An Evaluation of Velocity Clamping Particle Swarm Optimisation for Convergent Control Parameters

E.J. van den Heever
Stellenbosch Unviersity, Department of Computer Science
22547134
22547134@sun.ac.za

***Abstract*—This paper investigates the role that velocity clamping plays in particle swarm optimisation, especially when the control parameters are chosen such that particles converge. It is shown that for convergent control parameters, velocity clamping has no significant benefit. Four measures are used to establish the influence that velocity clamping has on the performance of the PSO algorithm - namely quality of solution, diversity of the swarm, velocity magnitude of particles and percentage of particles that leave the search space.**

Keywords—particle swarm optimisation, velocity clamping, particle convergence

I. Introduction

Particle swarm optimisation (PSO) is a swarm intelligence algorithm and widely used technique for optimizing benchmark functions. It simulates the social behaviour of birds in a flock [1, p. 289]. The performance of an optimisation algorithm is heavily influenced by the exploration-exploitation trade-off. Particle swarm optimisation addresses these objectives through the velocity update equation. Early applications of basic PSO found that the velocity of particles far from the neighbourhood and personal best positions quickly explode to large values, resulting in large position updates and particles leaving the search space [1, pp. 303-304].

To prevent particle divergence, velocities can be clamped to keep particles within the boundary constraints. This method sets the velocity of a particle to a pre-determined velocity when the calculated velocity of a particle exceeds some maximum velocity [1, p. 304]. This paper aims to establish whether velocity clamping is necessary when the inertia weight, social component and cognitive component of the velocity update equation are chosen such that the particles converge. The role of velocity clamping is evaluated for three different choices of maximum velocity, using converging and non-converging selections of control parameters.

This study evaluates PSO simulations with converging and non-converging control parameters on five benchmark functions. Four measures are used to evaluate the role of velocity clamping in PSO. Data about the swarm diversity over time is collected and plotted for simulations without velocity clamping and simulations with velocity clamping for various maximum velocities. Swarm diversity over time is used as a measure of how well the particles in a swarm converge. Data regarding the quality of solution over time is also analysed to establish whether velocity clamping improves the quality of a solution for converging and non-converging control parameters. The influence of velocity clamping on velocity magnitude is measured and plotted for different maximum velocities. And lastly, the percentage of particles to leave the search space at each iteration is calculated and plotted for simulations with and without velocity clamping.

This research found that although velocity clamping is effective in improving the performance of PSO when control parameters do not result in convergence, it is unnecessary when control parameters are chosen such that particles converge to a solution. In some cases, PSO performs better when velocity clamping was not applied. Furthermore, good choice of control parameters has a greater influence on the performance of PSO than velocity clamping and thus should be the preferred strategy.

This paper is structured as follows: The second section gives necessary background for this paper, including a brief description of the global best PSO algorithm and an explanation of how it can be adapted to apply velocity clamping. The third section outlines the methodology for this research, and the fourth section discusses the chosen empirical procedure. Section five presents the results and a discussion thereof. The sixth and final section provides a conclusion for the study.

II. Background

Particle swarm optimisation (PSO) is a nature-inspired stochastic optimisation algorithm introduced by Kennedy and Eberhart [2]. A swarm of particles is initialised within the search space. Particles then move through the search space over time, using global and local information to inform their movement. The position of a particle represents a possible solution to an optimisation problem. This paper uses the inertia weight global best PSO model proposed by Shi and Eberhart [3].

A. Global Best PSO

Global Best (gbest) PSO updates the position of particles with time, according to a formula that considers the best-known position of the swarm and the best-known position of the particle. For global best PSO, the velocity of particle i in dimension j at time $t+1$ is calculated as

$$\begin{aligned} v_{ij}(t+1) = w v_{ij}(t) + c_1 r_1(t)\left[y_{ij}(t) - x_{ij}(t)\right] \\ + c_2 r_2(t)\left[\hat{y}_j(t) - x_{ij}(t)\right] \end{aligned} \tag{1}$$

where y_{ij} is the personal best position of particle i in dimension j and $\hat{y}_j$ is the global best position in dimension j. $x_{ij}(t)$ is the position of particle i in dimension j at time t. c_1 and c_2 are constants used to scale the cognitive and social components. The cognitive component is representative of the particle's memory of its best-known position and the social component draws the particle to the best-known position found by the swarm. w is the inertia weight.

