{I}| < n$). Real world problems in high dimensional spaces are thus likely to have optima that are not reachable by a swarm.

If the swarm's velocities are initialised to zero and the initial personal best positions are set equal to the initial positions, then the portion of the search space that can be reached by the particles is even smaller. Additionally, the swarm may lose degrees of freedom throughout the search. Since the algorithm is executed on a computer with limited precision, some of the vectors in $\mathcal{I}$ may be cancelled out further in the search. Though unlikely, the span of the swarm may in fact decrease as the search progresses. Thus, if the size of the swarm is much smaller than the dimensionality of the search space, then the swarm will be unable to reach a large part of the search space. Unfortunately, simply increasing the number of particles in the swarm is not an adequate solution, because it greatly increases the computational cost. Additionally, the swarm

size parameter influences the swarm's searching behaviour in other ways, so changing the swarm size drastically may have unintended consequences (Engelbrecht 2014; Malan and Engelbrecht 2008).

4 Performance Comparison

This section compares the performance of the vector and scalar swarms on a large suite of unbiased benchmark functions. The optima for these functions are either shifted by a predefined constant (as specified in the corresponding technical papers) or by a random vector, distributed uniformly over the search space. Section 4.1 details the empirical method and Section 4.2 discusses the results.

4.1 Empirical Method - Performance Comparison

The benchmark suite consisted of 28 base functions which are listed in Table 1. A given function f was shifted and rotated to produce f_{ShRot} according to

$$f(\mathbf{x})_{ShRot} = f(Q(\mathbf{x} - \boldsymbol{\gamma})) + \beta \tag{9}$$

where β is a constant scalar, $\boldsymbol{\gamma}$ is either constant or uniform random over the search space in each dimension, and Q is a randomly generated orthogonal matrix. The constants are specified in Table 1.

The "Rot" column in Table 1 indicates whether the function was rotated or not. The transformations provided a total of 46 benchmark functions. The benchmark suite contains uni- and multi-modal functions that are both separable and non-separable. The definitions of the functions and the corresponding bounds were used as in (Engelbrecht 2013), (Jamil and Yang 2013) and (Ramezani and Lotfi 2012). The "Src" column of Table 1 lists the identifier of each function according to its source. Function i from Engelbrecht (2013) is denoted by f_i; function i from Jamil and Yang (2013) is denoted by F_i and function i from Ramezani and Lotfi (2012) is denoted by G_i. The vector and scalar swarms were run on each of the benchmark problems 30 times for statistical significance. Each simulation ran for 2000 iterations.

The experiments used PSO with inertia weight as introduced by Shi and Eberhart (1998) with the global best topology. The selected inertia weight, $w = 0.7298$ and the acceleration coefficients $c_1 = c_2 = 1.49618$, are known good values suggested by Clerc and Kennedy (2002) that guarantee convergence of the swarm (in terms of expectation and variance of particle positions (Cleghorn and Engelbrecht 2018)). As suggested by Bratton and Kennedy (2007), all personal and global best positions were restricted to be within the search space. Each swarm consisted of 10 particles ($m = 10$), so that m is low enough to test problems of dimensionality 5 times larger than m without venturing into large scale optimization). The particles' initial positions were initialised uniform randomly throughout the search space. Each particle's initial personal

Table 1 Benchmark Functions

Function Name	**Src**	γ	β	**Rot**	**Function Name**	**Src**	γ	β	**Rot**
Absolute Value	f_1	Rand	0.0	No	Rastrigin	f_{12}	Rand	0.0	No
Ackley	f_2	Rand	0.0	No	Rastrigin Rot	f_{12}	2.0	-330	No
Ackley Sh	f_2	10.0	-140	No	Rastrigin Sh	f_{12}	0.0	0.0	Yes
Ackley Rot	f_2	0.0	0.0	Yes	Rastrigin ShRot	f_{12}	1.0	-330	Yes
Ackley ShRot	f_2	-32.0	-140	No	Rosenbrock	f_{13}	Rand	0.0	No
Alpine	F_7	Rand	0.0	No	Rosenbrock Sh	f_{13}	10.0	390	No
Brown	F_{25}	Rand	0.0	No	Rosenbrock Rot	f_{13}	0.0	0.0	Yes
Dixon-Price	F_{48}	Rand	0.0	No	Salomon	f_{14}	Rand	0.0	No
Egg Holder	f_4	Rand	0.0	No	Schaffer 6	f_{15}	Rand	0.0	No
Elliptic	f_5	Rand	0.0	No	Schaffer 6 ShRot	f_{15}	20.0	-300	Yes
Elliptic Sh	f_5	10.0	-450	No	Schwefel	G_5	0.0	0.0	No
Elliptic Rot	f_5	0.0	0.0	Yes	Schwefel 1.2	f_{16}	Rand	0.0	No
Elliptic ShRot	f_5	10.0	-450	Yes	Schwefel 1.2 Sh	f_{16}	10.0	-450	No
Griewank	f_6	Rand	0.0	No	Schwefel 1.2 Rot	f_{16}	0.0	0.0	Yes
Griewank Sh	f_6	10.0	-180	No	Schwefel 2.21	f_{19}	Rand	0.0	No
Griewank Rot	f_6	0.0	0.0	Yes	Schwefel 2.22	f_{20}	Rand	0.0	No
Griewank ShRot	f_6	-60.0	-180	Yes	Shubert	f_{21}	Rand	0.0	No
HyperEllipsoid	f_7	Rand	0.0	No	Spherical	f_{22}	Rand	0.0	No
Michalewicz	f_8	Rand	0.0	No	Spherical Sh	f_{22}	10.0	-450	No
Norwegian	f_9	Rand	0.0	No	Step	f_{23}	Rand	0.0	No
Powell Singular 2	F_{92}	Rand	0.0	No	Vincent	f_{24}	Rand	0.0	No
Quadric	f_{10}	Rand	0.0	No	Weierstrauss	f_{25}	Rand	0.0	No
Quartic	f_{11}	Rand	0.0	No	Weierstrauss Sh	f_{25}	1.0	-130	No

best position was set equal to its initial position, and its velocity was initialised to zero. Thus, the scalar swarm was limited to the span of its initial positions. All of the functions were minimized in 5, 10, 15, 20, and 25 dimensions. If the hypothesis proved in the previous section holds, then it is expected for the scalar swarm's performance to deteriorate as the problem dimensionality exceeds the number of particles in the swarm.

