Movement Patterns of a Particle Swarm in High Dimensional Spaces

Oldewage, E.T.[a,*], Engelbrecht, A.P.[b], Cleghorn, C.W.[c]

[a] *Department of Engineering, University of Cambridge, Cambridge, UK*
[b] *Department of Industrial Engineering and Computer Science Division, Stellenbosch University, Stellenbosch, South Africa*
[c] *Department of Computer Science, University of Pretoria, Pretoria, South Africa*

Abstract

In high dimensional problem spaces, particle swarm optimization (PSO) is prone to unwanted roaming behaviour due to initial velocity explosion. A particle swarm's movement patterns are strongly influenced by the inertia weight and acceleration coefficients. This paper investigates whether the initial velocity explosion can be curbed by appropriate choice of the inertia weight and the acceleration coefficients, which restrict the standard deviation of particle positions. It is shown that roaming behaviour cannot be solved by reducing swarm variance directly, but that the relationship between the parameters must also be considered. Furthermore, the paper investigates different movement patterns that may be exhibited by the swarm. It is shown that optimal parameter configurations differ between low and high dimensional problems. Specifically, parameter configurations which produce very smooth particle trajectories and restrict the swarm's movement range are advantageous in high dimensional spaces. These movement patterns correspond to high inertia weight and low acceleration coefficients (eg. $w = 0.9694$, $c_1 = c_2 = 0.099381$). Swarms with smooth particle trajectories exhibited significantly less unwanted roaming behaviour than swarms with chaotic or oscillating particle trajectories.

Keywords: particle swarm optimization, large scale optimization, swarm

[*] Corresponding author
Email address: etv21@cam.ac.uk (Oldewage, E.T.)

movement patterns, inertia weight, acceleration coefficients, high dimensions

1. Introduction

Particle swarm optimization (PSO) has been shown to suffer from the "curse of dimensionality" [9]. The "curse of dimensionality", a term coined by Bellman in his book on dynamic programming in 1957 [1], generally refers to the unintuitive phenomena that arise in high dimensional spaces, but are typically not observed in low dimensional spaces. One example is that the notion of proximity may become ill-defined as dimensionality increases. As shown by Beyer et al., for certain sampling distributions and distance metrics, all points in a sample approach the same distance apart as the problem dimensionality goes to infinity [2].

High dimensional phenomena generally arise from the exponential increase in hyper-volume of the corresponding problem space. The exponential increase in the search space's hyper-volume causes data points to be sparse, making it difficult to perform accurate analysis of the space. Due to the sheer size of the search space, algorithms and techniques that require systematic sampling or exploration of the search space often become infeasible.

Literature has shown that PSO parameter configurations which are successful in low dimensional problem spaces often perform poorly when the problem dimensionality increases [18, 40, 41]. Thus, in addition to the problem becoming intrinsically harder due to the exponential increase in search space hyper-volume, the algorithm parameters must be tuned especially for high dimensional problems. Tuning for high dimensional problems typically requires a great deal of computation, since the objective function evaluations become more computationally expensive with increasing dimensionality. Guidance regarding parameter selection thus becomes invaluable for high dimensional problems.

Unwanted particle roaming forms a substantial part of the problematic behaviour exhibited by PSO in high dimensional search spaces [18, 21, 40, 41]. Particle roaming refers to a phenomenon that usually occurs in the first few iterations of the search where particles are likely to leave the search space due to velocity explosion. In low dimensional search spaces, the particles usually return to the search space if no better solution is found outside the search space. In fact, early roaming behaviour may even be beneficial to the search if the attractors are constrained inside the search space [17]. However, in high dimensional spaces, the particles fail to return to the search space [40, 41].

The relationship between particle roaming and problem dimensionality has been considered theoretically in literature. Helwig and Wanka related problem dimensionality to the probability of particles leaving the search space [21]. It was proven that the probability of a particle leaving the search space rapidly approaches 1 as the dimensionality, n, is increased. The result was proven for any neighbourhood topology and for a number of velocity initialization strategies (including uniform random initialization, initialization to zero, and initialization to half the difference between the upper and lower search space bounds).

Several methods of reducing particle roaming behaviour have been examined in literature, the principal method being boundary handling methods. However, most boundary handling methods have been shown to bias the search behaviour of the particles [19, 20, 23, 29]. The optimal boundary handling method is thus heavily problem-dependent. Other ways of addressing particle roaming include different initialization strategies [40] and introducing coupling between problem variables [41]. This paper investigates whether the swarm's roaming can be prevented by appropriate parameter selection. Such an approach does not bias the swarm's search behaviour directly since it does not modify the position and velocity update equations.

Two sets of experiments are performed: the first examined the variance of particle positions and attempted to prevent the initial velocity explosion by selecting algorithm parameters that bring about small variance in particle positions. The second experiment selected algorithm parameters based on existing literature, that are known to bring about certain movement patterns. These swarm configurations were then tested on high dimensional problems to determine whether particular types of movement are more successful in high dimensional spaces.

Key contributions of this paper include:

- showing that reducing the variance of particle positions is, by itself, not sufficient to prevent particle roaming.
- showing that the optimal movement strategies for PSO differ significantly between high and low dimensional problems. Specifically, movement patterns that produce very smooth particle trajectories and restrict the swarm's range of movement perform well on high dimensional problems. In low dimensions, movement strategies that are less restrictive and produce more chaotic particle trajectories perform best.
- showing that particle roaming in high dimensions can be prevented by appropriate choice of acceleration coefficients and inertia weight, such as $w = 0.9694$ and $c_1 = c_2 = 0.099381$.

The remainder of the paper proceeds as follows: section 2 provides a brief overview of PSO, introduces the necessary notation, and discusses existing literature regarding the variance of particle positions. Section 3 discusses the first experiment, in which particle roaming is addressed by reducing particle variance through the choice of parameters. Section 4 discusses the second experiment, which identifies movement patterns that are advantageous in high dimensional search spaces. Section 5 considers whether velocity clamping can be applied to further improve the performance of configurations that perform well in high dimensional spaces. Finally, section 6 concludes the paper and describes possible directions of future work. B provides additional results for the experiments described in sections 3 and 4.

2. Background

This section provides a brief overview of the particle swarm optimization algorithm and introduces the notation used throughout the rest of the paper.

This section also discusses existing literature regarding to the variance of particle movement.

2.1. Particle Swarm Optimization

Particle swarm optimization (PSO) is a population-based, stochastic optimization algorithm, first proposed by Eberhart and Kennedy [13]. A swarm of particles, each representing a solution to the optimization problem are initialized in the search space. The particles move through the search space in search of an optimal solution for a number of iterations. A particle's movement is guided by

- the particle's momentum.
- knowledge of the best position that the particle has encountered thus far, called the personal best position.
- knowledge of the best position encountered by other particles in the swarm, called the neighbourhood best position or the global best position (depending on whether the set of other particles is a strict subset of the swarm).

A particle's movement can be described exactly by the position and velocity update equations. At iteration t, particle i's position is updated according to

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

where $\mathbf{x}_i^t$ and $\mathbf{v}_i^{t+1}$ respectively denote the position and velocity of particle i at iteration t. This paper considers PSO with inertia weight as introduced by [35]. The velocity update equation for the j-th dimension is given by

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

The three terms in the velocity update equation correspond to the three components of particle movement listed previously: the first term is the momentum component. The second term is called the cognitive component and causes the particle to move towards its personal best position, $\mathbf{y}_i^t$. The final term is called the social component and causes the particle to move towards the best position in its neighbourhood, denoted by $\hat{\mathbf{y}}_i^t$. If the swarm uses a ring topology, then a particle's neighbourhood consists of the particle itself and its immediate two neighbours. This paper makes use of the star neighbourhood topology, which defines a particle's neighbourhood to be the entire swarm. All particles are thus guided by the global best position, $\hat{\mathbf{y}}^t$ so that $\hat{\mathbf{y}}_i^t = \hat{\mathbf{y}}^t$ for every particle i. According to literature [16], neither the ring nor the star topology is always optimal for a given class of optimization problems. The star topology is used in this paper because the swarm's movement dynamics are more readily apparent using a global best position.

Every dimension of the cognitive and social components are also weighted by uniform random variables, $r_{1,j}$ and $r_{2,j}$, resampled at every iteration in the range $[0, 1]$. This introduces stochasticity to the search, thereby allowing the swarm to explore the search space in the areas between and slightly behind its current position and its attractors (the personal and global best positions).

The inertia weight, $w \in (0, 1)$, provides control over the balance of exploration to exploitation. There must be a balance between exploration and exploitation: a swarm that only explores and never refines good solutions will waste resources exploring fruitless areas of the search space. A swarm that only exploits is likely to converge prematurely to a sub-optimal solution. The optimal balance between exploration and exploitation is problem dependent [6] [8]. The momentum component causes each particle to maintain its current trajectory. This forces the particle to move through areas of the search space that may not be directly on course to previously observed successes [35]. The momentum component also prevents the particle from oscillating rapidly if its personal and global best positions update frequently and are far from their previous locations.

The cognitive and social acceleration coefficients, c_1 and c_2, govern the influence of the personal best and global best positions respectively. The values of w, c_1 and c_2 play a large role in determining the swarm's search behaviour and can be used to achieve a balance between exploration and exploitation. As will be discussed in section 2.2, these parameters influence important characteristics of swarm movement such as the variance of particle positions. The parameters also determine whether the swarm diverges to infinity, exhibits oscillatory behaviour or converges [38, 39]. The relationship between these parameters and swarm behaviour is the main focus of the paper. Particularly, this paper determines what movement patterns are more successful in high dimensional search spaces.

2.2. A Brief History of Variance

This section provides a brief overview of existing literature regarding the variance of particle positions and its relationship to the inertia weight and acceleration coefficients.

Theoretical analysis of particle behaviour usually requires some form of the *stagnation assumption* which assumes that the personal best positions and the global best position have stopped improving. Under stagnation, each particle behaves independently of the other particles since no new information is introduced by a global or personal best position update. Thus, each particle's behaviour can be studied separately when stagnation is assumed. Furthermore, each particle's dimensions are independent, so the particle need only be analyzed in one dimension without loss of generality. This significantly reduces the complexity of theoretical analysis. Due to the independence of particles and problem dimensions, the subscript i and j for particle positions and velocities may be dropped for the purposes of the discussion that follows.

The convergence of particle positions in terms of both expectation and variance has been studied extensively throughout literature [22, 30, 26]. These studies yielded *convergence regions* for the choice of w, c_1 and c_2 parameter values, within which all the particles' positions are guaranteed to converge in expectation.

A detailed overview of the older literature is provided by [5] and [11]. In recent literature, [3] derived convergence boundaries that are necessary and sufficient for convergence of position variance when assuming that y and $\hat{y}$ are random variables (as opposed to constant values, as in earlier literature). Cleghorn and Engelbrecht made a further generalization under the non-stagnant distribution assumption (i.e. without assuming stagnation) where the personal and global best positions were considered as convergent sequences of random variables [10, 11]. The analysis presented by Cleghorn and Engelbrecht can be applied to obtain the convergence boundaries for general classes of PSOs. Furthermore, the assumption that the expected value and variance of personal and global best positions are convergent sequences was shown to be a necessary condition for convergence (i.e. is the weakest possible assumption under which the expected value and variance of particle positions converge).

2.3. Particle Movement Patterns

The values for the coefficients w, c_1 and c_2 play a large role in determining a swarm's behaviour. Existing literature has studied the relationship between swarm parameters and the corresponding movement patterns of the swarm particles [4, 12, 37]. A detailed overview of the existing research regarding movement patterns is provided by [5]. [37] categorized the movement patterns into four groups:

1. non-oscillatory - the particle's position does not oscillate throughout the search,
2. harmonic - the particle's position oscillates smoothly in a wave-like motion,
3. zigzagging - the particle's position oscillates significantly from one iteration to the next, and
4. harmonic-zigzagging - the particle's position displays a combination of harmonic and zigzagging motion.

Bonyadi and Michalewicz provided a generalization of these categories by performing frequency analysis [4], which is discussed throughout the remainder of this section.

A particle's trajectory may be characterized in terms of the particle's range of movement and its base frequency. The base frequency of a particle, denoted by F, is defined to be the largest amplitude among the Fourier series coefficients of the particle's positions throughout the search [4]. Particles with small values for F typically exhibit smooth trajectories, while large F-values are prone to more oscillations with large steps between positions.

Range of movement refers to the size of the hyper-volume bounded by all the particles' positions from the start to the end of the search. The range of movement is characterized by the variance of a particle's position, denoted by σ^2 where σ is as defined by [30]:

$$\begin{aligned}\sigma &= \frac{1}{2}\sqrt{\frac{c(w+1)}{c(5w-7)-12w^2+12}}\,|\hat{y}-y| \\ &= V_c|\hat{y}-y| \qquad (3)\end{aligned}$$

where $c_1 = c_2 = c$ and V_c is given by

$$V_c = \frac{1}{2}\sqrt{\frac{c(w+1)}{c(5w-7)-12w^2+12}} \qquad (4)$$

The coefficient V_c is independent of the global and local best positions, and characterizes the swarm's ability to explore.

For a given F and V_c, corresponding velocity update coefficients, w, c_1 and c_2 can be calculated as described by [4]. If the velocity update coefficients are assumed to be constant, and $c_1 = c_2 = c$ and $w > 0$, then the simultaneous equations (5) and (6) are sufficient to calculate the corresponding velocity update coefficients:

$$c = 1 + w - 2cos(2\pi F)\sqrt{w} \qquad (5)$$

$$c = \frac{-48V_cw^2 + 48V_c}{28V_c + w - 20V_cw + 1} \qquad (6)$$

Equations (5) and (6) allow the practitioner to choose a pattern of movement for particle trajectories that is most suited to the problem and to calculate values for w, c_1 and c_2 that bring about the desired behaviour.

The base frequency, F should be in the range $[0, 0.5]$ where $F = 0$ implies that the particle's position does not oscillate at all. On the other hand, $F = 0.5$ means that the particle's position will oscillate with every iteration. Figure

1 from [4] shows the correlation measure as a function of the base frequency. Each point is the correlation between corresponding particle positions for a given base frequency, averaged across different variance values (with bars indicating the standard deviation of the correlation measure). Bonyadi and Michalewicz observed that positions are positively correlated when $F < 0.25$, so the particle moves smoothly, with high dependence between subsequent positions. When $F > 0.25$, particle positions are negatively correlated implying no dependence between subsequent positions. When F is near 0.25, the correlation between particle positions is close to zero, so particle movement may be chaotic.

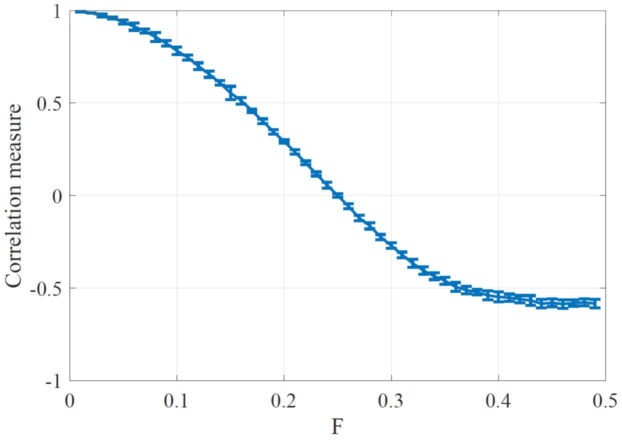

Figure 1: Relationship between base frequency and correlation of particle positions [4]

The value for the variance of movement, V_c, can be any number larger than zero. However, if the value of V_c is very small, then the particle is not guaranteed to sample positions outside the boundaries $[\min\{y, \hat{y}\}, \max\{y, \hat{y}\}]$, because the expected value of the particle's position is between its personal and global best positions [38]. Thus, the particle's exploration ability will be limited.

The following theorem from [4] (simplified for constant w, c_1 and c_2) provides a minimum bound for V_c to ensure that at least one of the particle's positions is not on the line between y and $\hat{y}$.

Theorem 1. *Assume that $y \neq \hat{y}$. If*

$$V_c > \left(\frac{\max\{c_1, c_2\}}{c_1 + c_2}\right)^2 \tag{7}$$

then for any distribution of x_i^t generated by the velocity update equation (2), the number of points generated by x_i^t outside the interval

$$(\mathbb{E}(x_i^t) - \min\{y, \hat{y}\}, \mathbb{E}(x_i^t) + \max\{y, \hat{y}\}) \tag{8}$$

is non-zero.

Bonyadi and Michalewicz suggested that large V_c-values are generally the better choice, observing that small values of V_c prevent the particles from exploring the search space sufficiently [4]. The experiments performed by [4] were only performed for low dimensional problems, with $n = 10$ and $n = 30$. In high dimensional spaces, literature suggests that search strategies which focus on local exploitation may be more effective than searches that attempt exploration, due to the exponential growth of the search space with dimensionality [40, 41]. Thus, smaller values for V_c may prove better than large values, by curbing the particles' exploration.