The position of particle i at time $t+1$ is updated according to

$$\boldsymbol{x}_i(t+1) = \boldsymbol{x}_i(t) + \boldsymbol{v}_i(t+1) \tag{2}$$

Engelbrecht [1, p. 291] proposes the following algorithm for global best PSO

Algorithm 1: *gbest PSO*

Create and initialize an n_x-dimensional swarm
repeat
  for *each particle* $i = 1, \dots, n_s$ **do**
    // set the personal best position
    if $f(\boldsymbol{x}_i) < f(\boldsymbol{y}_i)$ **then**
      $\boldsymbol{y}_i = \boldsymbol{x}_i$;
    end
    //set the global best position **if** $f(\boldsymbol{y}_i) < f(\hat{\boldsymbol{y}})$ **then**
      $\hat{\boldsymbol{y}} = \boldsymbol{y}_i$;
    end
  end
  for *each particle* $i = 1, \dots, n_s$ **do**
    update the velocity using (1)
    update the position using (2)
  end
until *stopping condition has been met*

B. Velocity Clamping

Velocity clamping is an adaptation of PSO that prevents rapid velocity growth with the aim of limiting the number of particles that leave the search space [2]. It limits the step size of a particle to a maximum which is chosen as a fraction of the search space domain. This research updates velocities according to the formula proposed by Engelbrecht [1, p. 304]

$$v_{ij}(t+1) = \begin{cases} v'_{ij}(t+1) & \text{if } v'_{ij}(t+1) < V_{max,j} \\ V_{max,j} & \text{if } v'_{ij}(t+1) \geq V_{max,j} \end{cases} \tag{3}$$

Where $v'_{ij}(t+1)$ is calculated according to (1) and $V_{max,j}$ is calculated as

$$V_{max,j} = k \times (x_{max,i} - x_{min,i}), \forall i = 1, \dots, n_x, \text{with } k \in \{0.1, 0.3, 0.5\} \tag{4}$$

Algorithm 1 can be adapted to use the velocity update equation (3) instead of (1) to implement velocity clamping.

Note that velocity clamping does not confine particles to the search space, but rather limits the step sizes of particles as determined from the particle velocity [1, p. 304]. Although velocity clamping helps to prevent the explosion of velocities, it also changes the direction of a particle's movement, which may reduce the ability of PSO to find an optimum [1, p. 305].

C. Selection of Control Parameters

This research aims to test whether velocity clamping is necessary when the control parameters w, c_1 and c_2 are selected such that convergence conditions are satisfied. To do this, both non-converging and converging selections of control parameters must be considered. Poli and Broomhead [4] showed that particle trajectories are guaranteed to converge when w, c_1 and c_2 are chosen such that

$$c_1 + c_2 < \frac{24(1-w^2)}{7-5w} \tag{5}$$

III. Methodology

This section describes the implementation of PSO with and without velocity clamping.

This research implements a PSO simulation in Java according to Algorithm 1, with a stopping condition of 5000 iterations and dimension of 30. A swarm size of 30 particles is used and particles are initialised with a velocity of $\mathbf{0}$ and a uniform random position within the search space,

$$\boldsymbol{x}(0) = x_{min,j} + r_j(x_{max,j} - x_{min,j}), \quad \forall j = 1, \dots, n_x, \forall i = 1, \dots, n_s \tag{6}$$

as suggested by [1, p. 297], where $r_j \sim U(0,1)$ is generated using Math.random() in Java. The personal best position of each particle is initialised to its initial position, $\boldsymbol{x}(0)$. The global best variable is updated synchronously as suggested by Eberhart and Kennedy [2]. To guarantee that particle attractors always remain within the search space, personal and global best positions are only updated if they are within bounds, where the bounds are defined according to the corresponding choice of benchmark test. This is implemented by marking particles as 'dead' when they leave the search space and then re-labelling them if they re-enter the search space.

The PSO simulation was adapted to take an input $k \in \{0, 0.1, 0.3, 0.5\}$. When $k = 0$, velocity clamping was not applied and when $k > 0$, velocity clamping was applied and $V_{max,j}$ is calculated using k according to (4). It also takes control parameters w, c_1 and c_2 as input values, where the combinations of control parameters used are listed in Table I.

TABLE I. CHOICE OF CONTROL PARAMETERS

Combination	Control Parameter Values			
	w	c_1	c_2	***Convergent***
A	1.0	2.0	2.0	no
B	0.7	1.4	1.4	yes
C	0.9	2.0	2.0	no
D	0.9	0.7	0.7	yes

A suite of benchmark functions was constructed according to Table II. The suite consists of minimisation problems and contains a selection of multimodal and non-multimodal functions. All functions are chosen to be non-random.