4.2 Results - Performance Comparison

Table 2 summarizes the results of the performance comparison. Every row of the table corresponds to the results for a given problem dimensionality. Friedman tests with a p-value of 0.05 and a sample size of 30 were used to detect statistically significant differences between the scalar and vector PSO's performance (in terms of the best fitness attained by the swarm on a given function). If the Friedman test indicated a significant difference, pairwise comparisons were done by Mann-Whitney U tests with a p-value of 0.05. If no statistically significant difference was found, the result was recorded as a draw. The scalar swarm consistently performed better than the vector swarm on the Quadric

Table 2 Comparison of Vector and Scalar Swarms Across Dimensionality

Dimensionality	Scalar Wins	Draws	Vector Wins
$n = 5$	0	2	44
$n = 10$	1	0	45
$n = 15$	1	0	45
$n = 20$	1	2	43
$n = 25$	1	1	44

function (f_{10}) for $n > 5$. However, the vector swarm outperformed the scalar swarm on nearly all of the benchmark functions, even when the problem dimensionality is low enough for the scalar swarm to reach the optimum. The scalar swarm's linear movement thus prevents the swarm from finding good solutions even inside the swarm's initial subspace.

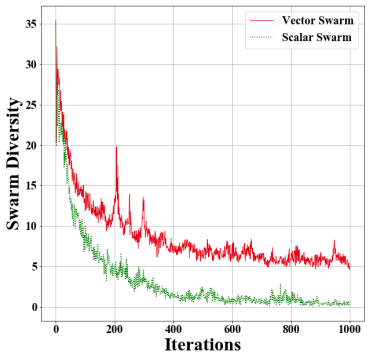

Fig. 1 Diversity, Ackley ShRot ($n = 5$)

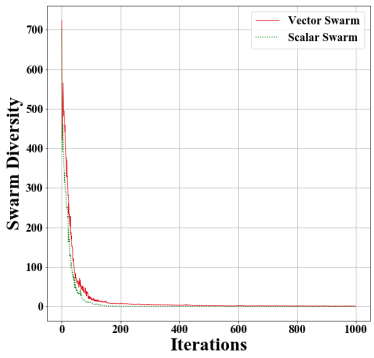

Fig. 2 Diversity, Griewank ($n = 5$)

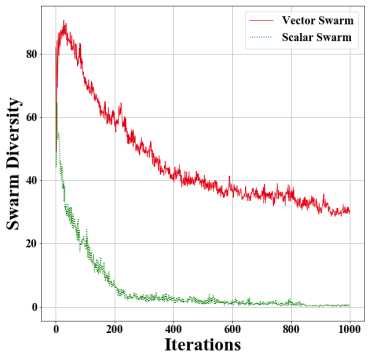

Fig. 3 Diversity, Ackley ShRot ($n = 25$)

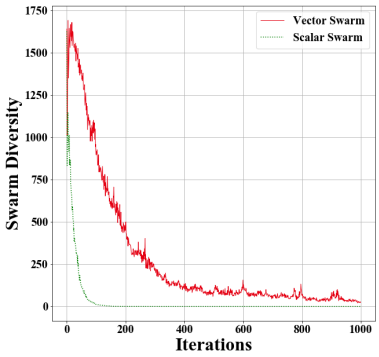

Fig. 4 Diversity, Griewank ($n = 25$)

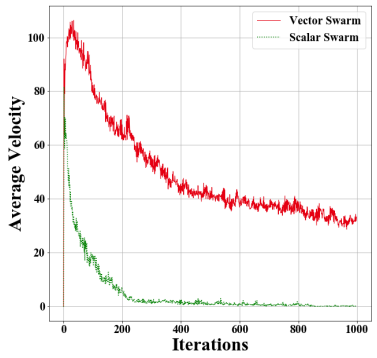

Fig. 5 Average Swarm Velocity, Ackley ShRot ($n = 25$)

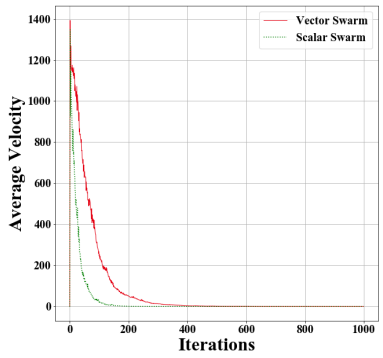

Fig. 6 Average Swarm Velocity, Griewank ($n = 25$)

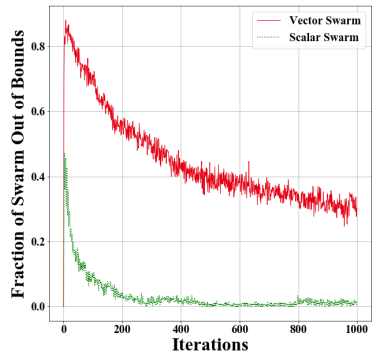

Fig. 7 Fraction of Swarm Out of Bounds, Ackley ShRot ($n = 25$)

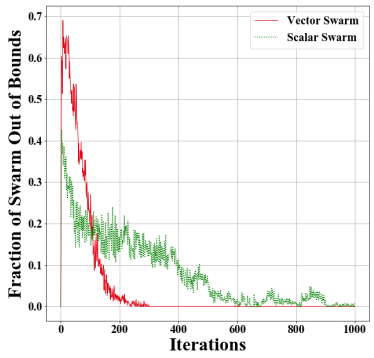

Fig. 8 Fraction of Swarm Out of Bounds, Griewank ($n = 25$)

The strong restriction imposed on the scalar swarm becomes apparent in Figures 1 to 4, which plot typical profiles of the swarm diversity, averaged over all runs for $n = 5$ and $n = 25$. The swarm diversity (Olorunda and Engelbrecht 2008) is given by

$$\mathcal{D} = \frac{1}{m} \sum_{i=1}^{m} \sqrt{\sum_{j=1}^{n} (x_{ij} - \bar{x}_j)^2} \tag{10}$$

where $\bar{\mathbf{x}}$ denotes the swarm centre, given as

$$\bar{\mathbf{x}} = \frac{1}{m} \sum_{i=1}^{m} \mathbf{x}_i \tag{11}$$

As the problem dimensionality increases, the vector PSO's diversity also increases. In contrast, the scalar PSO's diversity profile remains unchanged even as the dimensionality increases. Figures 5 and 6 illustrate that the velocity of scalar swarms (averaged across particles and problem dimensions) is generally lower than the vector swarms, especially during the first part of the search.

Figures 7 and 8 show the fraction of the swarm out of bounds throughout the search. Most benchmark functions exhibited profiles similar to Figure 7, where the vector swarm roams considerably more than the scalar swarm throughout the search. The decreased roaming by the scalar swarm may be due to its strong movement restrictions and lower overall velocities.

For a few benchmark functions, both the vector and scalar swarms returned to the search space within the first few hundred iterations (with the scalar swarm returning faster than the vector swarm) and remained inside the feasible space.

Some of the benchmark functions exhibited roaming similar to Figure 8, where the vector swarm unexpectedly returned to the search space faster than the scalar swarm. In these cases the scalar swarm typically exhibited very low diversity which decreased rapidly within the first hundred iterations.