3. Restricting Variance

This section uses existing theory regarding the variance of particle positions to restrict particle movement. Section 3.1 explains that the variance of particle positions can be reduced to a chosen fraction of its original value by calculating corresponding values for the inertia weight and acceleration coefficients. Swarms with variance restricted to different values were then tested on the benchmark suite from the CEC'2010 special session and competition on large-scale optimization [36]. Section 3.2 describes the experimental method and section 3.3 analyses the results of the experiments. The effects of restricting the variance are discussed and the best performing swarm configurations are identified. Section 3.4 summarizes this section and provides motivation for the section that follows.

3.1. Restricting the Variance of Particle Positions

It has been shown that particles exhibit an initial velocity explosion which leads to particle roaming behaviour [18, 21, 40, 41]. In low dimensional search spaces, the particles usually return to the search space and the search continues to progress [17]. However, it has been shown that in high dimensional search spaces, the particles leave the search space immediately and fail to return [40, 41, 18]. Continued roaming behaviour occurs even when using convergent parameter values, which guarantee that the expected value and variance of the particle positions will converge to a constant.

Due to the initial velocity explosion, the particles leave the search space immediately. As proposed by [15] as well as [7], the personal and global best positions are not updated unless they are within the search space. Thus, the personal best and global best positions of the swarm are not updated while the swarm is roaming because all the particles are out of bounds. In fact, since the particles fail to return to the search space, the personal and global best positions are never updated. The swarm thus converges within the first iteration and the variance of the particle positions immediately becomes a large constant. Even if the attractors are confined to the search space, the resulting variance is large enough for all the particles to stay outside the search space. After the velocity explosion, the remainder of the search thus takes place under stagnation.

According to [30], the standard deviation of a particle's position under stagnation is given by

$$\begin{aligned}\sigma &= \frac{1}{2}\sqrt{\frac{c(w+1)}{c(5w-7)-12w^2+12}}\,|\hat{y}-y| \\ &= V_c|\hat{y}-y| \qquad (9)\end{aligned}$$

where $c_1 = c_2 = c$ and V_c is given by equation (4). If the conditions for convergence in expectation and variance as given by [30] hold, then the standard deviation will only be zero if $y = \hat{y}$.

For the usual "good parameters" for the inertia weight and acceleration coefficients ($w = 0.7298$ and $c_1 = c_2 = 1.49618$) [12], the coefficient in equation

(9) evaluates to 1.0432, which is larger than one. If the upper boundary of the search space is denoted by U and the lower boundary is denoted by L, then the maximum possible value for $|y - \hat{y}|$ is given by $U - L$, i.e. the range of the search space. Thus, for the given c and w values, the maximum possible standard deviation is larger than the size of the search space. It has been shown that under stagnation, when the personal and global best are approximately equidistant from the center of the search space, and initial particle positions are set to zero, then particle positions appear to be distributed normally around the center of the search space [24]. For the argument that follows, it is thus assumed that the particles are distributed normally around the center of the search space. If $|y - \hat{y}| = U - L$ for each of the particles in the swarm, then approximately 38% of the particles will be located within the search space (i.e. within half a standard deviation of the center). The probability of a particle's next position being outside the search space is thus much higher than the probability of being inside the search space, therefore most of the swarm will be located outside of the search space (see figure 2).

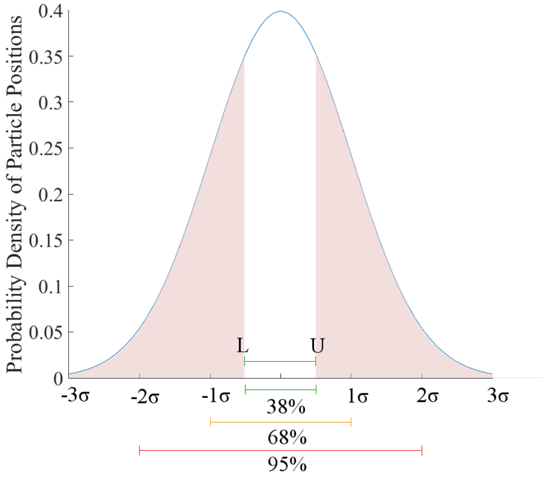

Figure 2: Probability distribution of swarm for worst case scenario in 1-D

Of course, this is a worst case scenario. Consider a better scenario, in which the distance between the personal best position and the global best position is approximately half of the search space, i.e. $|y - \hat{y}| = \frac{1}{2}(U - L)$. On average, approximately 68% of the particles will be found within the search space (i.e. within one standard deviation of the center), which leaves a third of the particles outside the search space.

If the value of $|y - \hat{y}|$ is known or can be estimated, then equation (9) can be solved for values of c and w that ensure that a particle is likely to stay within the search space. If y and $\hat{y}$ are both independent, uniformly distributed variables between L and U, then the expected value of $|y - \hat{y}|$ is zero. However, y and $\hat{y}$ are not independent, since both variables depend on the location of the benchmark function's optimum.

Thus, calculating the coefficient V_c to obtain an exact standard deviation is a difficult, problem-dependent exercise. Instead, the value of the coefficient V_c can be used to restrict the standard deviation fractionally, i.e. to some fraction of what it would be otherwise. For example, suppose that the aim is to restrict the standard deviation to some fraction, γ, of what it would be otherwise. Then setting the value of V_c to γ and solving for c yields:

$$\gamma = V_c = \frac{1}{2}\sqrt{\frac{c(w+1)}{c(5w-7) - 12w^2 + 12}} \tag{10}$$

$$\therefore c = \frac{12(1-w^2)}{7 - 5w + \frac{w+1}{4\gamma^2}} \tag{11}$$

By allowing w to range between 0 and 1 and solving for the corresponding c-value, it is possible to produce a set of (c, w) pairs for which the standard deviation will be the desired fraction of what it would have been otherwise.

Two questions arise in the context of large scale optimization:

- To what fraction should the standard deviation be restricted?
- How are the optimal w and c chosen from the resulting sets?

Both of these questions will be considered in the empirical study presented in section 3.3. The second question, regarding the optimal relationship between c

and w, is investigated further in section 4. The remainder of this section is devoted to motivating the method of restricting the standard deviation fractionally and the applicability to high dimensional spaces in particular.

The discussion regarding the standard deviation of a particle's position has thus far considered only a single dimension. An n-dimensional particle's standard deviation, σ, will be a vector with each component j given by

$$\sigma_j = V_c|\hat{y}_j - y_j| \tag{12}$$

A particle is considered out of bounds if the boundary constraints are violated in even one dimension. If the variance of a component is large, then the particle is more likely to leave the search space in that dimension. As the problem dimensionality grows, the probability of $|\hat{y}_j - y_j|$ being large enough to cause roaming in at least one dimension increases. This information is captured well by the infinity norm:

$$\begin{aligned} \|\sigma\|_\infty &= \max_{j=1..n} \{|\sigma_j|\} && (13) \\ &= V_c \max_{j=1..n} \{|\hat{y}_j - y_j|\} && (14) \end{aligned}$$

It may thus be possible to obtain coefficients that will negate the effect of dimensionality on the standard deviation's norm and restrict the particles' movement, hopefully preventing roaming.

Note that even if the swarm was not normally distributed (eg. the personal and global best positions are not maximally far apart), then restricting the variance may reduce particle roaming by the same argument as above.

Restricting the variance of particle positions will weaken the swarm's ability to explore, especially in dimensions where a particle's variance is small. However, literature shows that, in practice, a local optimum found by a swarm with weakened exploration ability is often better than the optimum found by a swarm which attempts to explore the entire search space [18, 40, 41]. When the problem dimensionality is sufficiently high, it becomes infeasible to explore the search space thoroughly. Instead, the swarm should focus on local searching.

3.2. Experimental Method

An experiment was performed to test whether restricting swarm variance can mitigate particle roaming and achieve good performance in high dimensional search spaces. This section describes the experimental methodology that was followed.

Empirical work has shown that swarms in high dimensions roam regardless of swarm size [27]. Specifically, on 1000-dimensional problems, swarms as small as 5 particles or as large as 250 exhibit continual roaming behaviour. The paper thus considers a fixed swarm size for the experiments regarding the effect of restricting swarm variance. The experiment was performed using a swarm size of 50. Larger swarm sizes required more objective function evaluations and led to long execution times and smaller swarms exhibited large variance in performance, so more runs would have been required.

No velocity clamping was applied so that the effect of the different parameter values are not obscured by applying clamping. Swarm updates were performed synchronously. The swarm was allowed to run for 5000 iterations, which provides a total of 50×5000 function evaluations for each simulation.

Initial particle velocities were set to zero [14]. Particle positions were initialized by sampling from a uniform random distribution in every dimension, so that $x_{i,j}^0 \sim U(L, U)$. Particles' personal best positions were initialized in the same manner, then both the current position and personal best position were evaluated. If the current position had a better score than the personal best position, the two were swapped, so that the semantic meaning of the personal best position is preserved.

To ensure that the solution found by the swarm is within the search space, the global best and personal best positions were only updated if they were within the search space. Thus, both particle attractors were always within the search space, as suggested by [7, 15].

Three different γ values were tested, specifically $\gamma \in \{1.0, 0.75, 0.5\}$. Under the assumptions in section 3.1, when $\gamma = 0.5$, approximately 95% of the swarm should be within the search space even when the personal best and global best are maximally far apart. The swarms with $\gamma = 0.5$ are thus expected to exhibit less roaming. Section 3.1 assumes swarm stagnation and the strategy of restricting variance intends to prevent premature stagnation due to roaming. Thus, the assumptions in section 3.1 will likely not hold throughout the search and very restrictive values for γ may not be the best, so $\gamma = 0.75$ is also tested; $\gamma = 1.0$ is used as a baseline.

For each γ, a range of w-values from 0.1 to 1.0 were tested, so that all the configurations would guarantee swarm convergence according to [30] (except $w = 1.0$, which was included as a baseline). The tested inertia weight values are given in the expression below:

$$w \in \{0.1 + 0.05k \mid 0 \leq k \leq 18\} \tag{15}$$

For the chosen (γ, w) pair, the corresponding value for the acceleration coefficient was calculated according to equation (11), except for $w = 1$, where the corresponding c values were not unique (i.e. regardless of γ, if $w = 1$ then $c = 0$). Table 1 summarizes all the c-w combinations that were tested and the corresponding γ values.

The benchmark functions from the CEC'2010 special session and competition on large-scale optimization were used [36] with $n = 1000$ (which is a sufficiently high dimensionality for PSO to exhibit dimensionality-related roaming, but low enough for simulations to complete within reasonable time scales). The benchmark suite allows the degree of separability within certain functions to be specified using the parameter m. In order to scale the problems down to 10 dimensions for later experiments in section 4.2.1, the experiment used a value of $m = 10$. The benchmark suite includes separable functions, partially separable functions, and non-separable functions. The search spaces for all the benchmark problems had the same upper and lower bounds in every dimension, denoted by U and L respectively. These values correspond to the specifications by [36]. Additional details regarding the benchmark suite are provided in appendix A.

Each swarm configuration was tested on each of the benchmark functions by running 30 independent simulations to achieve statistical significance. The

Table 1: Parameter configurations for swarms with fractionally restricted standard deviations

$\gamma = 1.0$		$\gamma = 0.75$		$\gamma = 0.5$	
w	c	w	c	w	c
0.1	1.7535	0.1	1.6998	0.1	1.5632
0.15	1.7943	0.15	1.7349	0.15	1.5851
0.2	1.8286	0.2	1.7633	0.2	1.6
0.25	1.8557	0.25	1.7841	0.25	1.6071
0.3	1.8747	0.35	1.7967	0.3	1.6059
0.35	1.8846	0.35	1.800	0.35	1.5955
0.4	1.8841	0.4	1.7929	0.4	1.575
0.45	1.8719	0.45	1.774	0.45	1.5435
0.5	1.8462	0.5	1.7419	0.5	1.5
0.55	1.8049	0.55	1.6947	0.55	1.4431
0.6	1.7455	0.6	1.6302	0.6	1.3714
0.65	1.6649	0.65	1.5457	0.65	1.2833
0.7	1.5592	0.7	1.4381	0.7	1.1769
0.75	1.4237	0.75	1.3034	0.75	1.05
0.8	1.2522	0.8	1.1368	0.8	0.9
0.85	1.0366	0.85	0.9322	0.85	0.7239
0.9	0.7664	0.9	0.6817	0.9	0.5182
0.95	0.4274	0.95	0.3754	0.95	0.2786
1.0	0.0	1.0	0.0	1.0	0.0

performance of an algorithm on a given benchmark function was characterized in terms of the best objective function value attained in a given simulation. For all of the benchmark functions, the best possible objective function value is 0.

In addition to the best possible objective function value, other metrics such as the swarm diversity and average velocity magnitude were also measured. Swarm diversity characterizes the spread of the swarm and may be used to illustrate exploration and exploitation behaviour. Different measures for swarm diversity have been suggested such as the swarm diameter and radius [31], the average distance around the swarm centre [25], and the normalized average distance around the swarm centre [31]. Olorunda and Engelbrecht found that the average distance from the swarm centre provides a good compromise between accuracy, robustness and computational efficiency [28], so the average distance from the swarm centre was used to measure diversity. The swarm diversity is given by

$$\mathcal{D} = \frac{1}{n_s} \sum_{i=1}^{n_s} \sqrt{\sum_{j=1}^{n} (x_{i,j} - \hat{x}_j)^2} \tag{16}$$

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

$$\hat{\mathbf{x}} = \frac{1}{n_s} \sum_{i=1}^{n_s} \mathbf{x}_i \tag{17}$$

The average particle velocity magnitude is given by

$$\mathcal{V} = \frac{1}{n_s} \sum_{i=1}^{n_s} \sqrt{\sum_{j=1}^{n} v_{i,j}^2} \tag{18}$$

The performance of different swarm configurations were compared by assigning each configuration a score relative to all the other configurations, as described by [4]. Generally, a configuration's score is calculated in terms the number of "wins" achieved over all the other configurations, across all of the benchmark functions. A configuration's "win" over another on a particular function is determined by comparing the median best objective function value of the two configurations using a Mann-Whitney U test with $p = 0.05$.

Let the problem dimension n be fixed. Let two different algorithm configurations (i.e. with different values for γ and w) be denoted by g and h. For every (g, h) pair and function f, define $s_{g,h,f}$ as follows

$$s_{g,h,f} = \begin{cases} 1 \text{ if } R_{g,f} < R_{h,f} \text{ with } p \leq 0.05 \\ 0 \text{ if } p \geq 0.05 \\ -1 \text{ if } R_{h,f} < R_{g,f} \text{ with } p \leq 0.05 \end{cases} \tag{19}$$

where $R_{g,f}$ denotes the median of the best score attained by configuration g when optimizing function f and p denotes the confidence bound of the Mann-Whitney U test. For every configuration pair (g, h), the "wins" of g over h is measured in terms of a point system. This measure is denoted by $z_{g,h}$ and is calculated by

$$z_{g,h} = \begin{cases} 3 \text{ if } \sum_{f=1}^{F} s_{g,h,f} > 0 \\ 1 \text{ if } |\sum_{f=1}^{F} s_{g,h,f}| = 0 \\ 0 \text{ if } \sum_{f=1}^{F} s_{g,h,f} < 0 \end{cases} \tag{20}$$

where F denotes the number of functions in the benchmark suite. For the benchmark suite in appendix A, $F = 20$.

The total "score" of a configuration g is the sum of its wins over all the other configurations. The total score, M_g, of configuration g is denoted by

$$M_g = \sum_{h=1;h\neq g}^{J} z_{g,h} \tag{21}$$

where J denotes the number of configurations. For these experiments, $J = 3 \times 18 + 1 = 55$ (three different γ-values, each with 18 different w-values, plus the configuration where $w = 1$). If $M_g > M_h$, then configuration g performed better on the benchmark suite than configuration h.

3.3. Variance and Dimensionality

Restricting the standard deviation influenced the swarm's average velocity magnitude, as expected. It may have been that the swarm was not roaming due to the distance between the particles' personal and global best positions, but rather because the swarm was attracted to a position near the boundary of the search space. Then the particles exploring the region near the global best may have been likely to be out of bounds in at least a few dimensions, due to the positioning of the attractor, even though the particles' velocities were not very high. If this was the case, then restricting the variance would not reduce the swarm's average velocity.

Figures 3 to 7 illustrate the typical profile of the swarms' average velocity magnitude. Each figure is for a fixed inertia weight and shows the effect of varying γ. In all cases, the smaller standard deviation resulted in lower average velocity, as expected. Unless otherwise stated, all figures illustrate typical swarm behaviour that was observed across all benchmark functions. The benchmark functions chosen for illustration purposes were not selected preferentially.

Figures 8 to 10 show the average velocity magnitude for a fixed γ as w varies. For each γ, lower inertia weights result in higher average velocities and vice versa. As γ becomes larger, the diversity increases asymptotically as w decreases. Thus, the movement of swarms with higher inertia weights is generally more restrained than swarms with lower inertia weights.