TABLE II. SUITE OF BENCHMARK FUNCTIONS

Benchmark Function	Properties		
	Formula	***Domain***	***Global Minimum***
Styblinski Tank	$f(\boldsymbol{x}) = \frac{1}{2}\sum_{i=1}^{d}(x_i^4 - 16x_i^2 + 5x_i)$	$x_i \in [-5, 5], \forall i \in [\![1, d]\!]$	$f(-2.903534, \dots, -2.903534) = -39.16599d$
Schwefel	$f(\boldsymbol{x}) = 418.9829d - \sum_{i=1}^{d} x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	$x_i \in [-500, 500], \forall i \in [\![1, d]\!]$	$f(420.9687, \dots, 420.9687) = 0$
Spherical	$f(\boldsymbol{x}) = \sum_{i=1}^{d} x_i^2$	$x_i \in [-5.12, 5.12], \forall i \in [\![1, d]\!]$	$f(0, \dots, 0) = 0$
Powell	$f(\boldsymbol{x}) = \sum_{i=1}^{d} \lvert x_i \rvert^{i+1}$	$x_i \in [-1, 1], \forall i \in [\![1, d]\!]$	$f(0, \dots, 0) = 0$
Brown	$f(\boldsymbol{x}) = \sum_{i=1}^{d-1} (x_i^2)^{x_{i+1}^2+1} + (x_{i+1}^2)^{x_i^2+1}$	$x_i \in [-1, 4], \forall i \in [\![1, d]\!]$	$f(0, \dots, 0) = 0$

Each simulation is run for 5000 iterations on each benchmark function. And 20 independent samples are collected for each combination of input parameters. The input parameters consist of the control parameters w, c_1 and c_2 and the value of k.

Four performance measures are used to evaluate PSO for different input values. The diversity of the swarm over time is measured as the average Euclidean distance of particles from the swarm centre of mass. The quality of the solution over time is calculated as the minimum at the global best position found by the swarm. The average velocity magnitude and percentage of particles to leave the search space during each iteration is also measured. The data produced by these performance measures is output to .csv files for further analysis.

IV. Empirical Procedure

This researched aimed to establish whether velocity clamping is necessary when the control parameters w, c_1 and c_2 are chosen such that the convergence conditions in (5) are satisfied. To observe the effects of velocity clamping, PSO simulations were run with and without velocity clamping for different choices of control parameters.

Simulations were conducted for the combinations of control parameters described in Table I, where combinations A and C do not guarantee the convergence of particle trajectories according to (5) and combinations B and D do guarantee the convergence of particle trajectories. For each simulation, data was collected when no velocity clamping was applied and when velocity clamping was applied for k values of 0.1, 0.3 and 0.5. For each combination of control parameters and velocity clamping, 20 independent runs of the simulation were conducted. This produced a total of 320 independent runs for each benchmark function.

Five benchmark tests were selected to evaluate the effects of velocity clamping for different control parameters. The choice of benchmark functions was made to include multimodal and non-multimodal functions. Table II summarises the selected benchmark functions that were used to evaluate PSO for the minimisation of these functions. Schwefel and Styblinski Tank are multimodal functions and Brown, Powell and Sphere are non-multimodal functions [5]. All the selected benchmark functions are defined for $d \in \mathbb{N}_{+}^{*}$.

The effects of velocity clamping were evaluated against four measures at each iteration:

A. *The Diveristy of the Swarm*

The diversity of the swarm over time indicates the exploration-exploitation abilities of the swarm. The swarm diversity is calculated as the average Euclidean distance of particles from the swarm centre of mass. Where the swarm centre of mass at time t is calculated as

$$\hat{\boldsymbol{x}} = \frac{1}{n_s}\sum_{i=1}^{n_s} x_i\,(t) \tag{7}$$

Large diversity values indicate large distances between particles and explorative behaviour of the swarm. Small diversity values indicate the convergence of particles and suggest exploitative behaviour of the swarm.

B. *The Quality of the Solution*

The minimum at the global best position found by the swarm was used as a measure of the quality of the solution. Smaller minimum values indicate better PSO performance. The global best position found by the swarm can be compared to the known global minimum of the benchmark function to determine the effectiveness of a PSO implementation.

C. *The Average Velocity Magnitude of Particles*

The average velocity magnitude of particles gives an indication of the likelihood of a particle leaving the search space. Particles with large velocities have large position updates according to (2), which results in a higher probability of the particle leaving the boundaries of the search space [1, p. 304].