For most of these cases, the optima were shifted to random points in the search space. Thus, it may be that the scalar swarms were initialized in a subspace that could not reach the optimum. The nearest point to the optimum within reachable space may have been near the boundary of the search space, causing most of the swarm to be near the boundary (resulting in the observed low diversities), but with frequent roaming as the swarm attempted to approach the optimum and failed. The vector swarms in contrast, would be able to find a region near the optimum within the search space.

As shown before (Oldewage et al. 2017; van Zyl and Engelbrecht 2016), restricting the swarm's movement may be beneficial in high dimensional spaces for the very reason that the initial velocity explosion is mitigated, causing the unchanged diversity profile observed here. However, in low dimensional spaces, the vector swarm outperformed the scalar swarm.

4.3 Empirical Exploration

The degree to which the vector swarm explores the region of the search space outside its initial subspace can be quantified by measuring the size of its steps inside the initial subspace, compared to steps in the orthogonal subspace. The measure was introduced by Chen et al. (2015), but this paper proposes a modification which leads to a more accurate reflection of swarm exploration.

As proposed in Chen et al. (2015), the magnitude of a particle's step within the initial subspace can be measured by

$$S_I = \frac{\|P\mathbf{v}_i^t\|_2}{|\mathcal{I}|} \tag{12}$$

where P is a projection matrix onto the initial subspace and $\|.\|_2$ denotes the Euclidean norm. P is given by

$$P = AA^T \tag{13}$$

where A is an n-by-$|\mathcal{I}|$ matrix, whose columns form an orthonormal basis for the initial subspace. The orthonormal basis may be obtained by applying the Gram-Schmidt method to $\mathcal{I}$, the set of non-zero, independent initial particle positions, velocities, and personal best positions. Note that A^T denotes the transpose of the matrix A.

A particle's step outside the initial subspace (in the orthogonal space) is given by

$$S_O = \frac{\|P'\mathbf{v}_i^t\|_2}{n - |\mathcal{I}|} \tag{14}$$

where P' is a projection matrix onto the space orthogonal to the initial subspace. P' may be obtained by finding an orthonormal basis for the initial subspace's orthogonal complement and applying equation (13). The step size within a given subspace is normalized by the number of basis vectors for that subspace ($|\mathcal{I}|$ for the initial space and $n - |\mathcal{I}|$ for the space orthogonal to the initial space). This is intended to allow direct comparison between S_I and S_O to determine how much movement took place in dimensions inside and outside the initial subspace.

Based on these measures, Figure 9 shows that the vector swarm focuses more on the initial subspace than the orthogonal subspace. Similar observations were made by Chen et al. (2015). However, bias towards the initial subspace does not agree with theory: since stochastic scaling takes place component-wise, the particles should behave independently in each dimension.

This paper argues that the observations by Chen et al. (2015), as illustrated in Figure 9, are not due to actual bias in the swarm, but rather due to incorrect normalization in equations (12) and (14).

A particle's step size in the subspace orthogonal to the initial space is measured by projecting its velocity into the orthogonal subspace and taking the Euclidean norm. Let $P'\mathbf{v}_i^t = \mathbf{z}'$. Then the Euclidean measure of $\mathbf{z}'$ is given by

$$\|\mathbf{z}'\|_2 = \sqrt{z_1^2 + z_2^2 + ... + z_n^2} \tag{15}$$

The projected velocity, $\mathbf{z}$, will be zero in the dimensions of the initial subspace and non-zero in all other dimensions. If the swarm has M particles and $\mathbf{v}_i^0 = \mathbf{0}$ and $\mathbf{y}_i^0 = \mathbf{x}_i^0$ for each particle i, then the initial subspace is M-dimensional. The projected velocity will thus be non-zero in all but M dimensions.

If the problem dimensionality is increased, then $\|\mathbf{z}'\|_2$ grows proportionally to $\sqrt{n - M}$, since $\|\mathbf{z}'\|_2$ will have an additional, non-zero term inside the square-root of Equation (15) (assuming the swarm size remains the same).

The normalization constants should thus be proportional to the **square root** of the number of dimensions in the subspaces instead, since distance (i.e.

step size) grows proportionally to $\sqrt{k}$ as k increases, where k is the number of dimensions in the space.

Scaling the projected step by k unfairly penalizes the subspace with more basis vectors, creating the illusion that more movement is taking place in the smaller subspace. In this case, the space orthogonal to the initial subspace is much larger than the initial subspace. Scaling $\|P'\mathbf{v}_i^t\|$ by $n - |\mathcal{I}|$ would thus create the impression that movement in the space orthogonal to the initial subspace is much less than it actually is, because the step magnitudes were scaled by a number that is too large. The modified measures are thus given by

$$S_I = \frac{\|P\mathbf{v}_i^t\|}{\sqrt{|\mathcal{I}|}} \tag{16}$$

$$S_O = \frac{\|P'\mathbf{v}_i^t\|}{\sqrt{n - |\mathcal{I}|}} \tag{17}$$

Figure 10 shows that, when the corrected measure is applied, the vector swarm's exploration is not biased towards the subspace within which it is initialized. The swarm focuses almost equally on the initial subspace and the subspace orthogonal to the initial subspace. Throughout the remainder of the paper, step magnitudes in a given space will be scaled proportionally to the square root of the number of basis vectors for the space.

Figure 11 shows the magnitude of steps inside and outside the initial subspace for both the vector and scalar swarms. As expected, the scalar swarm exhibits zero exploration outside the initial subspace. The scalar swarm's steps in the initial subspace are smaller than the vector swarm's steps, indicating that the scalar swarm is not only restricted to the subspace within which it is initialized, but also explores less within that subspace than the vector swarm.

5 Inducing Coupling - An Intermediate Approach

So far, the paper has considered two extremes: $\mathbf{r}_1$ and $\mathbf{r}_2$ were either scalars or n-dimensional vectors. The behaviour of the swarm was thus either entirely linear, or independent in each dimension. This section considers the implications of inducing coupling between dimensions by applying the same stochastic scalar to groups of dimensions. Section 5.1 explains how the coupling is induced and the experimental setup is given. Section 5.2 discusses the swarms' behaviour based on the experimental results, specifically in comparison with scalar and vector swarms.