Restrictions to the standard deviation combined with high w-values cause the corresponding acceleration coefficients to have relatively low values. Such swarms will rely more heavily on momentum than on new information received from its neighbours, or learned throughout its own search. These particles will exhibit smooth trajectories due to the regularizing influence of the large momentum component and because any direction alterations introduced by their personal and global best positions will be small, since the acceleration coefficients are small.

If the inertia weight is low and the acceleration coefficients are high, then the particle is more likely to divert its course if its attractors are updated. If the location of the particle's attractors changes drastically, from one side of the search space to the other, this may lead to the particle taking huge steps, causing velocity explosion. Though this is true for low dimensional problems,

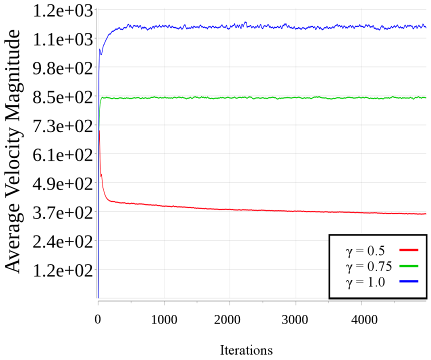

Figure 3: Average velocity magnitude for $w = 0.95$ on F7 with $n = 1000$

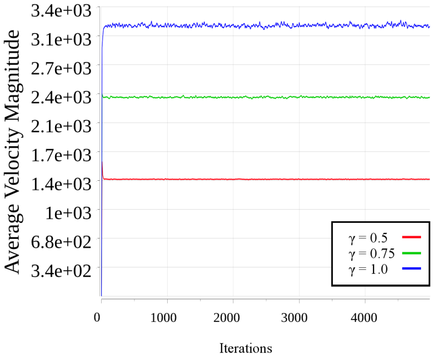

Figure 4: Average velocity magnitude for $w = 0.75$ on F7 with $n = 1000$

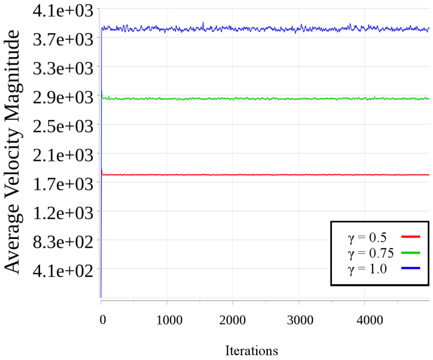

Figure 5: Average velocity magnitude for $w = 0.55$ on F7 with $n = 1000$

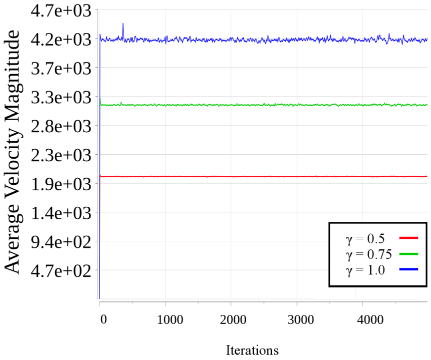

Figure 6: Average velocity magnitude for $w = 0.35$ on F7 with $n = 1000$

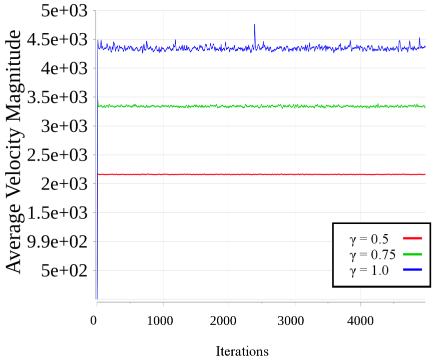

Figure 7: Average velocity magnitude for $w = 0.15$ on F7 with $n = 1000$

the effects are exacerbated in high dimensions.

Figure 11 shows the score of a swarm with a given γ and w on the 1000 dimensional problem suites. The color of a block shows its score, with lighter indicating a better score. Comparisons were done across all combinations of γ and w, so that the overall best combinations can be determined. Figure 12 visualizes the performance of all the c-w combinations. For every parameter configuration, the corresponding score of the configuration is reflected in the color of the data point, where pink is the best possible score and blue is the worst score.

Table 2: Coefficient values for best-performing combinations

γ	w	c	Score
0.5	0.95	0.2786	162
0.5	0.9	0.5182	159
1.0	0.95	0.4274	156
0.75	0.95	0.3754	153
0.75	0.9	0.6817	150

Table 3: Coefficient values for worst-performing combinations

γ	w	c	Score
1.0	1.0	0.00	23
0.5	0.4	1.575	25
1.0	0.5	1.8462	25
0.5	0.65	1.2833	26
1.0	0.7	1.5592	28

For completeness, the mean and standard deviation of the best fitness for the three best and worst configurations are provided in tables 14 and 15 in the appendix. The w, c and score of the five best combinations are given in table 2. All five of the best combinations had high inertia weights and relatively low acceleration coefficients. In general, configurations with high inertia weights performed the best and values for w below 0.75 generally did not perform well in comparison to the others. Configurations with severe restrictions to the standard deviation (i.e. lower γ) performed well, but the best performing swarms varied across all three chosen γ.

High inertia weights reduce the danger of a particle taking huge steps from one end of the search space (due to the particle's attractors being updated) because the influence of the social and cognitive components is small. Instead,

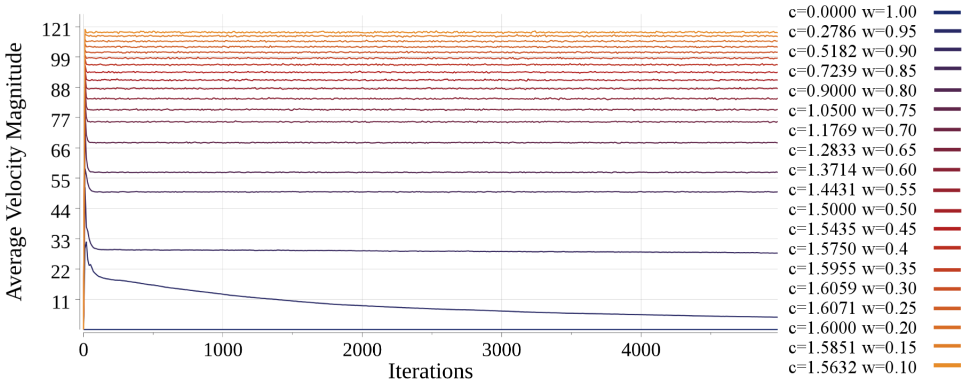

Figure 8: Average velocity magnitude for $\gamma = 0.5$ on F2 with $n = 1000$

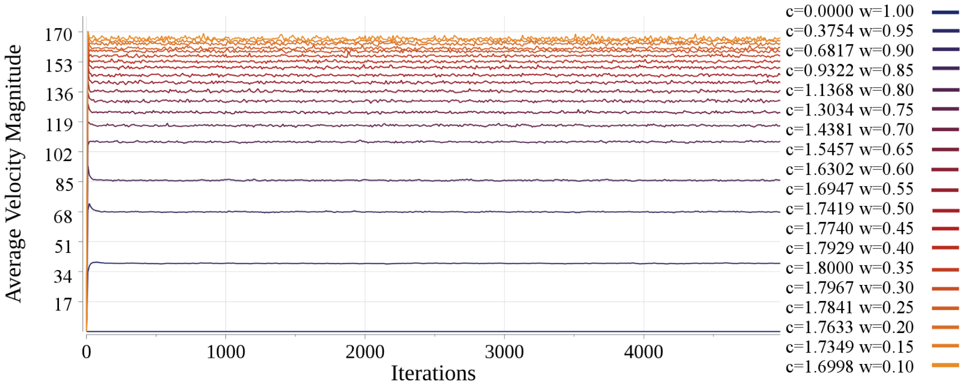

Figure 9: Average velocity magnitude for $\gamma = 0.75$ on F2 with $n = 1000$

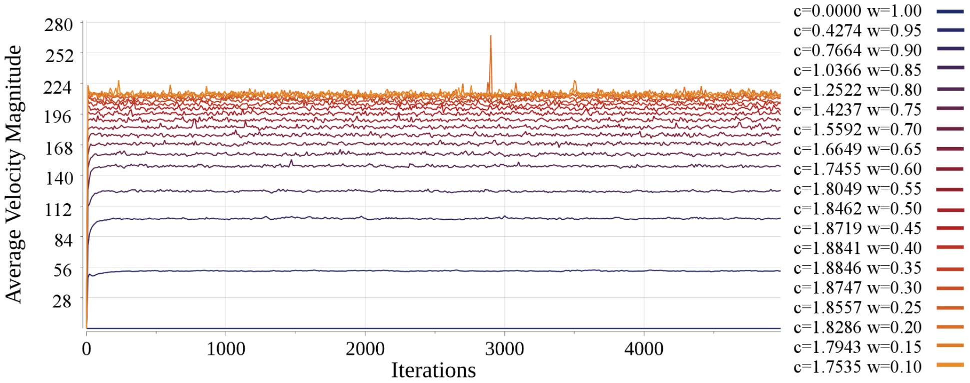

Figure 10: Average velocity magnitude for $\gamma = 1.0$ on F2 with $n = 1000$

the particle's trajectory may be pulled towards the updated attractors gradually, without fear of velocity explosion (as discussed previously, based on figures 8 to 10).

The w, c and score of the five worst combinations are given in table 3. The worst performing swarms were either $\gamma = 1$ or $\gamma = 0.5$. The worst configuration used only inertia and had no social or cognitive component whatsoever. It is not surprising that the swarm did not perform well, since the particles could not incorporate any additional information into their search direction. The search will devolve into sampling a number of points along a line, where the line's direction was determined by the randomly initialized personal and global best positions. The other configurations that scored in the bottom five all had large acceleration coefficients and medium-range inertia weights. Disconcertingly, some of these values were not far away from the rule of thumb "good" parameters.

Simply restricting the standard deviation is thus not enough to ensure good performance: some of the worst-performers had severely restricted standard deviations. The swarm's performance is heavily dependent on the chosen values for w and c. This is not unexpected: just because a swarm's movement is restricted to be within the search space does not guarantee that the swarm will find a good solution within the search space.

Figure 13 shows the typical diversity profiles of the five best-performing configurations. In general, the configurations with better scores exhibited lower diversities. It is interesting to note that, although the chosen value for w and c plays a larger role in determining the swarm's behaviour, the role of γ is still visible here, even when comparing across different inertia weights: the swarms for which $\gamma = 1$ showed increasing diversity, in comparison to the configurations for which $\gamma < 1$, where the swarm diversity decreased or, at least, did not increase after the initial spike.

In general, larger γ values exhibited higher diversities and more dimensions out of bounds (see figures 13 and 14), although large values for c also played a role. For example, when $\gamma = 1$ and $c = 0.4274$, the diversity was lower than for $\gamma = 0.75$ and $c = 0.6817$. When c is larger, the local and global attrac-

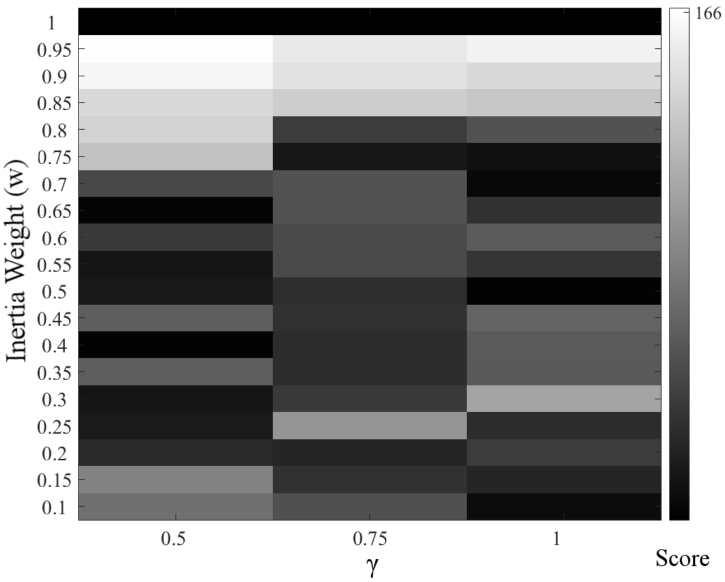

Figure 11: Performance of swarm configuration produced by restricting standard deviation of positions

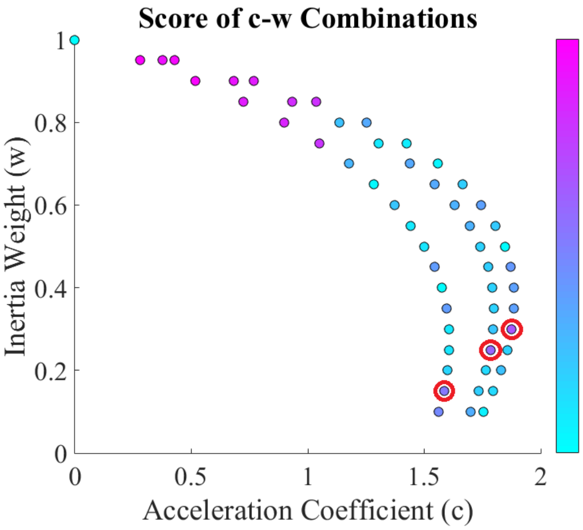

Figure 12: Score of acceleration coefficients and inertia weights produced by restricting standard deviation of positions

tors exert more influence over the particle trajectories, allowing the particles to be diverted towards updated attractors and increasing the swarm's diversity. Usually, none of the configurations' diversities went to zero, indicating that the particles did not converge to a single point, even for the configurations with very low acceleration coefficients.

Although restricting the variance reduced the average velocity magnitude and the average number of dimensions out of bounds, the average number of particles that were outside the search space was still very high, even for the five configurations with the best performance, as shown in figure 15. The configurations where $\gamma = 0.5$ exhibited less roaming than the other configurations, but there is no discernible difference in roaming behaviour between the configurations with $\gamma = 0.75$ and $\gamma = 1.0$. Future experiments may consider variance restrictions that are even more stringent, but in general, restriction of variance alone does not appear to mitigate particle roaming completely. Instead, investigating relationships between w and c may prove more fruitful, as will be shown in the remainder of the paper.

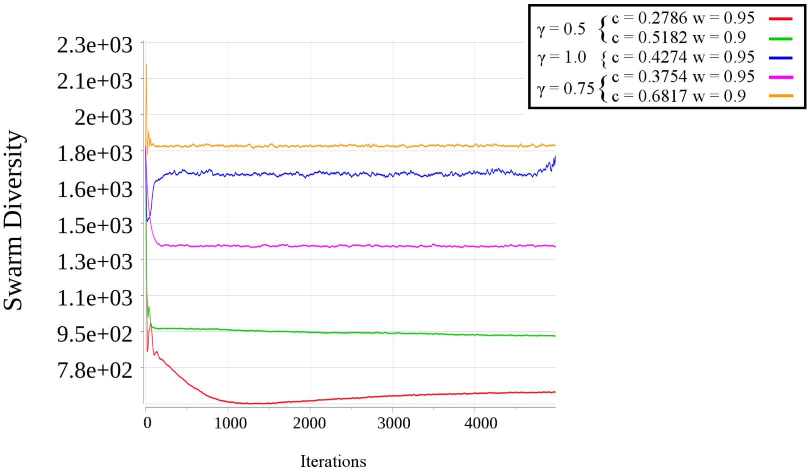

Figure 13: Swarm diversity on F18 with $n = 1000$ (5 best configurations)

From figures 11 and 12, there appear to be three points with low inertia

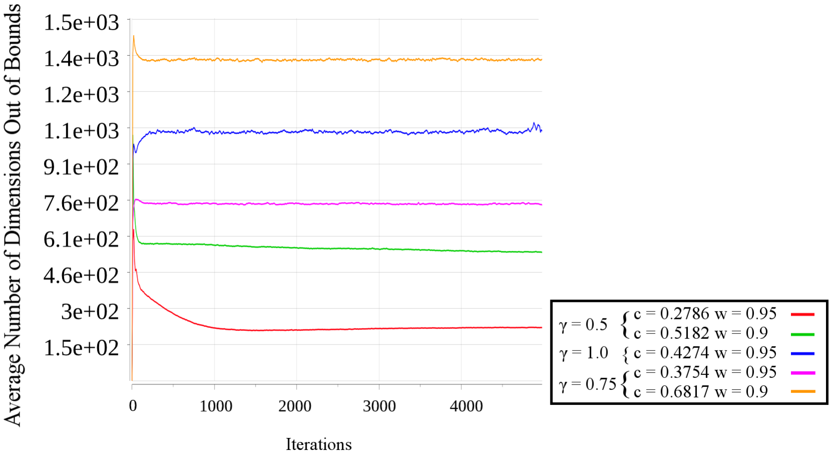

Figure 14: Average number of dimensions out of bounds on F18 with $n = 1000$ (5 best configurations)

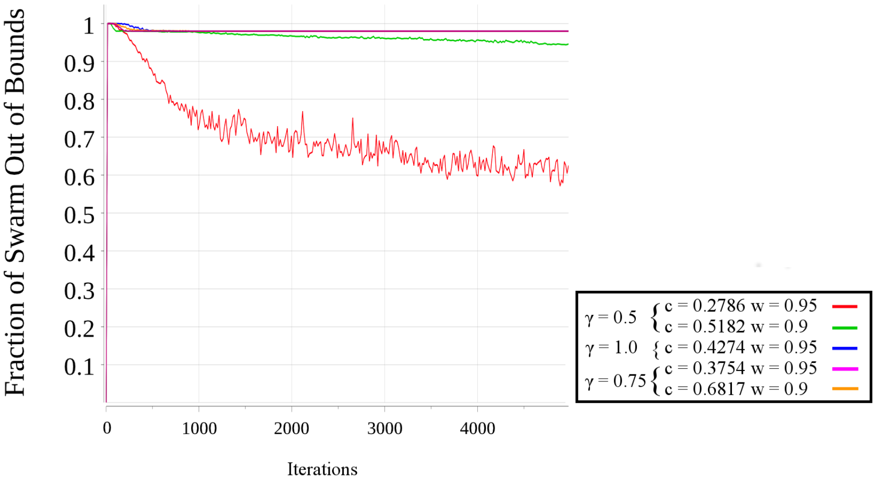

Figure 15: Fraction of swarm out of bounds on F18 with $n = 1000$ (5 best configurations)

weights and high acceleration coefficients that also perform well (circled in red on figure 12). However, further investigation of these configurations yielded nothing interesting about these configurations. In general, the swarms were out of bounds for almost the entire search and their personal best positions were almost never updated. The behaviour of these three points did not seem very different from their surrounding neighbours, as can be seen in figures 16 and 17. Figure 16 plots the diversities of the three outlier points and figure 17 includes the diversities of the outlier's surrounding neighbours for comparison. Since the calculated scores are relative, it may simply be that these points scored slightly higher than the other parameter configurations in the same vicinity, though their performance was poor.