D. *The Percentage of Particles to leave the search space*

Eberhart and Kennedy recommended velocity clamping as a tool for limiting rapid velocity growth and reducing the number of particles that leave the search space [2]. By measuring the percentage of particles that leave the search space at each iteration, the efficacy of velocity clamping in preventing particles from leaving the search space can be observed.

V. Results & Discussion

This section presents empirical results and observations of the effect of velocity clamping for converging and non-converging choices of control parameters.

A. *Non-Converging Control Parameters*

For non-converging choices of control parameters A and C, velocity clamping was found to improve performance across all measures.

a) Diversity

For all five benchmark functions, the diversity of the swarm exploded to infinity when velocity clamping was not applied. This indicates that instead of converging to a global best point, the swarm diverged. It should be noted that all values of k produced good results when compared with the results for no velocity clamping. $k = 0.1$ produced the best results, followed by $k = 0.3$ and then $k = 0.5$. Fig. 1. Shows the diversity of the Sphere benchmark function over time for control parameter choice A, with and without velocity clamping. This graph shows the efficacy of velocity clamping in preventing the explosion of swarm diversity. The Styblinski Tank, Schwefel, Brown and Powell benchmark functions produce similar results for control parameter choices A and C.

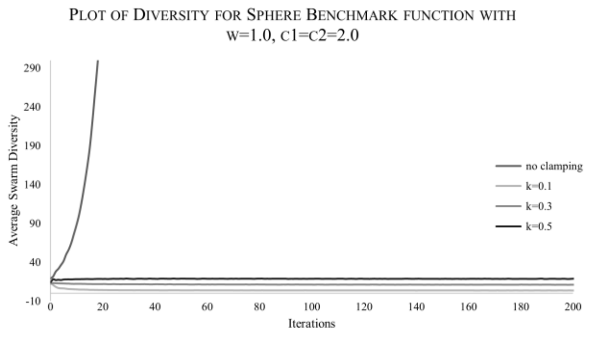

Fig. 1. Average Swarm Diversity over time for Sphere benchmark function (Averaged over 20 Independent Runs, n=30, j=30)

When combination A of control parameters was used, the Styblinski Tank, Powell, Brown and Sphere benchmark functions produced similar results with final diversities in the ranges $[1.528795, 7.298645]$, $[2.467041, 16.71813]$, and $[3.649122, 24.39041]$ for $k = 0.1$, $k = 0.3$ and $k = 0.5$ respectively. However, the Schwefel benchmark function had significantly worse diversities of 1369.925, 1667.333 and 2004.964 for $k = 0.1$, $k = 0.3$ and $k = 0.5$ respectively, although these diversities were still better than the diversity for no clamping, which approached infinity.

Combination C of control parameters produced similar results to combination A. The Styblinski Tank, Brown, Sphere and Powell benchmark functions produced a final diversity in the ranges $[1.326308, 6.981867]$, $[2.163132, 10.74143]$ and $[3.018984, 16.46286]$ for $k = 0.1$, $k = 0.3$ and $k = 0.5$ respectively. The Schwefel benchmark function produced final diversities of 1373.708, 1613.331 and 1832.311 for $k = 0.1$, $k = 0.3$ and $k = 0.5$. The best diversity for all benchmark functions was obtained when k was chosen to be 0.1.

The data shows that while both choices A and C of control parameters produce reasonable results when velocity clamping is applied, choice C produces better results. Table III summarises the percentage improvement in final diversity between choices A and C.

TABLE III. PERCENTAGE IMPROVEMENT OF BENCHMARK FUNCTIONS FOR CONTROL PARAMETER CHOICES A AND C

k	**Percentage Improvement by Benchmark Function**				
	Styblinski Tank	***Powell***	***Brown***	***Sphere***	***Schwefel***
0.1	55.1892%	5.3986%	23.1491%	23.1784%	-0.2766%
0.3	41.6166%	12.3188%	23.8552%	23.0085%	3.2388%
0.5	32.5030%	17.2682%	24.2903%	24.2100%	8.6113%

The standard deviation (σ) of the diversity measure for all benchmark functions and non-convergent choices of control parameters tended towards a consistently small value <1 when velocity clamping was applied. However, when velocity clamping was not applied, the standard deviation tended towards infinity. This indicates a large amount of inconsistency in the diversity swarms when velocity clamping was not applied. Furthermore, when velocity clamping was applied, the PSO simulation performed consistently over the 20 independent runs, as indicated by the small standard deviation.

b) Quality of the Solution

For non-converging control parameter choices and no velocity clamping, the quality of all benchmark functions remained constant over time, indicating that the swarm was unable to find new global best positions that significantly improved on any prior global best solutions. When velocity clamping was applied to the PSO with non-converging control parameters, the quality of solutions improved with time, as shown by Fig. 2.