5.1 Grouping Dimensions

Problem dimensions were grouped into g-many groups. Let $\mathbf{r}_1$ and $\mathbf{r}_2$ be g-dimensional vectors of uniform random numbers between 0 and 1. The modified velocity update equation is given by:

$$v_{ij}^{t+1} = wv_{ij}^t + c_1 r_{1k}(y_{ij}^t - x_{ij}^t) + c_2 r_{2k}(\hat{y}_{ij} - x_{ij}^t) \tag{18}$$

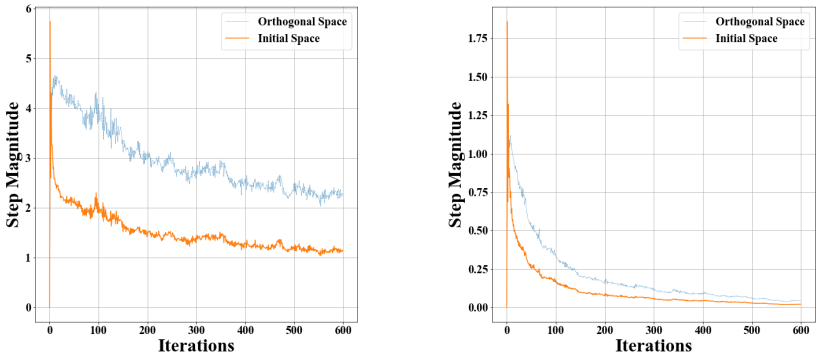

Fig. 9 Normalized by $\frac{1}{n}$, as in equations (12) and (14): Magnitude of steps in the initial subspace compared to the orthogonal space for a vector swarm on Schaffer 6 ShRot and Rosenbrock Sh ($n = 50$)

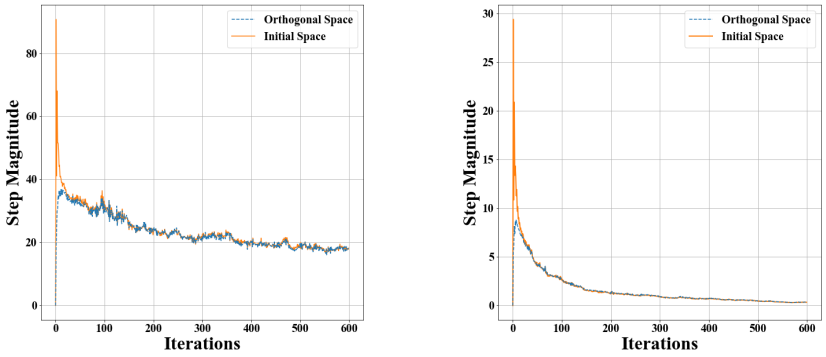

Fig. 10 Normalized by $\frac{1}{\sqrt{n}}$, as in equations (16) and (17): Magnitude of steps in the initial subspace compared to the orthogonal space for a vector swarm on Schaffer 6 ShRot and Rosenbrock Sh ($n = 50$)

where $k = j \bmod g$, so that every k-th dimension is assigned to group k. This induces coupling among the dimensions in the same group, causing the swarm to move linearly in dimensions assigned to the same group. The movement of swarms with a low number of groups will be highly constrained and the swarm is expected to be biased towards searching in the initial subspace. Increasing the number of groups will reduce dimensional coupling and allow the swarm increased freedom of movement. If $g = n$ or $g = 1$, then the swarm is equivalent to the vector or scalar swarms from the previous sections (respectively).

The same benchmark suite from Table 1 in Section 4.1 was used to compare the behaviour and performance of the swarms with different degrees of dimensional coupling. Values for g were in $\{1, 2, 5, 10, 13, 25\}$. All other algorithm

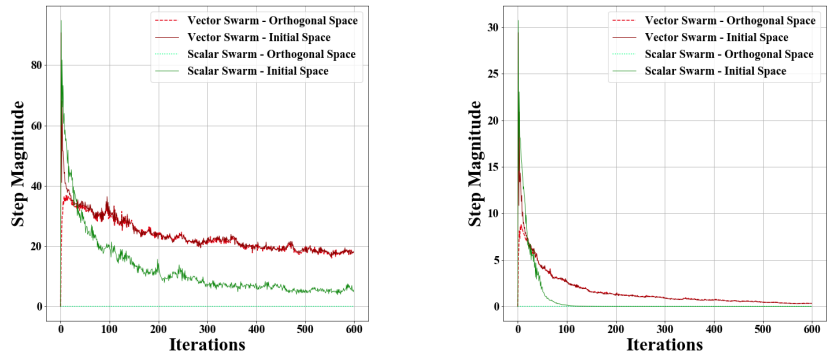

Fig. 11 Step magnitudes inside the initial subspace compared to the orthogonal space for vector and scalar swarms on Schaffer 6 ShRot and Rosenbrock Sh ($n = 50$).

settings were as before, including the swarm size of 10. For these experiments, the problem dimensionality was set to 25.

5.2 Results and Discussion

The magnitude of the swarms' steps in the initial subspace and in the orthogonal subspace are shown in Figure 12. As expected, swarms with low g-values exhibited less exploration in the orthogonal space and focus more on searching in the initial subspace. As the g-value is increased, the step sizes in the orthogonal space become larger (and steps in the initial subspace become smaller) until, eventually, the swarm searches equally in both subspaces.

In general, swarms with fewer groups had smaller step magnitudes and thus explored less than swarms with more dimensional groups. This is illustrated in Figures 13 and 14 which show the step sizes of swarms with different degrees of dimensional coupling. This phenomenon was also observed in the previous section, where the step size of the scalar swarm was smaller than that of the vector swarm, even when comparing only steps inside the initial subspace. The relationship between exploration and g-values is also apparent when examining the swarm diversities (see Figures 15 and 16).

Swarms with fewer dimensional groups exhibited less roaming behaviour than swarms with high g-values, as can be seen in Figure 17. Since particles of swarms with low g-values had smaller step sizes, it may be expected that such swarms would be less prone to roaming.

Table 3 compares the performance of the swarms with induced coupling to a vector swarm. The vector swarm performed significantly better than all the other swarms on most of the benchmark functions. Figures 18 and 19 show typical plots of the best fitness achieved at each iteration by swarms with different degrees of induced coupling. As the number of groups was increased, the