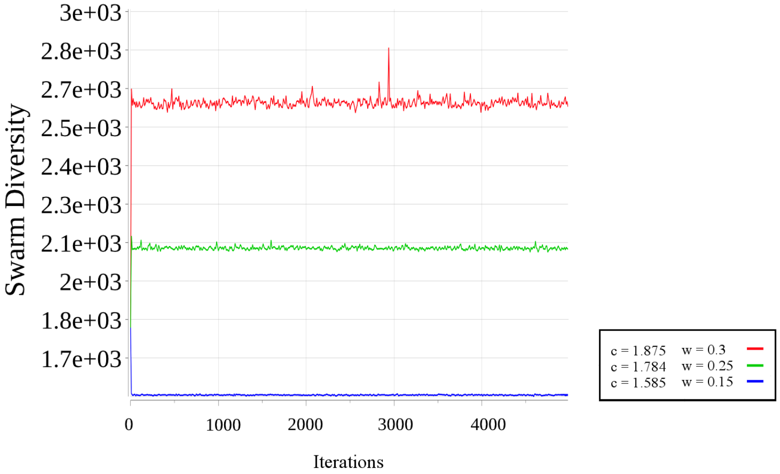

Figure 16: Swarm diversity of outlier configurations on F9 with n=1000

3.4. Summary

Particle movement can be restricted by choosing an inertia weight and acceleration coefficients to reduce the variance of particle positions. For a given inertia weight, decreasing γ caused the swarm to exhibit lower velocity magnitudes, thereby reducing the step sizes of the particles and encouraging a more

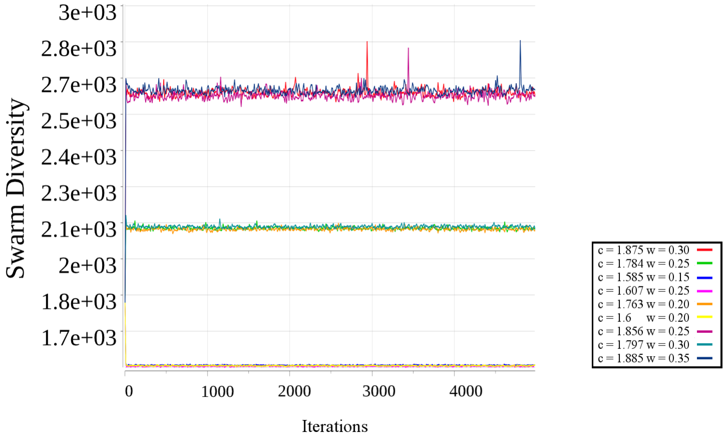

Figure 17: Swarm diversity of outlier configurations and surrounding configurations on F9 with n=1000

granular search. For a fixed γ, increasing the inertia weight caused the swarm's average velocity magnitude to decrease and vice versa. High inertia weights regularize the particle's movement because the particle is less likely to divert its course rapidly if the positions of its attractors changes.

Thus, if the particle's attractors change drastically from one iteration to the next, a momentum-focused particle will be less prone to oscillation and velocity explosion than a particle that has high acceleration components. However, restricting the variance is not sufficient to mitigate particle roaming. High inertia weights and low acceleration coefficients also play an important role.

The analysis performed in this section was based on empirical observations regarding the reasons for particle roaming. The section that follows makes use of literature introduced in section 2.3, regarding the swarm's movement patterns, to provide additional insight into the behaviour of PSO in high dimensional spaces.

4. Frequency and Variance of Particle Positions

This section is intended to provide further insight into the results of the previous section. The inertia weights and acceleration coefficients that performed well previously may correspond to certain classes of movement patterns, making it easier to interpret the swarm's behaviour. Section 4.1 describes the settings of the experiment, which examines the influence of the base frequency and range of movement on the swarm's behaviour in high dimensional spaces. Section 4.2 examines the results of the empirical experiment and determines what swarm configurations and corresponding movement patterns are optimal in high dimensional spaces in terms of performance. Section 4.3 summarizes the findings of this section.

4.1. Experimental Method

The relationship between movement patterns and problem dimensionality was investigated empirically. This section describes the experimental methodology that was followed.

The experiment was performed as in section 3.2, but the values for w, c_1 and c_2 were calculated for a given base frequency and variance of movement. The experiment tested different values for the frequency and variance of movement with

$$F \in \{0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.25, 0.4, 0.45\} \quad (22)$$

$$V_c \in \{0.1, 0.4, 1.6, 6.4, 25.6\} \quad (23)$$

which are the same as the values tested by [4] to allow for comparison. For each (F, V_c) pair, the corresponding values for the acceleration coefficients and inertia weight were calculated using equations (5) and (6). Setting (5) = (6) and solving for w yields a polynomial of order 4, which was solved using Matlab's *roots* function. The root with the smallest absolute value and no imaginary component was chosen as the w value. The w-value was then substituted into both equations (5) and (6) to calculate c. Due to the intrinsic error in calculating

w, equations (5) and (6) yield slightly different values for c. The average of the two values was used as c to compensate for any error in w.

To ensure that the results remain interpretable and are not skewed by the errors introduced when solving for w, the resulting w and c values were substituted back into equations (5) and (6) to solve for the frequency and variance. The recalculated frequency and variance, denoted by F' and V_c', capture the error on w and c. The resulting error between the desired frequency and variance (F, V_c), and the actual frequency and variance (F',V_c') can then be calculated. The error on the frequency, E_F, was calculated as

$$E_F = |F - F'| \tag{24}$$

The frequency error was never larger than 0.005, i.e. the error on the frequency was never larger than 10% of the frequency increment of 0.05. The error in frequency was thus deemed sufficiently negligible.

The relative error on the variance, E_{V_c}, was calculated using

$$E_{V_c} = \frac{|V_c - V_c'|}{V_c} \tag{25}$$

so that the error is normalized according to the size of the variance, since the tested variances had quite a large range. Figure 18 shows E_{V_c} for each F. Generally, the relative error increased as the frequency increased gradually until $F = 0.3$, then decreased again. For $V_c = 0.1$, the relative error became as high as 62%. The relative error of the other variances were lower and never went higher than 16%. The results for the following (V_c, F) pairs were deemed unacceptable due to high relative errors:

$$\{(0.1, 0.25), (0.1, 0.3), (0.1, 0.35)\} \tag{26}$$

The performance of swarms using the (V_c, F) pairs listed above were not considered in the analysis that follows. The calculation error of the remaining pairs was deemed sufficiently small to use for analysis. Future work may include finding a different formulation for these calculations that can be solved analytically or more readily by numerical methods.

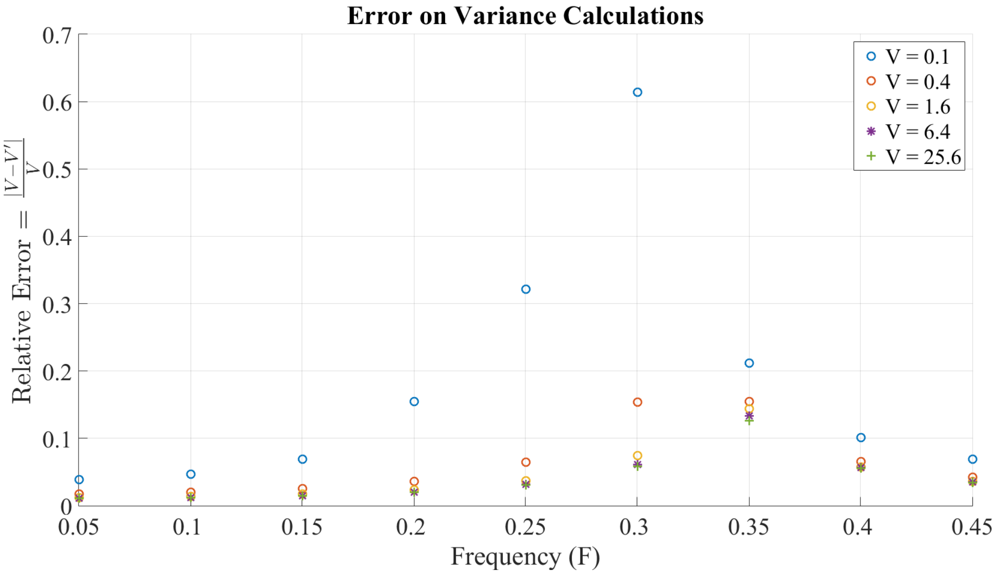

Figure 18: Relative error on variance across frequencies

4.2. Movement Patterns in High Dimensions

This section presents the results of the experiments described in the previous section. Section 4.2.1 examines all the tested swarm configurations and determines which configuration performed the best in high dimensional spaces. The behaviour of the best swarm configurations are examined in detail and possible reasons for their success are presented. Section 4.2.2 reconciles the results of the experiments in variance restriction (from section 3) with the empirical results regarding movement patterns in this section.

4.2.1. Optimal Frequency and Variance of Movement

This section determines the optimal combinations for F and V_c for low and high dimensional versions of the problem suite. From the frequency and variance values, which determine corresponding c and w values according to equations (5) and (6), it is possible to determine what kind of movement patterns are exhibited by the swarms that perform well. The section also shows that the optimal movement pattern for the swarm is different for high and low dimensional spaces.

For completeness, the mean and standard deviation of the best fitness for the three best and worst configurations are provided in tables 16 and 17 in the appendix. Figures 19 and 20 show the score of all the (F, V_c) pairs on the 10 and 1000 dimensional problem suites. The color of a block shows its score, with lighter indicating a better score. Comparisons were done across all combinations of F and V_c, so that the overall best combinations could be determined for each problem suite.

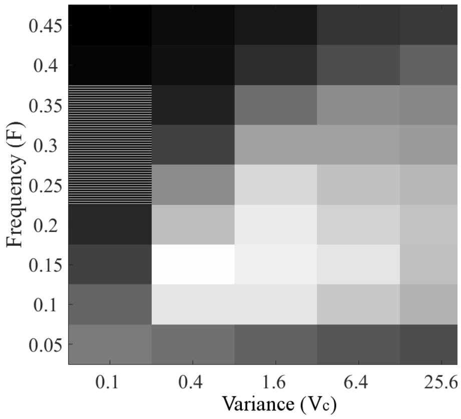

Figure 19: Optimal frequency-variance combinations (n=10)

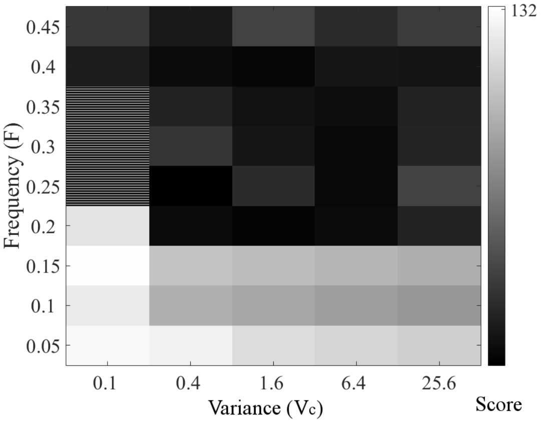

Figure 20: Optimal frequency-variance combinations (n=1000)

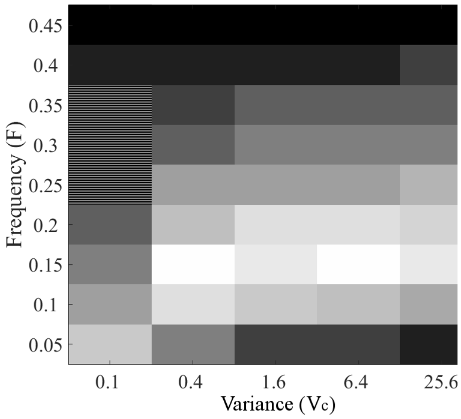

Figure 21: Optimal frequency for given variance (n=10)

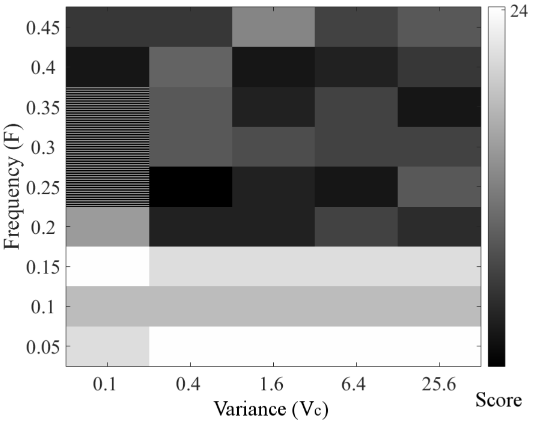

Figure 22: Optimal frequency for given variance (n=1000)

The frequency-range combinations that perform well are different between the low and high dimensional suites. In low dimensions, the frequency $F = 0.05$ never performed well and generally, values $F \in [0.1, 0.2]$ provided good performance. Thus, swarms exhibiting smooth trajectories with some positive correlation between particle positions performed well. Parameters causing very strong correlation between particle positions did not perform well.Thus, some chaotic behaviour is beneficial to the swarm's ability to find good solutions.

For high dimensional problems, the swarms exhibiting oscillatory behaviour performed much worse than swarms with smooth trajectories ($F \geq 0.25$ performed badly in comparison with $F < 0.25$). Swarms performance deteriorates as the variance increases. Of the five best configurations, four had the smallest possible variance of 0.1 and one had a variance of 0.4. All five of the best performing configurations had frequencies below 0.25. The frequency and variance as well as the corresponding momentum and acceleration coefficients are given in table 4.

Thus particles with low range of movement and smooth trajectories showed the best performance. For the lowest possible variance of 0.1, slightly higher frequencies (where position correlation is lower) performed better whereas for the other variances, the lowest possible frequency performed the best. This indicates that, when the particles' range of movement is sufficiently small, some chaos in movement may be beneficial; the small movement range is sufficient to restrict particle movement and small chaotic steps do not cause a velocity explosion. But as particle steps become larger, chaotic movement is detrimental to swarm performance and instead, highly correlated particle positions are required to prevent particles from roaming.

Observations regarding the good performance of small frequencies agree with previous literature. The observed benefit of using small variance differs from previous findings that were focused on low dimensional problems. However, this is in line with the findings from the previous section and with the hypothesis that fine-grained searching within small areas of the search space are the most effective in solving high dimensional problems.

Four of the five top combinations had $V_c = 0.1$, which does not satisfy the relation in equation (7):

$$\left(\frac{\max\{c_1, c_2\}}{c_1 + c_2}\right)^2 = \left(\frac{c}{2c}\right)^2 = \frac{1}{4} > V_c = 0.1 \tag{27}$$

Thus, there was no guarantee that the particles sampled any points outside the region between their personal and global best positions. This does not imply that no points outside this region were sampled, but it does point towards locally exploitative behaviour by the particles that performed well.