Combination A of control parameters produced comparative quality results for all benchmark functions. $k = 0.1$ produced a better overall quality in addition to faster convergence than other values of k. For control parameter choice A, Table IV summarises the best global best position found by PSO without velocity clamping and with velocity clamping for different values of k. Table IV shows that the

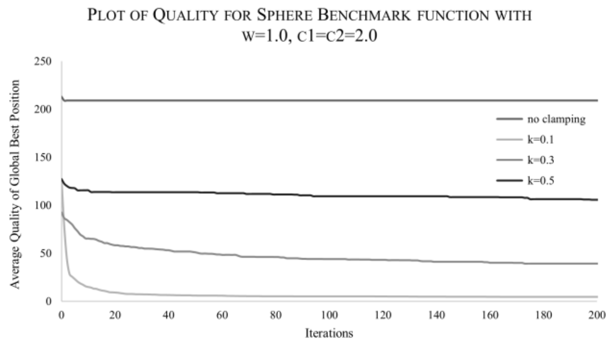

Fig. 2. Average Quality of Global Best Solution over time for Sphere benchmark function (Averaged over 20 Independent Runs, n=30, j=30)

global best position found when velocity clamping is applied is significantly lower than the global best position found by PSO without velocity clamping. This indicates that PSO performs better with non-convergent control parameters when velocity clamping is applied.

TABLE IV. GLOBAL BEST VALUE FOR CONTROL PARAMETERS A

Clamping	**Benchmark Function**				
	Styblinski Tank	***Schwefel***	***Brown***	***Powell***	***Sphere***
none	-295.83	1.16×10^4	8.3×10^{16}	2.150	209.060
$k = 0.1$	-997.08	7.88×10^3	1.390	0.000	3.110
$k = 0.3$	-776.60	8.76×10^3	12.573	0.013	28.660
$k = 0.5$	-655.50	9.5×10^3	259.241	0.139	77.654
known min	-1174.97	0	0	0	0

When control parameters were set according to combination C, PSO produced the best results when velocities were clamped with a k value of 0.1, followed by 0.3 and then 0.5. Table V shows the average value of the global best position found after 5000 iterations when PSO with various clamping strategies and control parameters C. Table V indicates that velocity clamping improved the quality of the global best position for all benchmark functions. It is worth noting that there is no significant trend in performance between control parameter choices A and C. The Styblinski Tank benchmark function produces a higher quality solution when $k = 0.1$ for choice A than for choice C, but when $k = 0.3$ and $k = 0.5$, choice C produces a better quality solution. For the Schwefel benchmark function, choice A of control parameters consistently produces higher quality results than choice C. However, the Brown and Powell benchmark functions consistently perform better under choice C control parameters. The Sphere benchmark function performs better with velocity clamping using control parameters C. There is also no significant correlation between quality of solution and modality of benchmark function.

TABLE V. GLOBAL BEST VALUE FOR CONTROL PARAMETERS C

Clamping	**Benchmark Function**				
	Styblinski Tank	***Schwefel***	***Brown***	***Powell***	***Sphere***
none	-250.875	11862.01	9.2×10^{15}	2.060	224.873
$k = 0.1$	-960.181	7244.173	0.761	6.6×10^{-5}	1.724
$k = 0.3$	-845.64	8125.516	6.227	0.003	15.501
$k = 0.5$	-733.171	9062.823	37.152	0.030	42.367
known min	-1174.97	0	0	0	0

The standard deviation of the solution quality was a large, constant number when velocity clamping was not applied for control parameters A and C. This indicates that global best solutions were found early on and that each independent execution of the PSO found vastly different solutions. It can be concluded that PSO performed very poorly for control parameter choices A and C when velocity clamping was not used. When velocity clamping was applied, the standard deviation of solution quality was significantly lower than when velocity clamping was not applied. The value of σ was shown to be heavily dependent on the value of k, with $k = 0.1$ producing the smallest standard deviation for all benchmark functions. This indicates that PSO performed consistently by finding a similar global best position at each iteration. The rate at which the standard deviation converged to a constant value gives an indication of how quickly PSO converged to a global best position. The rate of convergence of σ was found to be dependent on the choice of benchmark function and value of k.

c) Percentage of Particles to Leave the Search Space

For all benchmark functions and non-convergent choices of control parameters, almost 100% of particles leave the search space during every iteration when velocities are not clamped – as illustrated by Fig. 3. When velocities are clamped, the percentage of particles to leave the search space during each iteration can be reduced, however, the benefit of velocity clamping is highly dependent on the choice of k. For all benchmark functions, PSO with velocity clamping and $k = 0.5$ performed worse than PSO without velocity clamping.