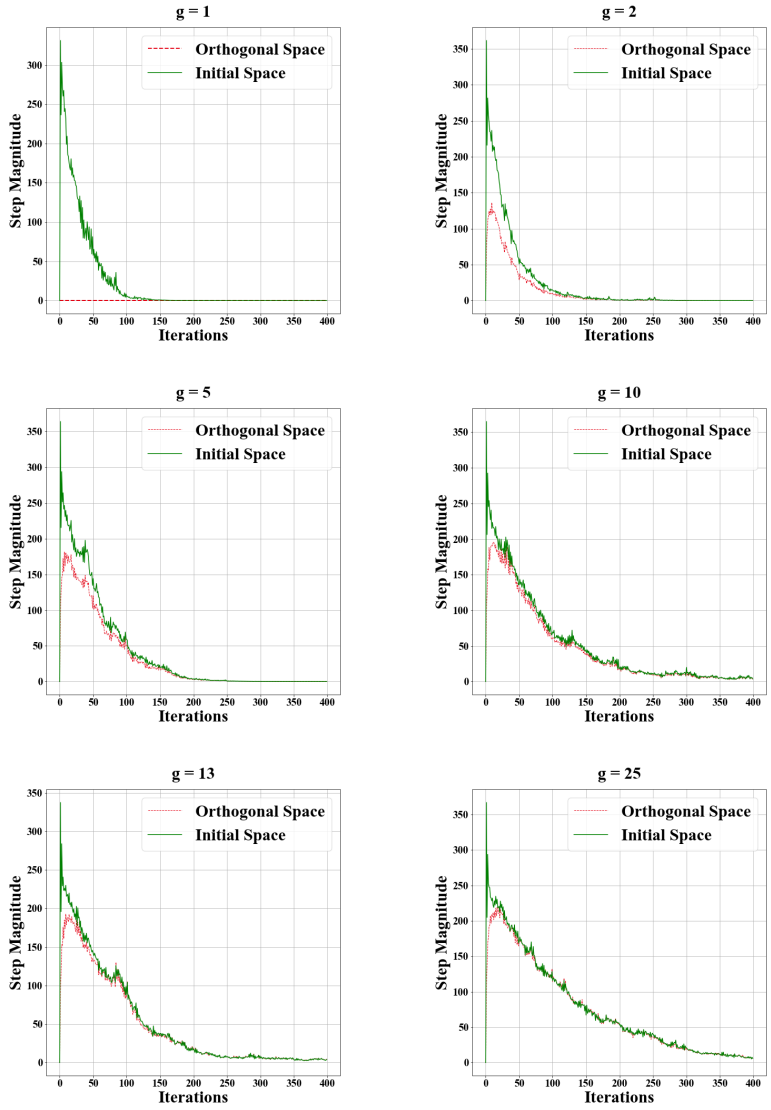

Fig. 12 Step magnitudes inside the initial subspace and the orthogonal subspace, across group numbers (Egg holder, $n = 25$)

swarm's behaviour became more like the vector swarm and their performance became closer to that of the vector swarm.

The scalar swarm can be considered an extreme case for the swarms with grouped dimensions, in which all of the dimensions are allocated to one group. Swarms with more groups, although less constrained than the scalar swarm,

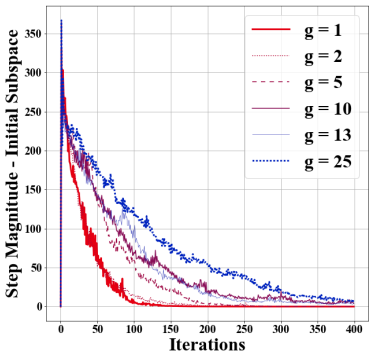

Fig. 13 Magnitude of steps inside the initial subspace for all group numbers on Egg holder ($n = 25$)

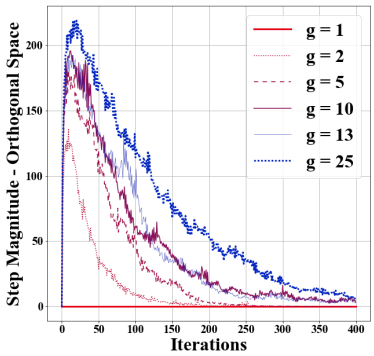

Fig. 14 Magnitude of steps orthogonal to the initial subspace for all group numbers on Egg holder ($n = 25$)

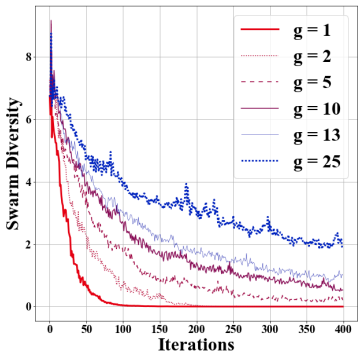

Fig. 15 Diversity, Rastrigin ShRot (f_{12}) ($n = 25$)

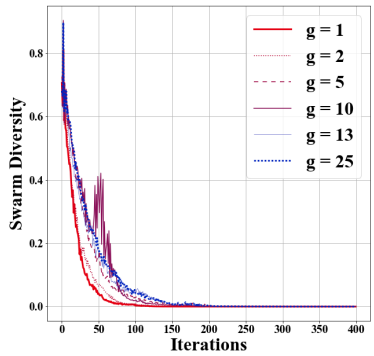

Fig. 16 Diversity, Weierstrass Sh (f_{25}) ($n = 25$)

are still not able to express every point in the search space, since all dimensions within the same group are limited to linear combinations of their initial values.

In many cases, such as on the Egg holder function, the swarm exhibited significantly less roaming when $g < 25$ than for $g = 25$. The step sizes inside and outside the initial subspace (from Figure 12), already appear very similar when $g = 13$ (though the performance comparison shows that there must be a marked difference between the swarms' behaviour). The $g = 13$ swarm thus sacrifices exploration for decreased roaming. In high dimensional spaces where particle roaming is a large concern, this trade-off may be justifiable if a balanced g-value is used. However, the empirical results presented here show that independent, component-wise stochasticity plays an important role in the searching ability and performance of PSO.

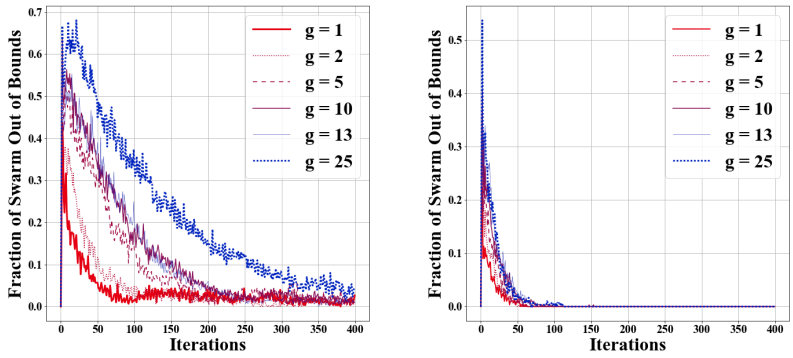

Fig. 17 Fraction of the swarm out of bounds for Egg holder and Griewank Sh over the first 400 iterations ($n = 25$)

Table 3 Comparison of Swarms with Induced Coupling to Vector Swarm

Number of Groups	Coupled Wins	Draws	Vector Wins
$g = 1$	1	4	41
$g = 2$	1	3	42
$g = 5$	2	3	41
$g = 10$	2	2	42
$g = 13$	2	5	39
$g = 25$	0	46	0