Figures 21 and 22 show the optimal F-value for a given V_c. Comparisons were thus done within each column (as opposed to over all the possible pairs, as done in figures 19 and 20). On the low dimensional suite, the optimal frequency was usually around 0.15, which produced smooth particle trajectories, but particle positions are not highly correlated.

For the high dimensional case, the optimal frequency was almost always 0.05, the smallest possible value. Similar to the low dimensional case, $V_c = 0.1$ was the exception. For all values of V_c, the optimal frequency was smaller on the high dimensional problems than it was on the low dimensional problems. Thus, highly correlated particle positions and granular searching behaviour is a requirement for good performance in high dimensional spaces. Additionally, observe that, unlike the low dimensional case, none of the F-values that induce oscillatory behaviour performed well in high dimensions. This further supports the hypothesis that fine-grained exploitation performs better than exploration-focused behaviour in high dimensional spaces.

Tables 4 and 5 show the acceleration coefficient and inertia weight corresponding to the best and worst (F, V_c) combinations. In addition, figure 23 plots the c and w values that correspond to the (F, V_c) combinations tested in the experiments. The colour of each data point corresponds to its score in comparison to all the other swarm configurations, where pink dots have the highest (best) score and light blue dots have the lowest (worst) score.

The best performing swarms generally had high momentum components and relatively low acceleration coefficients. This corresponds to the findings in figure

12, but the additional information provided by the corresponding base frequency and variance of movement has aided the discussion regarding the reasons why these swarms performed well. Configuration A's values were the most different, with nearly equal values for the inertia weight and acceleration coefficient, so that the acceleration coefficient's value was still significantly lower than the rule of thumb value. Figure 23 shows no configurations that perform well in the region $c \in (0.58, 1.88), w \in (0.15, 0.3)$, which supports the conclusion that the three outliers in figure 12 are not significantly better than their neighbours.

Table 4: Coefficient values for best-performing combinations

V_c	F	w	c	Score	Reference
0.1	0.15	0.70521	0.74883	132	Config A
0.1	0.05	0.96941	0.099381	129	Config B
0.4	0.05	0.98631	0.097915	126	Config C
0.1	0.1	0.87483	0.37326	123	Config D
0.1	0.2	0.40746	1.0965	120	Config E

Table 5: Coefficient values for worst-performing combinations

V_c	F	w	c	Score
0.4	0.25	0.55579	1.5763	21
1.6	0.2	0.81365	1.2588	23
1.6	0.4	0.15443	1.8091	24
6.4	0.25	0.70415	1.7052	25
6.4	0.3	0.51717	1.9633	25

Figures 24 and 25 show typical diversity profiles of the five best performing swarm configurations. The configuration with the largest variance consistently exhibited higher diversity than the other configuration, which is expected given that the swarm has larger range of movement. The configuration with $V_c = 0.4$ never converged to a point, i.e. the swarm's diversity never went to zero. In

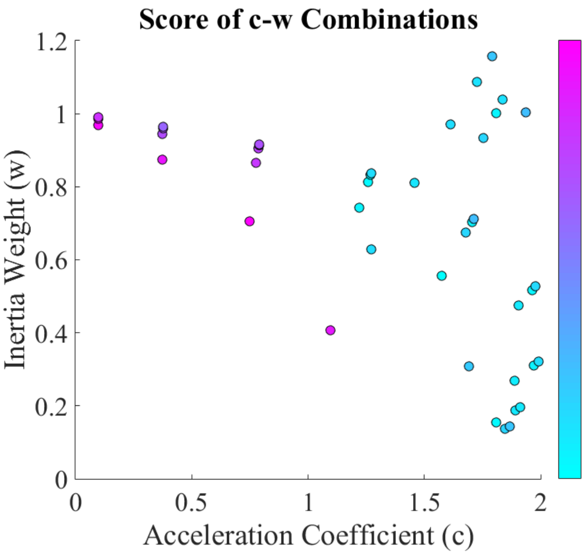

Figure 23: Score of acceleration coefficients and inertia weights produced by frequency and variance

contrast, all four of the other top performing swarms (all with $V_c = 0.1$) converged very close to a single point. The diversity of the best performing strategy, $V_c = 0.1$ and $F = 0.15$ decreased more slowly than the other strategies with $V_c = 0.1$, while the rest converged prematurely within the first 500 iterations. The c and w of the best configuration were nearly equal, indicating that the particles struck a nearly equal balance between restricting movement to the current trajectory and moving in the direction of good solutions.

Figures 26 and 27 both plot the average number of personal best updates per iteration for configurations A to E. This measure indicates whether the swarm is actively searching and improving its solutions or the swarm has stagnated. These figures varied greatly between benchmark functions, which is to be expected since the measure is highly problem dependent. Generally, configuration A consistently showed improvements throughout the search, with the number of personal best position updates decreasing towards the end of the search. Such behaviour adheres to the usual recipe of first exploring and then exploiting, and fits well with the swarm's diversity profiles.

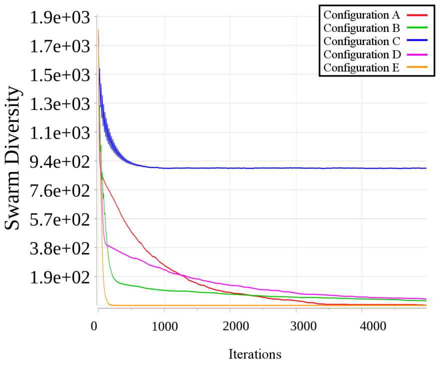

Figure 24: Swarm diversity on F7 with $n =$ 1000 (best 5 configurations)

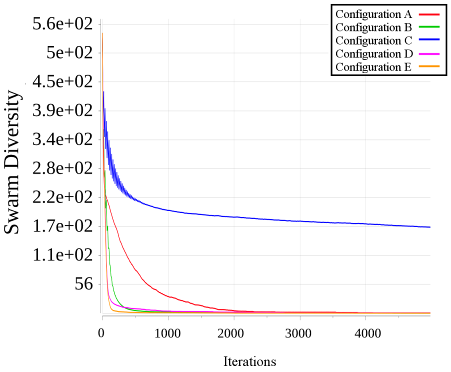

Figure 25: Swarm diversity on F11 with $n =$ 1000 (best 5 configurations)

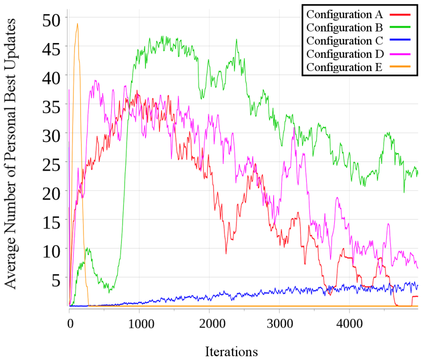

Figure 26: Average number of personal best updates on F7 with $n = 1000$ (best 5 configurations)

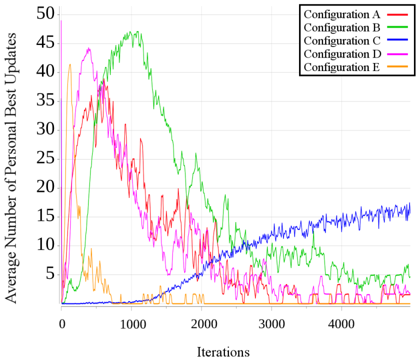

Figure 27: Average number of personal best updates on F11 with $n = 1000$ (best 5 configurations)

Although configuration B seemed to have converged according to the diversity plot, the swarm was usually actively improving for most of the search, 675 often displaying higher update counts than configuration A, especially later in the search. The swarm's personal best position update counts increased just before its diversity dropped, implying that the swarm was exhibiting exploitative behaviour.

Configuration C usually exhibited improvement throughout the search, with 680 update counts increasing as the search progressed, showing that the swarm was still exploring. Configuration C's low acceleration coefficients may explain why the swarm required so many iterations to begin exploiting: since the influence of a particle's attractors was so low, it may take many iterations for a particle's trajectory to move in the direction of its updated attractors.

Configuration D's behaviour varied, sometimes peaking early in the search and then converging prematurely. Other times, configuration D continued to improve throughout the search with update counts just below that of configuration B. Lastly, configuration E's update counts consistently peaked within the first 500 iterations and then went almost to zero, indicating premature convergence.

Of the five most successful strategies, most prevented particles from roaming outside the search space. Configuration C, with the largest variance, occasionally failed to return more than 10% of the swarm's particles to the search space (see figure 28). But in general, almost all of the particles returned to the search space throughout the search (see figures 29 and 30).

From table 4, it is apparent that particle movement can be restricted successfully by having high inertia weights and low acceleration coefficients. Configuration E is the exception, with a low inertia weight and a relatively large acceleration coefficient (though still smaller than the values that are generally accepted as "rule of thumb").

4.2.2. Movement Patterns with Restricted Variance

The previous two sections discussed the experiments where the base frequency and variance of particle positions were varied in order to find optimal movement patterns for high dimensional problems. This section considers the configurations tested in section 3 which were obtained by restricting the standard deviation of particle positions and then calculating the corresponding values for c and w. This section calculates the corresponding base frequency and variance for each of the configurations obtained in section 3. The purpose of this is two-fold: first, the corresponding base frequency will reveal the nature of particle trajectories, thereby explaining why configurations with the same standard deviation are able to perform differently. Secondly, in accordance with the

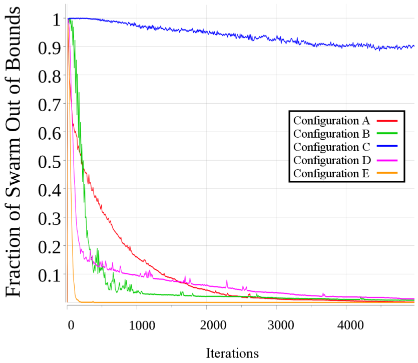

Figure 28: Fraction of swarm out of bounds on F8 with $n = 1000$ (best 5 configurations)

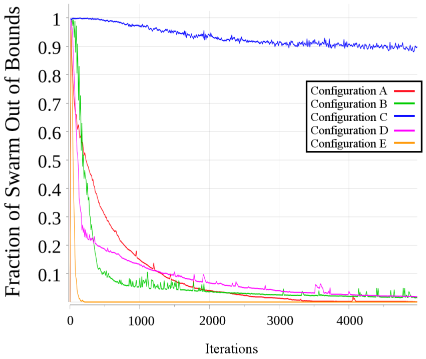

Figure 29: Fraction of of swarm out of bounds on F7 with $n = 1000$ (best 5 configurations)

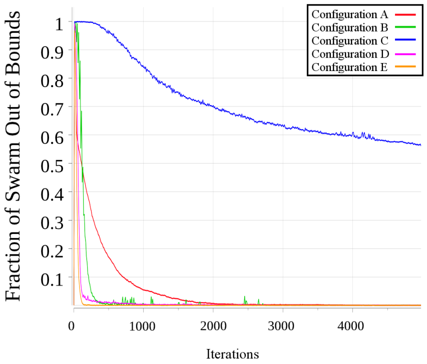

Figure 30: Fraction of of swarm out of bounds on F11 with $n = 1000$ (best 5 configurations)

findings in this section, it is expected that the corresponding base frequency and variance of the configurations that performed well should yield smooth particle trajectories. If so, then the findings of this section have been confirmed via an independent experiment.

715 Figure 31 is a scatter plot, showing the corresponding base frequency (F) and variance (V_c) for each of the configurations in table 1. The colour of a data point indicates its score, where pink data points correspond to high (good) scores and blue data points correspond to low (bad) scores.

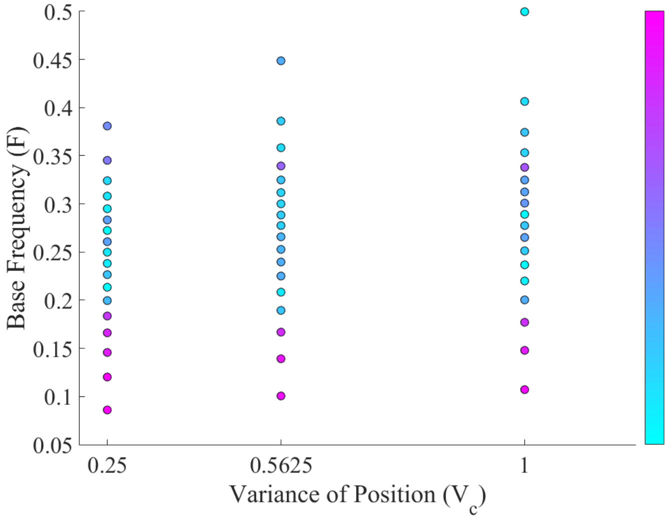

Figure 31: Performance and Movement Parameters of Configurations with Restricted Variance

The corresponding base frequencies of the configurations exhibiting good performance were low, implying that their trajectories were smooth with strongly correlated positions. This confirms previous statements regarding the regularizing effect that high inertia weight has on a particle's trajectory.

The configurations exhibiting poor performance had high base frequencies, implying that their trajectories were erratic and their positions were weakly correlated or independent. This confirms statements regarding the dangers of high acceleration components, which enable rapid oscillation across the search space when the location of the attractors change.

As can be seen from table 6, the best performing configurations fall within the regions predicted by figure 20 to perform well (i.e. where the base frequency is near 0.15). Similarly, according to table 7, the configurations that exhibited the worst performance were within the regions predicted by figure 20 to perform very poorly (i.e. where $F \in [0.2, 0.25]$). Thus, the results from section 3 align with the observations in this section. Note that for all the configurations in tables 6 and 7, the variance fell somewhere in the first three columns of figure 20. All three of these variance values were very low, which may explain why the influence of γ on search behaviour was less pronounced than the influence of the relationship between c and w. The relationship between the acceleration coefficient and the inertia weight is captured to a large extent by the base frequency.

Table 6: Movement parameters for best-performing combinations

c	w	F	V_c
0.2786	0.95	0.0860	0.25
0.5182	0.9	0.1202	0.25
0.4274	0.95	0.1073	1.0
0.3754	0.95	0.1003	0.5625
0.6817	0.9	0.1390	0.5625

Table 7: Movement parameters for worst-performing combinations

c	w	F	V_c
0.0	1.0	0.0	∞
1.575	0.4	0.2721	0.25
1.8462	0.5	0.2894	1.0
1.2833	0.65	0.2135	0.25
1.5592	0.7	0.2366	1.0

4.3. Summary

This section considered the influence of the base frequency and variance of movement on swarm movement patterns. The base frequency and variance influence the swarm's range of movement and other characteristics of the particles' trajectories such as smoothness, oscillation, and degree of correlation between consequent positions.

It was shown that the optimal value for the base frequency and variance depends on the dimensionality of the problem. On the high dimensional problems, the optimal configurations had low base frequencies and low variance of movement. For all except the smallest variance value, the optimal base frequency was the smallest possible value (0.05) and configurations with base frequencies larger than 0.15 exhibited poor performance. In contrast, for low dimensional problems, the optimal base frequency was usually larger, between 0.1 and 0.2. The low dimensional problems also benefited from larger variances than the high dimensional problems. In high dimensional spaces, particles should exhibit low range of movement and smooth trajectories for optimal performance. Parameters which bring about strong correlation between particle positions and low variance exhibited granular searching and reduced unwanted roaming behaviour.

The experiments presented in this section also confirmed that swarms with high inertia weights and low acceleration coefficients generally perform better than those with low inertia weights and high acceleration coefficients.

5. Further Improvement by Velocity Clamping

The previous two sections examined whether specific values of the inertia weight and acceleration coefficients are sufficient to reduce swarm variance and prevent particle roaming. Ideally, with optimal values for these parameters the swarm's movement patterns would facilitate efficient search space exploration.

Velocity clamping reduces the maximum particle step size, which influences the search dynamics. Although velocity clamping by itself is not sufficient to prevent particle roaming [18], it may be that swarms exhibiting movement patterns that are favourable for high dimensional problems (such as the configurations found in the previous section) may perform even better when velocity clamping is applied.

Section 5.1 discusses the clamping strategies under consideration. Both of these strategies are compared to unconstrained versions of the five best configurations from the previous section. Section 5.2 explains the experimental details, section 5.3 discusses the results, and section 5.4 concludes with a summary.

5.1. Velocity Clamping

Two methods of velocity clamping are considered: component-wise clamping in section 5.1.1, and scalar clamping in section 5.1.2.

5.1.1. Component-Wise Clamping

Component-wise clamping, as proposed by Eberhart and Kennedy [13], clamps the absolute value of the velocity in each dimension j to some maximum value, $v_{max,j}$. The value for $v_{max,j}$ is typically chosen as a fraction of the search space, as below:

$$v_{max,j} = \delta(U_j - L_j) \tag{28}$$

where $\delta \in [0, 1]$.

Clamping the velocity in every component separately decreases the magnitude of the velocity vector but may also change its direction. Velocity clamping per dimension may thus force particles into unfavourable regions of the search space, by distorting information from the local and global bests and from the particle's previous trajectory [32, 34].