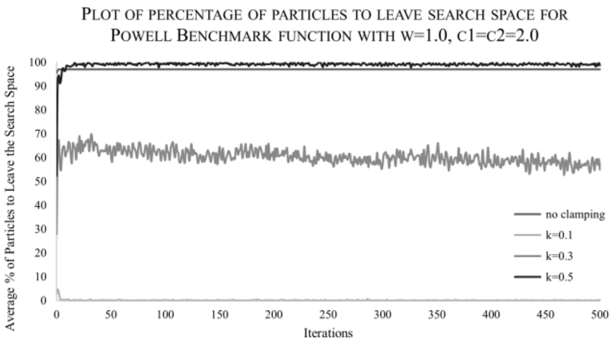

Fig. 3. Average Percentage of Particles to Leave the Search Space over time for Powell benchmark function (Averaged over 20 Independent Runs, n=30, j=30)

For non-converging choice of control parameters A, velocity clamping with $k = 0.1$ produced the best results across all benchmark functions. For the Brown, Powell, Sphere and Styblinski Tank benchmark functions, velocity clamping with $k = 0.1$ caused the percentage of particles that left the search space over time to converge to 0%. However, for the Schwefel benchmark function, the percentage of particles that left the search space per iteration only converged to 37%. When k values of 0.3 and 0.5 are chosen, the percentage of particles to leave the search space per iteration is significantly higher for all benchmark functions. The performance of velocity clamping with $k = 0.3$ is highly dependent on the choice of benchmark function. In the best case, $k = 0.3$ causes the percentage of particles to leave the search space to converge to 9% for the Sphere benchmark function. In the worst case, velocity clamping with $k = 0.3$ causes the percentage of particles to leave the search space per iteration to approach 100% for the Brown benchmark function. In all cases, velocity clamping with $k = 0.5$ performs worse than when PSO is conducted without velocity clamping. For both PSO with no velocity clamping and PSO with velocity clamping and $k = 0.5$, the percentage of particles that leave the search space during each iteration tends towards 100%.

Choosing control parameters according to C produced similar results to A. For the Styblinski Tank, Sphere, Brown and Powell benchmark functions, when velocity clamping was applied with $k = 0.1$, the percentage of particles to leave the search space converged to 0% and for the Schwefel benchmark function, to 15%. When $k = 0.3$, the performance was considerably less consistent with some benchmark functions performing as well as 1% and others converging to 94% of particles leaving the search space. For the Styblinski Tank, Schwefel, Brown and Powell benchmark functions, PSO with no velocity clamping and PSO with velocity clamping and $k = 0.5$ performed similarly, with close to 100% of particles leaving the search space during each iteration. However, for the sphere benchmark function, velocity clamping with $k = 0.5$ limited the percentage of particles that leave the search space to 70%, which is still better than the almost 100% of particles that left the search space when no velocity clamping was applied.

The data indicates that for the correct choice of k, velocity clamping prevents the percentage of particles that leave the search space from exploding to 100%. In all cases, smaller values of k, which indicate tighter velocity clamping, have a larger influence on the percentage of particles that leave the search space.

When velocity clamping was not applied, the standard deviation of the percentage of particles to leave the search space converged to a small constant quickly. This is because the percentage of particles leaving the search space approached 100% in a small number of iterations and it did this consistently across all independent runs of the PSO. When velocity clamping was applied, the standard deviation also converged to a small constant quickly. This reflects that the percentage of particles leaving the search space quickly converged to some constant as indicated by Fig. 3. It also suggests that this constant was similar for all independent runs of the PSO simulation.

d) Average Velocity Magnitude

For all non-convergent choices of control parameters, the average velocity magnitude of particles exploded to infinity when velocity clamping was not applied. Fig. 4. Illustrates that velocity clamping prevented the velocities of particles from exploding to infinity.