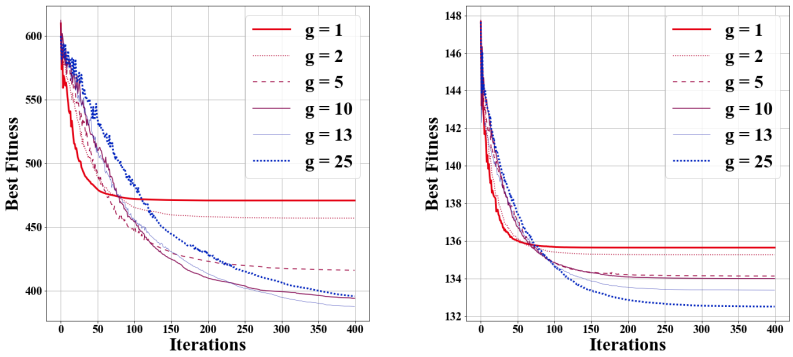

Fig. 18 Best Fitness, Rastrigin ShR (f_{12}, $n = 25$)

Fig. 19 Best Fitness, Weierstrass Sh (f_{25}, $n = 25$)

6 Conclusion

This paper demonstrated the importance of employing component-wise stochasticity both theoretically and empirically. Section 3 showed that a swarm's

movement is severely restricted by using scalar values for r_1 and r_2. In particular, it is emphasized that a scalar swarm is limited to the span of its initial particle positions, personal best positions, and velocities. Thus, the swarm may not be able to reach the optimum.

Section 4 demonstrated the performance difference between scalar and vector swarms on an extensive range of benchmark functions. It was shown that the vector swarm performed significantly better on almost all of the benchmark functions than the scalar swarm, even when the dimensionality is low enough for the scalar swarm to reach the optimum.

The paper challenges previous work regarding the exploration ability of vector swarms. Previous work observed that vector swarms are biased towards searching in the initial subspace. The swarm's exploration ability was quantified by measuring the magnitude of particles' steps in the initial subspace and comparing it to step magnitudes in the orthogonal space. This paper proposes that previously observed bias, which does not agree with theory, is due to incorrect normalization of the measure. The corrected measure, proposed in this paper, shows that a vector swarm explores equally in the initial subspace and its orthogonal space.

The paper also considered the effect of inducing dimensional coupling so that the swarm's behaviour can be analyzed on a spectrum between scalar and component-wise stochasticity. Dimensional coupling was introduced by using the same stochastic scaling value for multiple dimensions. Using the corrected measure for quantifying swarm movement, it was confirmed empirically that swarms with high dimensional coupling are biased towards searching in the initial subspace. The less dimensional coupling is imposed, the less biased the swarm's movement becomes. It is also shown that swarms with high dimensional coupling generally take smaller step sizes overall and exhibit less roaming behaviour than swarms with low dimensional coupling.

The vector swarm performed significantly better than the swarms with induced dimensional coupling on almost all benchmark functions. Swarm performance improved as the number of groups increased and the swarm became more like the vector swarm. As is the case with scalar swarms, swarms with grouped stochastic scalars are not able to reach every point in the search space, which not only limits the swarms' exploration ability but may also prevent the optimum from being attainable.

While there are some problem instances for which fine-tuned, strong dimensional coupling performs better than a vector swarm, the vector swarm remains the overwhelming best first choice.

References

Bratton, D. and Kennedy, J. (2007). Defining a standard for particle swarm optimization. In *Proceedings of the IEEE Swarm Intelligence Symposium*, pages 120–127. IEEE Computer Society.

Chen, S., Montgomery, J., and Bolufé-Röhler, A. (2015). Measuring the curse of dimensionality and its effects on particle swarm optimization and differential evolution. *Applied Intelligence*, 42(3):514–526.

Cleghorn, C. W. and Engelbrecht, A. P. (2018). Particle swarm stability: a theoretical extension using the non-stagnate distribution assumption. *Swarm Intelligence*, 12(1):1–22.

Clerc, M. and Kennedy, J. (2002). The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1):58–73.

Eberhart, R. and Kennedy, J. (1995). A new optimizer using particle swarm theory. In *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pages 39–43.

Engelbrecht, A. P. (2013). Particle swarm optimization: Global best or local best? In *Proceedings of the BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence (BRICS-CCI CBIC)*, pages 124–135.

Engelbrecht, A. P. (2014). Fitness function evaluations: A fair stopping condition? In *Proceedings of the IEEE Symposium on Swarm Intelligence*, pages 1–8.

Feng, Q., Cai, H., Li, F., Liu, X., Liu, S., and Xu, J. (2019). An improved particle swarm optimization method for locating time-varying indoor particle sources. *Building and Environment*, 147:146 – 157.

Fu, H., Li, Z., Liu, Z., and Wang, Z. (2018). Research on Big Data Digging of Hot Topics about Recycled Water Use on Micro-Blog Based on Particle Swarm Optimization. *Sustainability*, 10(7):1–15.

Garca-Gonzalo, E. and Fernndez-Martnez, J. L. (2014). Convergence and stochastic stability analysis of particle swarm optimization variants with generic parameter distributions. *Applied Mathematics and Computation*, 249:286 – 302.

Han, F. and Liu, Q. (2014). A diversity-guided hybrid particle swarm optimization based on gradient search. *Neurocomputing*, 137:234 – 240. Advanced Intelligent Computing Theories and Methodologies.

Jamil, M. and Yang, X.-S. (2013). A literature survey of benchmark functions for global optimization problems. *Int. Journal of Mathematical Modelling and Numerical Optimisation*, 4(2):150194.

Krohling, R. A. and dos Santos Coelho, L. (2006a). Coevolutionary particle swarm optimization using gaussian distribution for solving constrained optimization problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 36(6):1407–1416.

Krohling, R. A. and dos Santos Coelho, L. (2006b). Pso-e: Particle swarm with exponential distribution. In *2006 IEEE International Conference on Evolutionary Computation*, pages 1428–1433.

Li, H., Yang, D., Su, W., L, J., and Yu, X. (2019). An overall distribution particle swarm optimization mppt algorithm for photovoltaic system under partial shading. *IEEE Transactions on Industrial Electronics*, 66(1):265–275.

Malan, K. and Engelbrecht, A. P. (2008). Algorithm comparisons and the significance of population size. *Proceedings of the IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 914–920.

Oldewage, E. T., Engelbrecht, A. P., and Cleghorn, C. W. (2017). The merits of velocity clamping particle swarm optimisation in high dimensional spaces. pages 1–8.

Oldewage, E. T., Engelbrecht, A. P., and Cleghorn, C. W. (2018). The importance of component-wise stochasticity in particle swarm optimization. In *2018 International Conference on Swarm Intelligence (ANTS)*, pages 264–276.

Olorunda, O. and Engelbrecht, A. P. (2008). Measuring exploration/exploitation in particle swarms using swarm diversity. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 1128–1134.

Pandey, S., Wu, L., Guru, S. M., and Buyya, R. (2010). A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. In *2010 24th IEEE International Conference on Advanced Information Networking and Applications*, pages 400–407.