5.1.2. Scalar Clamping

Velocity clamping may also be applied based on the magnitude of the entire velocity vector, taking care not to influence its direction. The entire velocity vector is scaled so that $\|\mathbf{v}\|$ does not exceed v_{max}. As before, v_{max} is chosen as a function of the search space's size:

$$\begin{align} v_{max} &= \delta\sqrt{\sum_{j=1}^{n}(U_j - L_j)^2} \tag{29} \\ &= \delta\|\mathbf{U} - \mathbf{L}\| \tag{30} \end{align}$$

where $\mathbf{U} = [U_1, U_2, ..., U_n]^T$ and $\mathbf{L} = [L_1, L_2, ..., L_n]^T$ are n-dimensional vectors.

Although this method preserves search direction, clamping in this manner may "unfairly" penalize the entire velocity vector due to a few large components. This may cause particles to explore slowly in the majority of problem dimensions due to a few outlier dimensions with large absolute values.

5.2. Experimental Method

The experiment applied both component-wise and scalar clamping to the top five configurations from table 4. A range of δ values were considered, specifically:

$$\delta \in \{0.001, 0.005, 0.01, 0.05, 0.1, 0.15\} \tag{31}$$

A previous study of velocity clamping in high dimensional spaces found that component-wise clamping with very small δ-values performed well [18]. However, that study only considered the rule-of-thumb parameters: $w = 0.729844$ and $c_1 = c_2 = 1.49618$. Based on these previous findings, the δ-values under consideration include very small values.

All other experimental settings were the same as in sections 3.2 and 4.1.

5.3. Results of Velocity Clamping

Figures 32 to 36 indicate the best clamping strategy for each of the tested configurations.

Clamping per component was the better strategy for all configurations except configuration E. The optimal δ values were on the smaller end of the scale, between 0.001 and 0.05. It is not clear that further trends regarding optimal clamping strategies can be extrapolated from these five configurations.

Figures 37 to 46 illustrate the result of pairwise Mann-Whitney U tests (with $p = 0.05$) comparing each unconstrained configuration to its clamped counterparts. The number of functions on which the the unconstrained swarm performed statistically significantly better in terms of fitness than the clamped swarm is illustrated in green; ties are shown in yellow and wins by the clamped swarms are shown in red.

For configuration A, figures 37 and 38 show that the unconstrained swarm performed significantly better than all of the clamped swarms. It is thus possible to choose values of the inertia weight and acceleration coefficients that do not require the application of velocity clamping to perform well.

However, the pairwise comparisons for the remaining configurations imply a more complicated situation. In contrast to configuration A, component-wise clamping performed significantly better than or outperformed the unconstrained swarm on all but the most extreme δ-values. Scalar clamping generally performed the same or worse than the unconstrained swarm. The question of whether velocity clamping is beneficial thus depends strongly on the values of the inertia weight and acceleration coefficients.

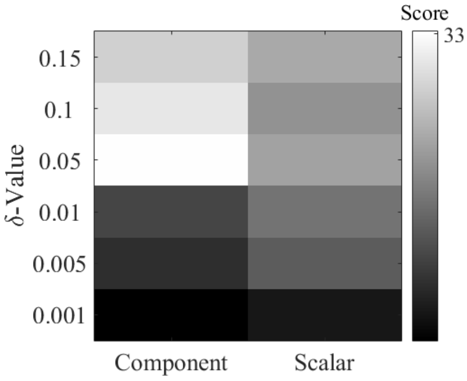

Figure 32: Relative performance of clamping strategies for configuration A (n=1000)

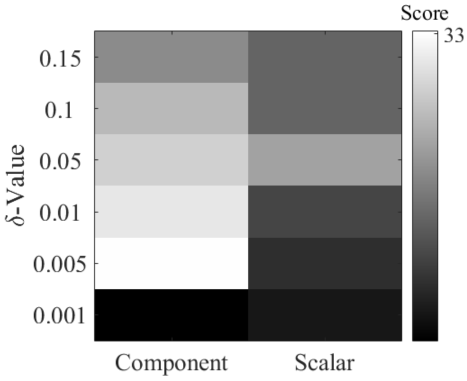

Figure 33: Relative performance of clamping strategies for configuration B (n=1000)

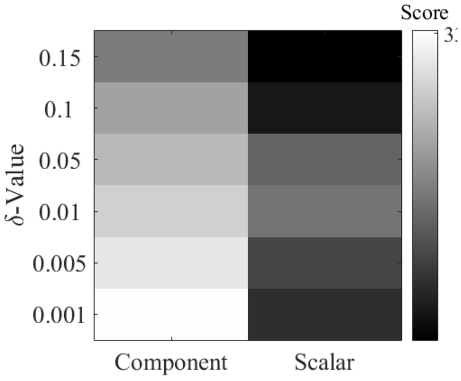

Figure 34: Relative performance of clamping strategies for configuration C (n=1000)

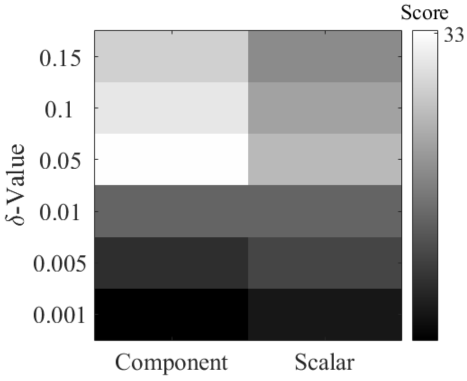

Figure 35: Relative performance of clamping strategies for configuration D (n=1000)

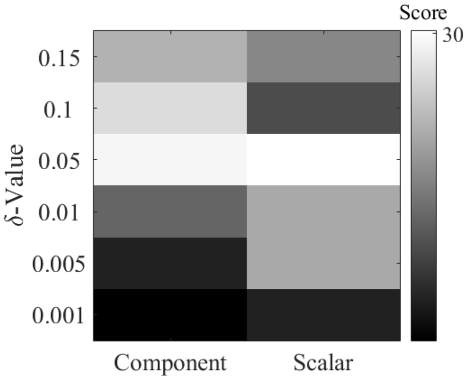

Figure 36: Relative performance of clamping strategies for configuration E (n=1000)

As with configuration B, component-wise clamping performed better than or the same as the unconstrained swarm. However, scalar clamping performed significantly better than the unconstrained swarm for δ-values smaller than or equal to 0.05.

For configuration D, both component-wise and scalar clamping performed better than or the same as the unconstrained swarm for δ-values larger than 0.01 (unlike configuration B where smaller δ-values performed better).

As with configuration A, the unconstrained swarm performed significantly better than the clamped swarms on most of the benchmark functions for configuration E.

It is thus not immediately apparent whether the application of velocity clamping will improve a swarm's performance. Configuration A and E had the lowest inertia weights and thus may have benefited less from velocity clamping than the other configurations. However, it remains unclear what the relationship between the velocity update parameters and the optimal δ-value may be.

Figure 47 compares the performance of all the optimally clamped swarms (optimal according to the results in figures 32 to 36) and their unconstrained counterparts. Although applying velocity clamping does not improve the performance of configuration A, an optimally clamped version of configuration C performs significantly better than any version of configuration A. The first and third best swarms utilized velocity clamping whereas the second and fourth best swarms were unconstrained. There is thus no clear rule of thumb as to whether clamping is generally beneficial: the optimal clamping strategy depends strongly on the values of the acceleration coefficients and the inertia weight.

5.4. Summary

Swarm configurations with high inertia weights may benefit from velocity clamping. The best configuration from section 4.2.1 performed significantly better than, or the same as all of the clamped swarms with the same velocity update coefficients. However, other configurations in the top five did benefit from velocity clamping and, in fact, performed significantly better than the

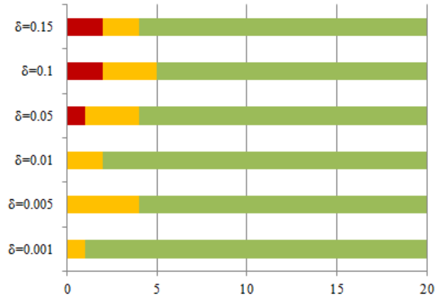

Figure 37: Unconstrained swarm compared to component-wise clamped for configuration A

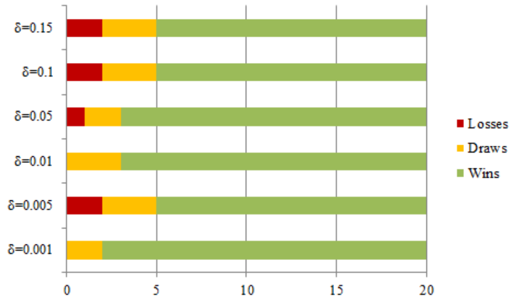

Figure 38: Unconstrained swarm compared to scalar clamped for configuration A

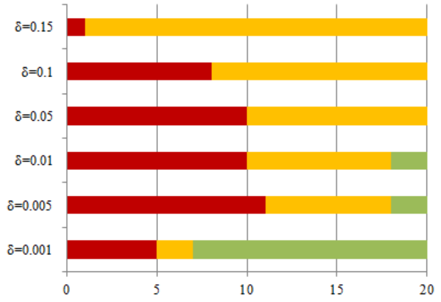

Figure 39: Unconstrained swarm compared to component-wise clamped for configuration B

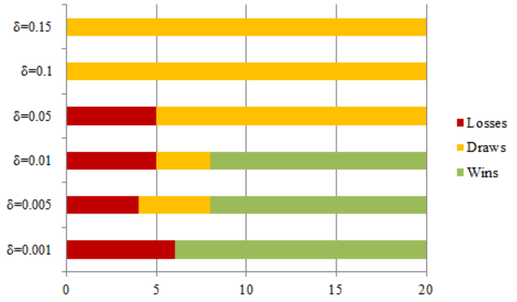

Figure 40: Unconstrained swarm compared to scalar clamped for configuration B

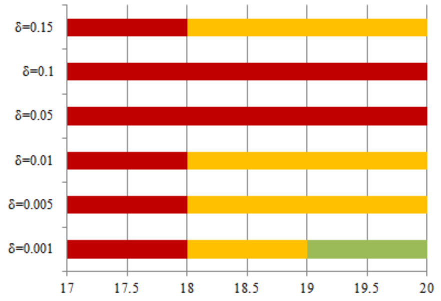

Figure 41: Unconstrained swarm compared to component-wise clamped for configuration C

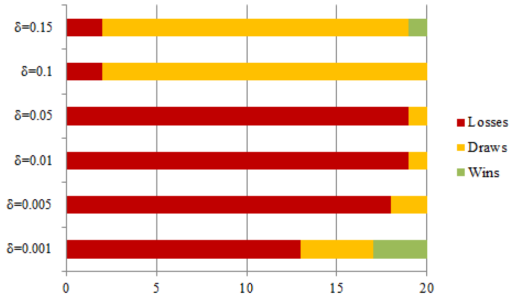

Figure 42: Unconstrained swarm compared to scalar clamped for configuration C

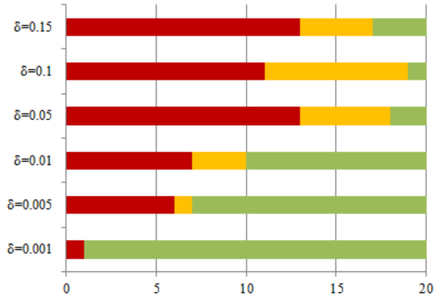

Figure 43: Unconstrained swarm compared to component-wise clamped for configuration D

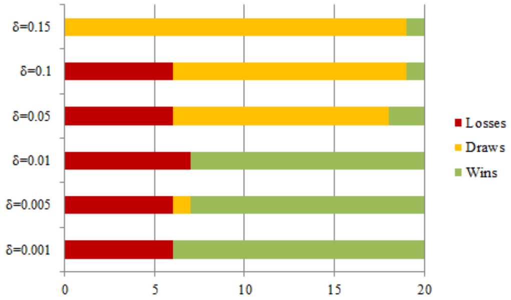

Figure 44: Unconstrained swarm compared to scalar clamped for configuration D

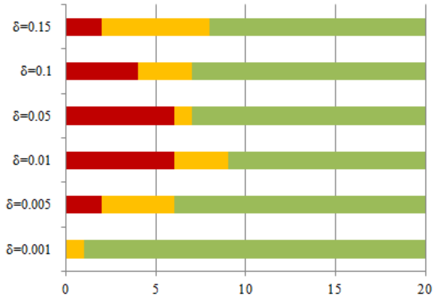

Figure 45: Unconstrained swarm compared to component-wise clamped for configuration E

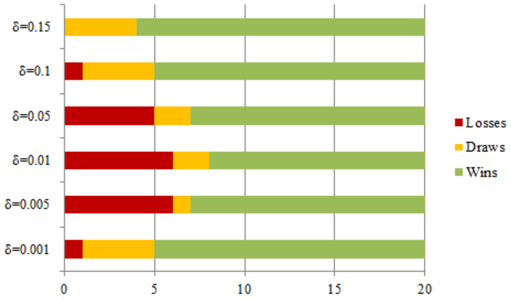

Figure 46: Unconstrained swarm compared to scalar clamped for configuration E

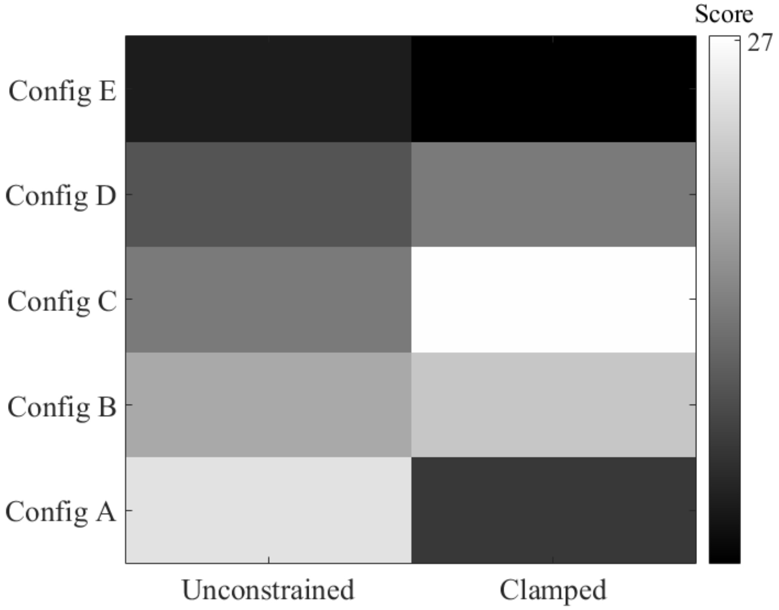

Figure 47: Comparison among the top 5 swarm configurations, with unconstrained swarms in the first column and their optimally clamped counterparts in the second column.

best, unconstrained configuration. It is thus not immediately clear whether a swarm with given inertia weight and acceleration coefficients will benefit from velocity clamping. The optimal clamping strategy and δ-values are also not easily inferred from the swarm's inertia weight and acceleration coefficients.

6. Conclusion

This work used the inertia weight and acceleration coefficients to control the variance of particle positions. Restricting the variance by a fraction successfully reduced the swarm's velocity. As the variance was restricted more severely, the average velocity magnitude of the swarm decreased, as desired. However, decreasing the variance of positions is not sufficient to guarantee good performance. Even swarms that were restricted to the same variance on position movements exhibited very different behaviour. Thus, the chosen values for the inertia weight and acceleration coefficients were shown to be important.

Swarms with high inertia weights and low acceleration coefficients performed the best in high dimensions. High inertia weights have a regularizing effect on a particle's trajectory, making it smooth and granular. The particle is less likely to rapidly divert its direction when its attractors are updated, since the influence of its momentum is stronger than the influence of its attractors. Thus, if the particle's attractors change drastically from one iteration to the next, a momentum-focused particle will be less prone to oscillation and velocity explosion than a particle that has high acceleration coefficients.

The effect of different movement patterns was also studied. Different movement patterns were brought about by changing the value of the base frequency and variance of movement which influence the swarm's movement range and other characteristics of the particles' trajectories such as smoothness, oscillation, and degree of correlation between consequent positions. Different configurations of the movement parameters were tested in high dimensions to determine whether particular movement patterns are advantageous in high dimensional search spaces.

It was found that small base frequencies and low variance were key factors in the swarms that performed well and exhibited significantly less roaming. Small base frequencies are associated with smooth, granular particle trajectories with weak to strong correlation between particle positions. Low variance is associated with restricted range of movement. Smooth trajectories with low variance of movement generally correspond to high inertia weights and low acceleration coefficients (eg. $w = 0.9694$ and $c_1 = c_2 = 0.099381$).

Lastly, the paper investigated the effects of velocity clamping on swarms with configurations that perform well in high dimensional spaces when unconstrained.

It was found that applying velocity clamping does not necessarily improve the performance of a swarm with a given inertia weight and acceleration coefficients. However, the configurations that perform well when unconstrained may perform better when velocity clamping is applied and vice versa. Future work may perform more extensive experimentation to investigate the relationship between velocity update coefficients and clamping strategies further.