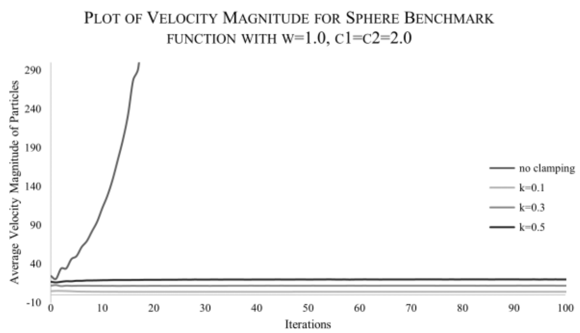

Fig. 4. Average Velocity Magnitude of Particles over time for Sphere benchmark function (Averaged over 20 Independent Runs, n=30, j=30)

When velocity clamping is applied, the magnitudes of velocities of particles converge to a value that is a fraction of the search space. By preventing the explosion of velocities, the step size of particles is also reduced according to (2). In all cases, the velocity magnitude when $k = 0.1$ is smaller than the velocity magnitude when $k = 0.3$ which is smaller than the velocity magnitude when $k = 0.5$. This is consistent with the clamping method proposed by Engelbrecht [1, p. 304] which clamps velocities at a fraction k of the size of the search space. Both choices of control parameters A and C produce similar results. In all cases where velocity clamping is not applied, the velocity magnitude of particles spikes to infinity more rapidly when control parameters A are chosen over control parameters C. However, the difference is negligible as both choices of control parameters result in the velocity magnitude of particles rapidly approaching infinity.

When velocity clamping is not applied, the standard deviation of the velocity magnitude quickly explodes to infinity. This indicates that particles started with similar velocities across all independent runs of the PSO simulation, but that the velocities of these particles increased at considerably different rates across independent runs, resulting in increasingly large standard deviations with time. When velocity clamping was applied, σ quickly approached a small constant value. This is because velocity clamping sets velocities to a constant V_{max} value, which is consistent across all independent runs of the PSO simulation.

B. Converging Control Parameters

For converging choices of control parameters B and D, velocity clamping was found to have no significant influence across all four performance measures. For several cases, PSO performed better when velocity clamping was not applied.

a) Diversity

For all five benchmark functions, the diversity of the swarm converged with and without the application of velocity clamping. For the Sphere, Powell, Styblinski Tank and Brown benchmark functions, the diversity of the swarm converged to the same value, regardless of whether velocity clamping was applied. For all cases where velocity clamping was applied, the diversity of the swarm decreased faster than when velocity clamping was not applied, as shown in Fig. 5.

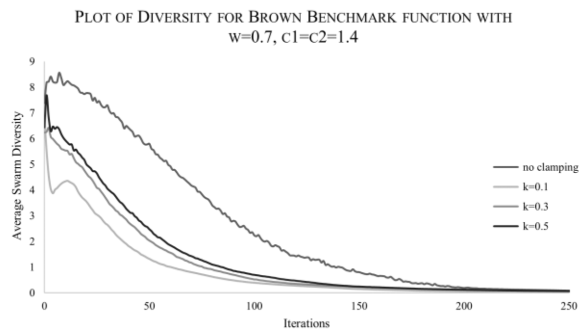

Fig. 5. Average Diversity of Swarm over time for Brown benchmark function (Averaged over 20 Independent Runs, n=30, j=30)

When control parameters were set according to B, the diversity of the swarm converged to values close to 0 for the Brown, Powell and Sphere benchmark functions. The swarm diversity converged to an average value of 14.78 for the Styblinski Tank benchmark function and to an average value of 1409.44 for the Schwefel benchmark function.

When PSO was performed using control parameters D, the Brown, Powell and Sphere benchmark functions had swarm diversities which approached values close to 0. The Styblinski Tank benchmark function approached an average swarm diversity value of 14.76 and the Schwefel benchmark function converged to an average of 1452.32.

This indicates that PSO performs equally well on swarm diversity with and without velocity clamping for convergent choices of control parameters.

The standard deviation of the diversity of the swarm converges to 0 because the diversity of the swarm approaches 0 as the exploitative behaviour of the swarm increases. The rate at which the standard deviation approaches 0 indicates how quickly the diversity of the swarm approaches 0. When the standard deviation at each iteration for different values of k is plotted on a scatter plot, it becomes clear that σ oscillates during early iterations of the PSO. This is because of the explorative behaviour of the swarm during early iterations of PSO, where particles move away from the global best and are then attracted back towards the global best by the social component of the velocity update equation. This behaviour produces inconsistency in the swarm diversity found during different independent runs of the simulation. However, once particles start to converge towards a single point, the independent runs of the simulation produce more stable diversity values, as is indicated by the convergence of σ.

b) Quality of the Solution

For both choice B and D of control parameters, PSO produced equally good global best positions with and without velocity clamping, as shown in Fig. 6. Although in some cases, velocity clamping resulted in the swarm finding a high-quality solution in fewer iterations.