Paquet, U. and Engelbrecht, A. P. (2007). Particle swarms for linearly constrained optimisation. *Fundamental Informatics*, 76(1-2):147–170.

Parsopoulos, K. E. and Vrahatis, M. N. (2010). *Particle Swarm Optimization and Intelligence: Advances and Applications*. Information Science Reference - Imprint of: IGI Publishing, Hershey, PA.

Poole, D. (2011). *Linear Algebra: A Modern Introduction, Third Edition*. Cengage Learning, Canada.

Ramezani, F. and Lotfi, S. (2012). The modified differential evolution algorithm (mdea). In Pan, J.-S., Chen, S.-M., and Nguyen, N., editors, *Intelligent Information and Database Systems*, volume 7198 of *Lecture Notes in Computer Science*, pages 109–118. Springer Berlin Heidelberg.

Shi, Y. and Eberhart, R. (1998). A modified particle swarm optimizer. In *Proceedings of the IEEE International Conference on Evolutionary Computation*, pages 69–73.

Sun, Y., Gao, Y., and Shi, X. (2019). Chaotic multi-objective particle swarm optimization algorithm incorporating clone immunity. *Mathematics*, 7(2).

van Zyl, E. T. and Engelbrecht, A. P. (2016). Group-based stochastic scaling for pso velocities. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pages 1862–1868.

Yoshida, H., Kawata, K., Fukuyama, Y., Takayama, S., and Nakanishi, Y. (2000). A particle swarm optimization for reactive power and voltage control considering voltage security assessment. *IEEE Transactions on Power Systems*, 15(4):1232–1239.

Zahara, E., Kao, Y. T., and Su, J. R. (2009). Enhancing particle swarm optimization with gradient information. In *2009 Fifth International Conference on Natural Computation*, volume 3, pages 251–254.

7 Appendix

The appendix proves Proposition 2, which is restated below for completeness.

Proposition 2 *Suppose $\mathcal{I}$ contains m linearly independent vectors. Let the search space, S, be an n-dimensional hypercube with lower bound L and upper bound U in all dimensions, i.e.$S = [L, U]^n$. If $m < n$ then span$(\mathcal{I}) \cap S \subsetneq S$. Thus $\mathcal{I}$ can only be a spanning set of S if it contains at least n linearly independent elements.*

It is a fundamental result of linear algebra that any two bases for some subspace of $\mathbb{R}^n$ must contain the same number of vectors (Poole 2011). Observe that if the search space, S, is a subspace of $\mathbb{R}^n$, then the standard unit vectors

$$\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_n =< 1, 0, 0, ..., 0 >^T, < 0, 1, 0, 0, ... >^T, ..., < 0, ..., 0, 1 >^T \quad (19)$$

form a basis for S. Thus, for $\mathcal{I}$ to span S, it must contain at least n linearly independent vectors. However, the search space $S = [L, U]^n$ does not constitute a subspace of $\mathbb{R}^n$, since it is not closed under addition and scalar multiplication. Proposition 2 proves the required result for this special case. The proof requires the use of the fundamental theorem of invertible matrices (given below from Poole (2011)) and Lemma 1.

Theorem 1 (Fundamental Theorem of Invertible Matrices (Poole 2011)) *Let A be an n-by-m matrix. Then the following statements are equivalent:*

1. *A is invertible*
2. *$A\mathbf{x} = \mathbf{0}$ has only the trivial solution*
3. *The column vectors of A are linearly independent*
4. *The column vectors of A span $\mathbb{R}^n$*

Lemma 1 *Let $S = [L, U]^n$. There exists a set $\tilde{E}$ that forms a "basis" for S, in the sense that $\tilde{E}$'s elements are linearly independent, any $\mathbf{x} \in S$ can be expressed as a linear combination of elements in S and $\tilde{E} \subset S$.*

Proof Let the center of the search space be denoted by

$$\mathbf{M} =< m_1, m_2, ..., m_n >^T = \left\langle \frac{L+U}{2}, \frac{L+U}{2}, ..., \frac{L+U}{2} \right\rangle^T \quad (20)$$

Since the search space is the same in every dimension, $\mathbf{M} =< c, c, ..., c >^T$ where $c = \frac{L+U}{2}$.

Let $\tilde{E} = \{\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2, ..., \tilde{\mathbf{e}}_n\}$ where

$$\begin{aligned} \tilde{\mathbf{e}}_1 &=< c + 0.5c, c, ..., c >^T \\ \tilde{\mathbf{e}}_2 &=< c, c + 0.5c, ..., c >^T \\ &... \\ \tilde{\mathbf{e}}_n &=< c, c, ..., c + 0.5c >^T \end{aligned} \quad (21)$$

so that for any $\tilde{\mathbf{e}}_i$, all coordinates except the i-th coordinate are equal to c. The i-th coordinate will be equal to $1.5c$. Clearly, $\tilde{E} \subset S$. Let A be the n by n matrix with column vectors $\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2, ..., \tilde{\mathbf{e}}_n$.

We prove that $A\mathbf{x} = \mathbf{0}$ has only the trivial solution. Then, by Theorem 1, it will follow that:

1. A's column vectors are linearly independent. In other words, $\tilde{E}$ is linearly independent.
2. The column vectors of A span $\mathbb{R}^n$, which means that E spans $\mathbb{R}^n$. Since $S \subset \mathbb{R}^n$, any element in S can thus be expressed as a linear combination of elements from E.

It will now be proved that $A\mathbf{x} = \mathbf{0}$ has only the trivial solution, thereby proving the required properties of $\tilde{E}$. The system in $A\mathbf{x} = \mathbf{0}$ can be written as a system of linear equations as illustrated below:

$$\begin{array}{rcrcccrcl} (c+0.5c)x_1 & + & cx_2 & + & ... & + & cx_n & = & 0 \\ cx_1 & + & (c+0.5c)x_2 & + & ... & + & cx_n & = & 0 \\ & & & \vdots & & & & & \\ cx_1 & + & cx_2 & + & ... & + & (c+0.5c)x_n & = & 0 \end{array} \tag{22}$$

were the j-th equation is formed from the j component of the system $A\mathbf{x} = \mathbf{0}$. In turn, the system of linear equations can be represented in augmented matrix form as below:

$$\begin{pmatrix} 1.5c & c & c & ... & c & 0 \\ c & 1.5c & c & ... & c & 0 \\ ... & ... & ... & ... & ... & ... \\ c & c & c & ... & 1.5c & 0 \end{pmatrix} \tag{23}$$

where the i-th column contains the coefficients of x_i. The final column contains the right hand side of all the equations in the linear system (in this case, the zero vector). The augmented matrix is manipulated by means of matrix row operations, which take one of the following forms:

1. Switch the i-th and j-th rows (denoted by $R_i \leftrightarrow R_j$, where R_i denotes the i-th row).
2. Multiply the i-th row by c, a non-zero constant (denoted by $R_i \rightarrow cR_i$).
3. Add row j to row i (denoted by $R_i \rightarrow R_i + R_j$).