This paper considered the influence of the inertia weight and acceleration coefficients on particle roaming in isolation from other swarm parameters such as neighbourhood topology and swarm size. Further research may consider the effect of these aspects on optimal movement patterns in high dimensional spaces.

Future work may consider developig a self-adaptive PSO which calculates the appropriate inertia weight and acceleration coefficients to bring about a desired movement pattern. The inertia weight and acceleration coefficients can be recalculated whenever the global and personal best positions are updated, to maintain the desired variance in particle positions.

References

[1] Bellman, R. (1957). *Dynamic Programming*. Rand Corporation research study. Princeton University Press.

[2] Beyer, K. S., Goldstein, J., Ramakrishnan, R., & Shaft, U. (1999). When

is "nearest neighbor" meaningful? In *Proceedings of the 7th International Conference on Database Theory* (pp. 217–235). Springer-Verlag.

[3] Bonyadi, M. R., & Michalewicz, Z. (2016). Stability analysis of the particle swarm optimization without stagnation assumption. *IEEE Transactions on Evolutionary Computation*, *20*, 814–819. doi:10.1109/TEVC.2015.2508101.

[4] Bonyadi, M. R., & Michalewicz, Z. (2017). Impacts of coefficients on movement patterns in the particle swarm optimization algorithm. *IEEE Transactions on Evolutionary Computation*, *21*, 378–390. doi:10.1109/TEVC.2016.2605668.

[5] Bonyadi, M. R., & Michalewicz, Z. (2017). Particle swarm optimization for single objective continuous space problems: A review. *Evolutionary Computation*, *25*, 1–54. URL: https://doi.org/10.1162/EVCO_r_00180. doi:10.1162/EVCO_r_00180.

[6] Bosman, P., & Engelbrecht, A. (2014). Diversity rate of change measurement for particle swarm optimisers. In *Swarm Intelligence* (pp. 86–97). Cham: Springer International Publishing.

[7] Bratton, D., & Kennedy, J. (2007). Defining a standard for particle swarm optimization. In *Proceedings of the IEEE Swarm Intelligence Symposium* (pp. 120–127). IEEE Computer Society. doi:10.1109/SIS.2007.368035.

[8] Chen, J., Xin, B., Peng, Z., Dou, L., & Zhang, J. (2009). Optimal contraction theorem for explorationexploitation tradeoff in search and optimization. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, *39*, 680–691. doi:10.1109/TSMCA.2009.2012436.

[9] Chen, S., Montgomery, J., & Bolufé-Röhler, A. (2015). Measuring the curse of dimensionality and its effects on particle swarm optimization and differential evolution. *Applied Intelligence*, *42*, 514–526. doi:10.1007/s10489-014-0613-2.

[10] Cleghorn, C. (2017). *Particle Swarm Optimization: Empirical and Theoretical Stability Analysis*. Ph.D. thesis University of Pretoria Pretoria, South Africa, South Africa.

[11] Cleghorn, C. W., & Engelbrecht, A. P. (2017). Particle swarm stability: a theoretical extension using the non-stagnate distribution assumption. *Swarm Intelligence*, . doi:`10.1007/s11721-017-0141-x`.

[12] Clerc, M., & Kennedy, J. (2002). The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, *6*, 58–73. doi:`10.1109/4235.985692`.

[13] Eberhart, R., & Kennedy, J. (1995). A new optimizer using particle swarm theory. In *Proceedings of the Sixth International Symposium on Micro Machine and Human Science* (pp. 39–43). doi:`10.1109/MHS.1995.494215`.

[14] Engelbrecht, A. (2012). Particle swarm optimization: Velocity initialization. In *Proceedings of the IEEE Congress on Evolutionary Computation* (pp. 1–8). doi:`10.1109/CEC.2012.6256112`.

[15] Engelbrecht, A. P. (2002). *Computational Intelligence: An Introduction*. Chichester, England: John Wiley & Sons.

[16] 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)* (pp. 124–135). doi:`10.1109/BRICS-CCI-CBIC.2013.31`.

[17] Engelbrecht, A. P. (2013). Roaming behavior of unconstrained particles. In *Proceedings of the 2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence* BRICS-CCI-CBIC '13 (pp. 104–111). IEEE Computer Society. doi:`10.1109/BRICS-CCI-CBIC.2013.28`.

[18] E.T., O., A.P., E., & C.W., C. (2017). The merits of velocity clamping particle swarm optimisation in high dimensional spaces. In *Proceedings of the IEEE Symposium on Swarm Intelligence* (pp. 785–792).

[19] Helwig, S., Branke, J., & Mostaghim, S. (2013). Experimental analysis of bound handling techniques in particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, *17*, 259–271. doi:10.1109/TEVC.2012.2189404.

[20] Helwig, S., & Wanka, R. (2007). Particle swarm optimization in high-dimensional bounded search spaces. In *2007 IEEE Swarm Intelligence Symposium* (pp. 198–205). doi:10.1109/SIS.2007.368046.

[21] Helwig, S., & Wanka, R. (2008). Theoretical analysis of initial particle swarm behavior. In *Proceedings of the 10th International Conference on Parallel Problem Solving from Nature - Volume 5199* (pp. 889–898). Springer-Verlag New York, Inc.

[22] Jiang, M., Luo, Y., & Yang, S. (2007). Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm. *Information Processing Letters*, *102*, 8 – 16. doi:https://doi.org/10.1016/j.ipl.2006.10.005.

[23] Jurez-Castillo, E., Acosta-Mesa, H.-G., & Mezura-Montes, E. (2018). Adaptive boundary constraint-handling scheme for constrained optimization, . (pp. 1–34). URL: https://app.dimensions.ai/details/publication/pub.1106285976. doi:10.1007/s00500-018-3459-4. Exported from https://app.dimensions.ai on 2019/01/20.

[24] Kennedy, J. (2003). Bare bones particle swarms. In *Swarm Intelligence Symposium, 2003. SIS '03. Proceedings of the 2003 IEEE* (pp. 80–87). doi:10.1109/SIS.2003.1202251.

[25] Krink, T., Vesterstrøm, J. S., & Riget, J. (2002). Particle swarm optimisation with spatial particle extension. In *Proceedings of the Congres on*

Evolutionary Computation CEC - Volume 02 CEC '02 (pp. 1474–1479). IEEE Computer Society.

[26] Liu, Q. (2015). Order-2 stability analysis of particle swarm optimization. *Evolutionary Computation*, *23*, 187–216.

[27] Oldewage, E. T. (2018). *The Perils of Particle Swarm Optimisation in High Dimensional ProblemSpaces*. Master's thesis University of Pretoria Pretoria, South Africa.

[28] Olorunda, O., & Engelbrecht, A. P. (2008). Measuring exploration/exploitation in particle swarms using swarm diversity. In *Proceedings of the IEEE Congress on Evolutionary Computation* (pp. 1128–1134). doi:10.1109/CEC.2008.4630938.

[29] Padhye, N., Deb, K., & Mittal, P. (2013). Boundary handling approaches in particle swarm optimization. In J. C. Bansal, P. K. Singh, K. Deep, M. Pant, & A. K. Nagar (Eds.), *Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications* (pp. 287–298). Springer India volume 1. doi:10.1007/978-81-322-1038-2_25.

[30] Poli, R. (2009). Mean and variance of the sampling distribution of particle swarm optimizers during stagnation. *IEEE Transactions on Evolutionary Computation*, *13*, 712–721. doi:10.1109/TEVC.2008.2011744.

[31] Riget, J., & Vesterstrøm, J. (2002). *A Diversity-Guided Particle Swarm Optimizer - the ARPSO*. Technical Report.

[32] Rini, D. P., Shamsuddin, S. M., & Yuhaniz, S. S. (2011). Article: Particle swarm optimization: Technique, system and challenges. *International Journal of Computer Applications*, *14*, 19–27. Full text available.

[33] Salomon, R. (1996). Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions. a survey of some theoretical and practical aspects of genetic algorithms. *Biosystems*, *39*, 263 – 278. doi:http://dx.doi.org/10.1016/0303-2647(96)01621-8.

[34] Shahzad, F., Baig, A. R., Masood, S., Kamran, M., & Naveed, N. (2009). Opposition-based particle swarm optimization with velocity clamping (ovcpso). In *Advances in Computational Intelligence* (pp. 339–348). Berlin, Heidelberg: Springer Berlin Heidelberg.

[35] Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. In *Proceedings of the IEEE International Conference on Evolutionary Computation* (pp. 69–73). doi:10.1109/ICEC.1998.699146.

[36] Tang, K., Li, X., Suganthan, P. N., Yang, Z., & Weise, T. (2009). *Benchmark functions for the cec2010 special session and competition on large-scale global optimization*. Technical Report Nature Inspired Computation and Applications Laboratory.

[37] Trelea, T. C. (2003). The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information Processing Letters*, *85*, 317 – 325. doi:http://dx.doi.org/10.1016/S0020-0190(02)00447-7.

[38] van den Bergh, F. (2002). *An Analysis of Particle Swarm Optimizers*. Ph.D. thesis University of Pretoria Pretoria, South Africa, South Africa. AAI0804353.

[39] van den Bergh, F., & Engelbrecht, A. P. (2006). A study of particle swarm optimization particle trajectories. *Information Science*, *176*, 937–971. doi:10.1016/j.ins.2005.02.003.

[40] van Zyl, E., & Engelbrecht, A. (2015). A subspace-based method for pso initialization. In *Proceedings of the IEEE Symposium Series on Computational Intelligence* (pp. 226–233). doi:10.1109/SSCI.2015.42.

[41] van Zyl, E., & Engelbrecht, A. (2016). Group-based stochastic scaling for pso velocities. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)* (pp. 1862–1868).

Elre Oldewage received her Masters in Computer Science from the University of Pretoria in 2018. She is currently pursuing a Ph.D in Engineering with the Computational and Biological Learning Lab at the University of Cambridge, Cambridge, U.K. Her interests include swarm intelligence, large scale optimization, machine learning and deep learning.

Andries Engelbrecht received the Masters and PhD degrees in Computer Science from the University of Stellenbosch, South Africa, in 1994 and 1999 respectively. He is Professor in the Department of Industrial Engineering and the Computer Science Division at Stellenbosch University. He currently holds the position of Voigt Chair in Data Science. His research interests include swarm intelligence, evolutionary computation, neural networks, and applications data mining, bioinformatics, finance, and difficult optimization problems. He has published over 350 papers and is author of the books, Computational Intelligence: An Introduction and Fundamentals of Computational Swarm Intelligence.

Christopher Cleghorn received his Masters and PhD degrees in Computer Science from the University of Pretoria, South Africa, in 2013 and 2017 respectively. He is a senior lecturer in the Computer Science Department at the University of Pretoria, and a member of the Computational Intelligence Research Group. His research interests include swarm intelligence, evolutionary computation, and machine learning, with a strong focus of theoretical research. Dr Cleghorn annually serves as a reviewer for numerous international journals and conferences in domains ranging from swarm intelligence and neural networks to mathematical optimization.

Appendices

A. Benchmarks

The benchmark suite used in the paper was specifically developed for testing large-scale global optimisation algorithms by Tang *et al.* [36]. This appendix provides a detailed explanation of the benchmark suite's construction. The first section explains how problems of varying separability may be constructed. Section A.2 defines the benchmark suite and section A.3 concludes the appendix with a brief summary of the benchmark functions' properties such as modality, domains and separability.

A.1. Separability and Basic Functions

The benchmark suite consists of minimization problems of varying degrees of separability. Problem separability is often used as a measure of the problem's difficulty. Separable problems are considered easy and fully-nonseparable problems are considered difficult.

The partially separable problems contained in the benchmark suite fall into one of three classes. The first class of functions contains a number of dependent variables with all the remaining variables independent. The second class consists of multiple independent sub-components, with each sub-component being m-nonseparable. The third class is a combination of these two and consists of a number of independent sub-components, some separable and some m-nonseparable. The benchmark suite comprises of separable problems, fully-nonseparable problems and partially separable problems from all three classes.

Functions of varying degrees of separability are constructed by dividing the input variables into several groups, each of which can be kept independent or made dependent by means of coordinate rotation [33]. Each group of variables is then evaluated by one of six basic functions and these values are added together to produce the benchmark function's value. The six basic functions are listed below:

1. Sphere Function (F_{sphere})
2. Rotated Elliptic Function ($F_{rot\_elliptic}$)
3. Schwefel's Problem 1.2 ($F_{schwefel}$)
4. Rosenbrock's Function ($F_{rosenbrock}$)
5. Rotated Rastrigin's Function ($F_{rot\_rastrigin}$)
6. Rotated Ackley's Function (F_{rot_ackley})

Apart from the Sphere function, the basic functions are nonseparable. The rotated functions listed above are calculated by multiplying the input vector with an orthogonal matrix, then evaluating the rotated vector on the benchmark function. The number of variable groups is determined by specifying m, the number of variables in each group. The degree of problem separability is thus determined by the m parameter. The paper uses $m = 10$ to accommodate comparison of high dimensional functions with problems of dimensionality as low as 10.

The random variable grouping may be achieved as follows: let P be a random permutation of $\{1, 2, ..., n-1, n\}$ and let $\mathbf{x}$ be an n-dimensional variable. Then $\mathbf{x}(P_1 : P_m) = (x_{P_1}, x_{P_2}, ..., x_{P_{m-1}}, x_{P_m})^T$ is a random group of size m chosen from the components of $\mathbf{x}$. The random permutation is used in this manner to index the objective variables and produce random groups of the desired number and size.

A.2. Benchmark Functions

The definitions of all the benchmark functions are provided in sub-sections A.2.1 to A.2.5. The function definitions are grouped according to separability. As before, n denotes the problem dimensionality, m denotes the group size and $\mathbf{x}$ denotes an input variable or candidate solution. The global optimum of a function is denoted by $\mathbf{o}$ and $\mathbf{z}$ denotes the shifted candidate solution $\mathbf{z} = \mathbf{x} - \mathbf{o}$. P denotes a random permutation of $\{1, 2, ..., n-1, n\}$ as described previously.

A.2.1. Separable Functions

The benchmark suite contains three separable functions, which are defined in table 8.

Table 8: Separable functions

Function	Name	Expression
F_1	Shifted Elliptic Function	$F_{elliptic}(\mathbf{z})$
F_2	Shifted Rastrigin's Function	$F_{rastrigin}(\mathbf{z})$
F_3	Shifted Ackley's Function	$F_{ackley}(\mathbf{z})$

A.2.2. Single-Group m-Nonseparable Functions

A partially separable benchmark function is called single-group m-nonseparable if it contains a number of dependent variables with all the rest of the variables independent. These functions fall into the first class of partially separable problems. The general form of these problems is given in Equation (32):

$$F(\mathbf{x}) = F_\gamma\big(\mathbf{z}(P_1 : P_m)\big) \times 10^6 + F_\alpha\big(\mathbf{z}(P_{m+1} : P_n)\big) \tag{32}$$

where F_γ is a nonseparable basic function, F_α is a separable basic function and $\mathbf{z}$ is obtained from $\mathbf{x}$ and $\mathbf{o}$ as described previously. There are five single-group m-nonseparable functions in the benchmark suite, provided in table 9 by specifying F_γ and F_α.

Table 9: Single-Group m-nonseparable Functions

Function	F_γ	F_α
F_4	$F_{rot\_elliptic}$	$F_{elliptic}$
F_5	$F_{rot\_rastrigin}$	$F_{rastrigin}$
F_6	F_{rot_ackley}	F_{ackley}
F_7	$F_{schwefel}$	F_{sphere}
F_8	$F_{rosenbrock}$	F_{sphere}

A.2.3. $\frac{n}{2m}$*-Group and m-Nonseparable Functions*

A partially separable benchmark function is called $\frac{n}{2m}$-group and m-nonseparable if it consists of $\frac{n}{2m}+1$ independent components, where the first $\frac{n}{2m}$ components are m-nonseparable and the last component is separable. These functions fall into the third class of partially separable problems. The general form of these problems is given in Equation (33):

$$F(\mathbf{x}) = \sum_{k=1}^{\frac{n}{2m}} F_\gamma\big(\mathbf{z}(P_{(k-1)m} : P_{km})\big) + F_\alpha\big(\mathbf{z}(P_{\frac{n}{2}+1} : P_n)\big) \tag{33}$$

where F_γ is a nonseparable basic function and F_α is a separable basic function. There are five $\frac{n}{2m}$-group and m-nonseparable functions in the benchmark suite, provided in table 10 below by specifying F_γ.

Table 10: $\frac{n}{2m}$-Group and m-nonseparable functions

Function	F_γ	F_α
F_9	$F_{rot\_elliptic}$	$F_{elliptic}$
F_{10}	$F_{rot\_rastrigin}$	$F_{rastrigin}$
F_{11}	F_{rot_ackley}	F_{ackley}
F_{12}	$F_{schwefel}$	F_{sphere}
F_{13}	$F_{rosenbrock}$	F_{sphere}

A.2.4. $\frac{n}{m}$*-Group and m-Nonseparable Functions*

A partially separable benchmark function is called $\frac{n}{m}$-group and m-nonseparable if it consists of $\frac{n}{m}$ independent sub-components, all nonseparable. These functions fall into the second class of partially separable problems. The general form of these problems is given in Equation (34):

$$\sum_{k=1}^{\frac{n}{m}} F_\gamma\big(\mathbf{z}(P_{(k-1)m} : P_{km})\big) \tag{34}$$

where F_γ is a nonseparable basic function. There are five $\frac{n}{m}$-group and m-nonseparable functions in the benchmark suite, provided in table 11 below by specifying F_γ.

Table 11: $\frac{n}{m}$-Group and m-nonseparable functions

Function	F_γ
F_{14}	$F_{rot\_elliptic}$
F_{15}	$F_{rot\_rastrigin}$
F_{16}	F_{rot_ackley}
F_{17}	$F_{schwefel}$
F_{18}	$F_{rosenbrock}$

A.2.5. Nonseparable Functions

The benchmark suite contains two nonseparable functions, which are defined in table 12.

Table 12: Nonseparable functions

Function	Name	Expression
F_{19}	Shifted Schwefel's Problem 1.2	$F_{schwefel}(\mathbf{z})$
F_{20}	Shifted Rosenbrock's Function	$F_{rosenbrock}(\mathbf{z})$

A.3. Benchmark Function Summary

The benchmark suite consists of 20 minimization problems, of varying degrees of separability. The optimal objective function value is zero for all the functions.

Table 13 summarizes each benchmark function's degree of separability, modality and domain. The class of partially separable functions is specified. The modality column denotes unimodal functions with "U" and multimodal functions with "M". The optimal objective function value for all of the benchmark functions is 0.

Table 13: Benchmark functions

Function	Separability	Modality	Domain
$F1$	separable	U	$[-100, 100]^n$
$F2$	separable	M	$[-5, 5]^n$
$F3$	separable	M	$[-32, 32]^n$
$F4$	partial class 1	U	$[-100, 100]^n$
$F5$	partial class 1	M	$[-5, 5]^n$
$F6$	partial class 1	M	$[-32, 32]^n$
$F7$	partial class 1	U	$[-100, 100]^n$
$F8$	partial class 1	M	$[-100, 100]^n$
$F9$	partial class 3	U	$[-100, 100]^n$
$F10$	partial class 3	M	$[-5, 5]^n$
$F11$	partial class 3	M	$[-32, 32]^n$
$F12$	partial class 3	U	$[-100, 100]^n$
$F13$	partial class 3	M	$[-100, 100]^n$
$F14$	partial class 2	U	$[-100, 100]^n$
$F15$	partial class 2	M	$[-5, 5]^n$
$F16$	partial class 2	M	$[-32, 32]^n$
$F17$	partial class 2	U	$[-100, 100]^n$
$F18$	partial class 2	M	$[-100, 100]^n$
$F19$	fully nonseparable	U	$[-100, 100]^n$
$F20$	fully nonseparable	M	$[-100, 100]^n$

B. Additional Results

This appendix provides the empirical results for the three best and three worst configurations from each experiment. The tables list the mean and standard deviation of the best fitness achieved by a swarm over all 30 simulations.

Table 14: Best Fitness Achieved for Three Best Configurations (Section 3)

Func	**c=0.2786 w=0.95**	**c=0.5182 w=0.9**	**c=0.4274 w=0.9**
F1	$1.091e+11 \pm 5.762e+09$	$2.756e+11 \pm 4.894e+09$	$2.994e+11 \pm 2.842e+09$
F2	$1.175e+12 \pm 3.919e+10$	$4.051e+12 \pm 1.444e+11$	$5.105e+12 \pm 3.349e+10$
F3	$2.044e+01 \pm 2.864e-02$	$2.129e+01 \pm 9.703e-03$	$2.133e+01 \pm 3.746e-03$
F4	$6.336e+13 \pm 2.401e+13$	$3.239e+14 \pm 4.920e+13$	$3.820e+14 \pm 5.644e+13$
F5	$6.736e+07 \pm 5.114e+06$	$1.090e+08 \pm 4.824e+06$	$1.225e+08 \pm 3.479e+06$
F6	$1.838e+07 \pm 2.753e+05$	$1.979e+07 \pm 1.400e+05$	$1.990e+07 \pm 1.116e+05$
F7	$1.758e+10 \pm 9.611e+09$	$2.612e+11 \pm 2.555e+10$	$3.179e+11 \pm 2.500e+10$
F8	$3.374e+14 \pm 1.319e+14$	$4.504e+15 \pm 5.148e+14$	$7.137e+15 \pm 6.462e+14$
F9	$1.589e+11 \pm 7.063e+09$	$3.557e+11 \pm 6.324e+09$	$3.920e+11 \pm 2.849e+09$
F10	$1.382e+04 \pm 1.567e+02$	$2.096e+04 \pm 1.055e+02$	$2.168e+04 \pm 4.785e+01$
F11	$1.019e+03 \pm 1.630e+00$	$1.074e+03 \pm 5.614e-01$	$1.077e+03 \pm 3.739e-01$
F12	$2.434e+07 \pm 8.031e+05$	$5.724e+07 \pm 8.087e+05$	$6.072e+07 \pm 4.287e+05$
F13	$4.546e+11 \pm 2.120e+10$	$1.777e+12 \pm 4.842e+10$	$2.112e+12 \pm 1.397e+10$
F14	$2.114e+11 \pm 8.269e+09$	$4.629e+11 \pm 9.380e+09$	$5.027e+11 \pm 5.006e+09$
F15	$1.382e+04 \pm 1.661e+02$	$2.087e+04 \pm 1.308e+02$	$2.161e+04 \pm 4.556e+01$
F16	$2.022e+03 \pm 3.090e+00$	$2.111e+03 \pm 1.042e+00$	$2.113e+03 \pm 6.447e-01$
F17	$6.211e+07 \pm 1.465e+06$	$1.185e+08 \pm 1.444e+06$	$1.256e+08 \pm 5.315e+05$
F18	$1.112e+12 \pm 3.951e+10$	$3.668e+12 \pm 1.559e+11$	$4.604e+12 \pm 2.349e+10$
F19	$7.397e+11 \pm 1.266e+10$	$1.426e+12 \pm 1.697e+10$	$1.505e+12 \pm 7.044e+09$
F20	$1.362e+04 \pm 1.464e+02$	$2.034e+04 \pm 1.670e+02$	$2.121e+04 \pm 5.826e+01$

Table 15: Best Fitness Achieved for Three Worst Configurations (Section 3)

Func	**c=0 w=1**	**c=1.575 w=0.4**	**c=1.8462 w=0.5**
F1	$4.496e+11 \pm 2.675e+09$	$4.444e+11 \pm 3.641e+09$	$4.427e+11 \pm 2.597e+09$
F2	$1.018e+13 \pm 3.378e+10$	$1.006e+13 \pm 3.097e+10$	$1.010e+13 \pm 3.363e+10$
F3	$2.154e+01 \pm 1.479e-03$	$2.154e+01 \pm 1.857e-03$	$2.154e+01 \pm 2.090e-03$
F4	$1.471e+15 \pm 1.004e+14$	$1.371e+15 \pm 1.405e+14$	$1.417e+15 \pm 1.040e+14$
F5	$1.979e+08 \pm 3.600e+06$	$1.935e+08 \pm 3.912e+06$	$2.023e+08 \pm 3.814e+06$
F6	$2.108e+07 \pm 3.818e+04$	$2.109e+07 \pm 4.500e+04$	$2.113e+07 \pm 2.845e+04$
F7	$8.677e+11 \pm 3.976e+10$	$8.929e+11 \pm 3.190e+10$	$8.447e+11 \pm 3.793e+10$
F8	$3.289e+16 \pm 2.626e+15$	$3.506e+16 \pm 2.100e+15$	$3.187e+16 \pm 2.145e+15$
F9	$5.896e+11 \pm 3.733e+09$	$5.934e+11 \pm 3.240e+09$	$5.807e+11 \pm 4.480e+09$
F10	$2.651e+04 \pm 3.950e+01$	$2.650e+04 \pm 3.370e+01$	$2.648e+04 \pm 3.328e+01$
F11	$1.092e+03 \pm 2.161e-01$	$1.092e+03 \pm 2.012e-01$	$1.092e+03 \pm 2.350e-01$
F12	$8.920e+07 \pm 4.882e+05$	$8.891e+07 \pm 3.758e+05$	$8.907e+07 \pm 4.550e+05$
F13	$4.343e+12 \pm 2.818e+10$	$4.364e+12 \pm 2.374e+10$	$4.325e+12 \pm 3.294e+10$
F14	$7.392e+11 \pm 4.580e+09$	$7.337e+11 \pm 5.809e+09$	$7.485e+11 \pm 4.630e+09$
F15	$2.648e+04 \pm 3.287e+01$	$2.649e+04 \pm 2.739e+01$	$2.636e+04 \pm 4.622e+01$
F16	$2.141e+03 \pm 2.518e-01$	$2.141e+03 \pm 2.234e-01$	$2.141e+03 \pm 2.810e-01$
F17	$1.796e+08 \pm 5.126e+05$	$1.797e+08 \pm 6.163e+05$	$1.792e+08 \pm 6.179e+05$
F18	$9.110e+12 \pm 3.600e+10$	$9.126e+12 \pm 3.815e+10$	$9.100e+12 \pm 4.008e+10$
F19	$2.128e+12 \pm 6.404e+09$	$2.130e+12 \pm 6.821e+09$	$2.147e+12 \pm 4.473e+09$
F20	$2.615e+04 \pm 3.502e+01$	$2.615e+04 \pm 3.573e+01$	$2.614e+04 \pm 4.158e+01$

Table 16: Best Fitness Achieved for Three Best Configurations (Section 4)

Func	**V=0.1 F=0.15**	**V=0.1 F=0.05**	**V=0.4 F=0.05**
F1	$5.699e + 10 \pm 1.706e + 09$	$9.199e + 10 \pm 1.659e + 09$	$1.262e + 11 \pm 4.066e + 09$
F2	$1.521e + 04 \pm 6.189e + 01$	$1.512e + 04 \pm 4.890e + 01$	$1.426e + 04 \pm 1.392e + 02$
F3	$2.068e + 01 \pm 7.470e - 03$	$2.067e + 01 \pm 7.686e - 03$	$2.062e + 01 \pm 1.883e - 02$
F4	$8.754e + 11 \pm 7.537e + 10$	$4.310e + 11 \pm 2.347e + 10$	$6.006e + 13 \pm 2.569e + 13$
F5	$5.322e + 07 \pm 2.772e + 06$	$5.086e + 07 \pm 2.727e + 06$	$6.121e + 07 \pm 3.996e + 06$
F6	$1.587e + 07 \pm 7.499e + 05$	$7.516e + 06 \pm 1.207e + 06$	$1.695e + 07 \pm 3.806e + 05$
F7	$4.554e + 06 \pm 3.218e + 04$	$4.553e + 06 \pm 3.699e + 04$	$6.032e + 10 \pm 1.628e + 10$
F8	$3.234e + 09 \pm 5.904e + 08$	$1.305e + 07 \pm 4.192e + 06$	$3.246e + 14 \pm 1.405e + 14$
F9	$7.957e + 10 \pm 2.002e + 09$	$1.168e + 11 \pm 1.998e + 09$	$1.680e + 11 \pm 4.895e + 09$
F10	$1.545e + 04 \pm 7.619e + 01$	$1.555e + 04 \pm 4.854e + 01$	$1.425e + 04 \pm 1.132e + 02$
F11	$1.029e + 03 \pm 7.894e - 01$	$1.019e + 03 \pm 1.003e + 00$	$1.033e + 03 \pm 2.640e + 00$
F12	$1.614e + 07 \pm 3.415e + 05$	$2.361e + 07 \pm 3.211e + 05$	$2.679e + 07 \pm 5.390e + 05$
F13	$2.351e + 11 \pm 7.481e + 09$	$3.680e + 11 \pm 8.716e + 09$	$4.736e + 11 \pm 2.022e + 10$
F14	$1.065e + 11 \pm 2.307e + 09$	$1.525e + 11 \pm 3.666e + 09$	$2.243e + 11 \pm 5.294e + 09$
F15	$1.538e + 04 \pm 7.738e + 01$	$1.533e + 04 \pm 4.065e + 01$	$1.412e + 04 \pm 1.430e + 02$
F16	$2.044e + 03 \pm 7.802e - 01$	$2.028e + 03 \pm 1.547e + 00$	$2.046e + 03 \pm 2.148e + 00$
F17	$4.815e + 07 \pm 6.165e + 05$	$5.886e + 07 \pm 4.518e + 05$	$6.416e + 07 \pm 1.048e + 06$
F18	$9.573e + 11 \pm 2.453e + 10$	$1.159e + 12 \pm 1.358e + 10$	$1.264e + 12 \pm 3.460e + 10$
F19	$5.808e + 11 \pm 7.488e + 09$	$7.224e + 11 \pm 6.170e + 09$	$7.869e + 11 \pm 1.580e + 10$
F20	$1.145e + 12 \pm 2.180e + 10$	$1.347e + 12 \pm 1.580e + 10$	$1.412e + 12 \pm 3.761e + 10$

Table 17: Best Fitness Achieved for Three Worst Configurations (Section 4)

Func	**V=0.4 F=0.25**	**V=1.6 F=0.2**	**V=1.6 F=0.4**
F1	$4.444e + 11 \pm 3.266e + 09$	$4.512e + 11 \pm 2.239e + 09$	$4.440e + 11 \pm 3.377e + 09$
F2	$2.616e + 04 \pm 3.566e + 01$	$2.610e + 04 \pm 4.071e + 01$	$2.613e + 04 \pm 3.013e + 01$
F3	$2.154e + 01 \pm 1.778e - 03$	$2.154e + 01 \pm 1.606e - 03$	$2.154e + 01 \pm 1.440e - 03$
F4	$1.370e + 15 \pm 8.519e + 13$	$1.540e + 15 \pm 1.098e + 14$	$1.539e + 15 \pm 9.432e + 13$
F5	$2.033e + 08 \pm 2.468e + 06$	$2.055e + 08 \pm 3.282e + 06$	$2.023e + 08 \pm 3.002e + 06$
F6	$2.108e + 07 \pm 2.989e + 04$	$2.109e + 07 \pm 2.422e + 04$	$2.107e + 07 \pm 4.509e + 04$
F7	$7.596e + 11 \pm 3.905e + 10$	$8.248e + 11 \pm 3.693e + 10$	$8.865e + 11 \pm 3.916e + 10$
F8	$3.207e + 16 \pm 1.982e + 15$	$3.070e + 16 \pm 2.123e + 15$	$2.944e + 16 \pm 1.997e + 15$
F9	$5.802e + 11 \pm 3.708e + 09$	$5.845e + 11 \pm 5.994e + 09$	$5.853e + 11 \pm 4.314e + 09$
F10	$2.658e + 04 \pm 3.897e + 01$	$2.649e + 04 \pm 4.551e + 01$	$2.648e + 04 \pm 3.892e + 01$
F11	$1.092e + 03 \pm 1.477e - 01$	$1.091e + 03 \pm 2.137e - 01$	$1.092e + 03 \pm 2.142e - 01$
F12	$8.891e + 07 \pm 4.625e + 05$	$8.878e + 07 \pm 3.704e + 05$	$8.904e + 07 \pm 5.086e + 05$
F13	$4.350e + 12 \pm 2.193e + 10$	$4.330e + 12 \pm 2.457e + 10$	$4.385e + 12 \pm 2.670e + 10$
F14	$7.443e + 11 \pm 4.730e + 09$	$7.407e + 11 \pm 5.505e + 09$	$7.400e + 11 \pm 5.159e + 09$
F15	$2.641e + 04 \pm 4.373e + 01$	$2.627e + 04 \pm 8.580e + 01$	$2.639e + 04 \pm 5.162e + 01$
F16	$2.141e + 03 \pm 2.278e - 01$	$2.141e + 03 \pm 2.379e - 01$	$2.141e + 03 \pm 1.646e - 01$
F17	$1.799e + 08 \pm 5.865e + 05$	$1.795e + 08 \pm 5.336e + 05$	$1.796e + 08 \pm 5.445e + 05$
F18	$9.123e + 12 \pm 3.425e + 10$	$9.139e + 12 \pm 3.371e + 10$	$9.097e + 12 \pm 4.112e + 10$
F19	$2.120e + 12 \pm 6.628e + 09$	$2.117e + 12 \pm 7.925e + 09$	$2.128e + 12 \pm 6.349e + 09$
F20	$1.010e + 13 \pm 2.730e + 10$	$1.011e + 13 \pm 4.310e + 10$	$1.008e + 13 \pm 3.336e + 10$