From these basic operations, more complex operations can be constructed such as subtracting one row from another ($R_i \rightarrow R_i - R_j$). For the sake of brevity, the notation $\forall_i R_i \rightarrow cR_i$ denotes that for each row i, the row is multiplied by a constant c. Similarly, $\forall_i R_i \rightarrow R_i - R_j$ indicates that the j-th row is subtracted from each of the other rows in turn. The system $A\mathbf{x} = \mathbf{0}$ will now be solved

below.

$$\left(\begin{array}{ccccc|c} 1.5c & c & c & ... & c & 0 \\ c & 1.5c & c & ... & c & 0 \\ ... & ... & ... & ... & ... & ... \\ c & c & c & ... & 1.5c & 0 \end{array}\right) \forall_i R_i \to \frac{1}{c} R_i \left(\begin{array}{ccccc|c} 1.5 & 1 & 1 & ... & 1 & 0 \\ 1 & 1.5 & 1 & ... & 1 & 0 \\ ... & ... & ... & ... & ... & ... \\ 1 & 1 & 1 & ... & 1.5 & 0 \end{array}\right)$$

$$\forall_i R_i \to R_i - R_n \left(\begin{array}{ccccc|c} 0.5 & 0 & 0 & ... & -0.5 & 0 \\ 0 & 0.5 & 0 & ... & -0.5 & 0 \\ ... & ... & ... & ... & ... & ... \\ 1 & 1 & 1 & ... & 1.5 & 0 \end{array}\right) \forall_i R_i \to 2R_i \left(\begin{array}{ccccc|c} 1 & 0 & 0 & ... & -1 & 0 \\ 0 & 1 & 0 & ... & -1 & 0 \\ ... & ... & ... & ... & ... & ... \\ 1 & 1 & 1 & ... & 1.5 & 0 \end{array}\right)$$

$$R_n \to R_n - R_1 - R_2 - ... - R_{n-1} \left(\begin{array}{ccccc|c} 1 & 0 & 0 & ... & -1 & 0 \\ 0 & 1 & 0 & ... & -1 & 0 \\ ... & ... & ... & ... & ... & ... \\ 0 & 0 & 0 & ... & 1.5-(-1)(n-1) & 0 \end{array}\right)$$

Therefore, $x_n(n+0.5) = 0$. Since $n \geq 1$, the equation can only have the trivial solution, $x_n = 0$. But then, for every $i = 1, ..., n-1$,

$$\begin{aligned} & x_i - x_n = 0 \\ \implies & x_i - 0 = 0 \\ \implies & x_i = 0 \end{aligned} \tag{24}$$

Therefore, $A\mathbf{x} = \mathbf{0}$ has only the trivial solution. The existence of the required set $\tilde{E}$ is thus proved.

Using Lemma 1, Proposition 2 can be proved as below.

Proof (Proposition 2) By Lemma 1, there exists a set $\tilde{E}$ that forms a "basis" for S. The set $\tilde{E}$ will be used to prove that if $\mathcal{I}$ contains m linearly independent vectors and $m < n$, then $\mathcal{I}$ spans only a subset of S.

Let $m < n$. Towards a contradiction, suppose that $\mathcal{I}$ and $\tilde{E}$ both form "bases" for S. In other words, $span(\mathcal{I}) \cap S = span(\tilde{E}) \cap S = S$.
Consider the equation,

$$c_1\tilde{\mathbf{e}}_1 + c_2\tilde{\mathbf{e}}_2 + ... + c_n\tilde{\mathbf{e}}_n = 0 \tag{25}$$

where $\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2, ..., \tilde{\mathbf{e}}_n \in \tilde{E}$ and $c_1, c_2, ..., c_n \in \mathbb{R}$. Since each $\tilde{\mathbf{e}}_i \in S$ and any element in S can be expressed as a linear combination of vectors in $\mathcal{I}$, each $\tilde{\mathbf{e}}_i$ can be written as

$$\begin{aligned} \tilde{\mathbf{e}}_1 &= a_{11}\mathbf{z}_1 + a_{12}\mathbf{z}_2 + ... + a_{1m}\mathbf{z}_m \\ &\vdots \\ \tilde{\mathbf{e}}_i &= a_{i1}\mathbf{z}_1 + a_{i2}\mathbf{z}_2 + ... + a_{im}\mathbf{z}_m \\ &\vdots \\ \tilde{\mathbf{e}}_n &= a_{n1}\mathbf{z}_1 + a_{n2}\mathbf{z}_2 + ... + a_{nm}\mathbf{z}_m \end{aligned}$$

Thus, equation (25) can be rewritten as follows:

$$\begin{aligned} & c_1(a_{11}\mathbf{z}_1 + a_{12}\mathbf{z}_2 + ... + a_{1m}\mathbf{z}_m) + ... \\ & \quad + c_i(a_{i1}\mathbf{z}_1 + a_{i2}\mathbf{z}_2 + ... + a_{im}\mathbf{z}_m) + ... \\ & \quad + c_n(a_{n1}\mathbf{z}_1 + a_{n2}\mathbf{z}_2 + ... + a_{nm}\mathbf{z}_m) \qquad (26) \\ & = (c_1a_{11} + c_2a_{21} + ... + c_na_{n1})\mathbf{z}_1 + ... \\ & \quad + (c_1a_{1j} + c_2a_{2j} + ... + c_na_{nj})\mathbf{z}_j + ... \\ & \quad + (c_1a_{1m} + c_2a_{2m} + ... + c_na_{nm})\mathbf{z}_m \\ = & \sum_{i=1}^{n} c_ia_{i1}\mathbf{z}_1 + \sum_{i=1}^{n} c_ia_{i2}\mathbf{z}_2 + ... + \sum_{i=1}^{n} c_ia_{im}\mathbf{z}_m \qquad (27) \end{aligned}$$

Now, $\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_m$ are linearly independent. Therefore, equation (27) has only the trivial solution and $\sum_{i=1}^{n} c_ia_{ij}$ must equal zero for all $j = 1, ..., m$. This can be written as a homogeneous system of m equations, each with n variables, $c_1, ..., c_n$. Since $m < n$, there are more variables than equations, so there must be infinitely many solutions. Particularly, there must be a non-trivial solution. But, this gives a non-trivial dependence relation in equation (25). By definition, $\tilde{E}$ must thus be a linearly dependent set of vectors. But this is a contradiction, since $\tilde{E}$ is linearly independent. Therefore, $\mathcal{I}$ spans a strict subset of S.