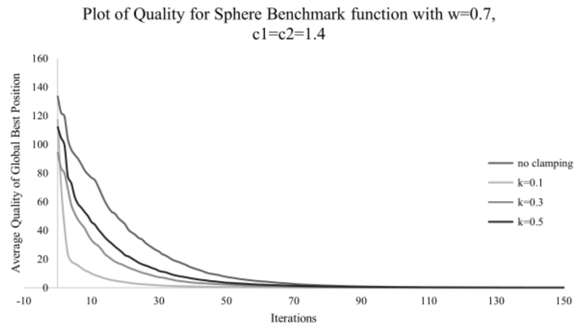

Fig. 6. Average Quality of Global Best Solution over time for Sphere benchmark function (Averaged over 20 Independent Runs, n=30, j=30)

Table VI shows the quality of PSO solutions for control parameter choice B. The Styblinski Tank, Brown, Powell and Sphere benchmark functions approach global best values close to the known global minimum of the function. The Styblinski Tank, Schwefel, Brown and Sphere benchmark functions all achieve the best quality solution when velocity clamping is applied with $k = 0.5$. The Powell benchmark function achieves the best quality solution when velocity clamping is applied with $k = 0.3$. However, the difference between qualities obtained when the optimal velocity clamping is applied and when no velocity clamping is applied is negligible.

TABLE VI. GLOBAL BEST VALUE FOR CONTROL PARAMETERS B

Clamping	Benchmark Function				
	Styblinski Tank	*Schwefel*	*Brown*	*Powell*	*Sphere*
none	-1014.53	5273.01	0.00678	8.5×10^{-17}	7.7×10^{-7}
$k = 0.1$	-998.276	6827.793	1.5×10^{-4}	7.2×10^{-17}	1.9×10^{-7}
$k = 0.3$	-1005.34	5520.697	1.8×10^{-4}	1.0×10^{-21}	1.9×10^{-6}
$k = 0.5$	-1019.48	5115.633	5.0×10^{-4}	5.7×10^{-18}	7.0×10^{-8}
known min	-1174.97	0	0	0	0

Table VII contains the quality of the global best position found by PSO using control parameters D after 5000 iterations. The data shows that PSO with velocity clamping and $k = 0.5$ produces the highest quality global best position for the Styblinski Tank and Brown benchmark functions. PSO achieves the best quality solution when no velocity clamping is applied for the Powell and Schwefel benchmark functions. The highest quality solution for the Sphere benchmark function is found when PSO with velocity clamping and $k = 0.1$ is used. The difference between the best global minimum found for each function using PSO, and minimum found using PSO without velocity clamping, is negligible. The data indicates that the optimal choice of control parameter is dependent on the choice of benchmark function and furthermore suggests that there exists an optimal control parameter choice for each benchmark function such that PSO without velocity clamping produces the best quality result. It should be noted that there is no correlation between the modality of the benchmark function and the quality of the solution found by PSO with or without velocity clamping.

TABLE VII. GLOBAL BEST VALUE FOR CONTROL PARAMETERS D

Clamping	Benchmark Function				
	Styblinski Tank	*Schwefel*	*Brown*	*Powell*	*Sphere*
none	-1020.89	5053.937	1.4×10^{-29}	2.8×10^{-55}	5.4×10^{-30}
$k = 0.1$	-996.862	6149.884	2.0×10^{-27}	3.4×10^{-50}	7.8×10^{-33}
$k = 0.3$	-1004.64	5359.996	4.3×10^{-30}	2.6×10^{-54}	2.0×10^{-29}
$k = 0.5$	-1032.2	5316.432	1.6×10^{-34}	9.0×10^{-49}	5.2×10^{-32}
known min	-1174.97	0	0	0	0

From these results, it is evident that the contribution of velocity clamping to the quality of a global best position is negligible when the control parameters are chosen to be convergent. It should also be noted that for all cases, control parameter choices B and D without velocity clamping produced higher quality results than any of the results produced for control parameter choices A and C.

When control parameters are chosen such that particles converge, the standard deviation of the quality of the global best position converges. The rate at which σ converges is an indicator of how quickly PSO finds the final global best position and the magnitude of σ indicates how consistently PSO finds the same global best position during independent runs of the simulations. The value of σ is significantly smaller for control parameter choices B and D than for A and C. This is because PSO approaches the same good global best position consistently when convergent control parameters are chosen. However, when non-convergent control parameters are chosen, it finds different low-quality solutions during early iterations of PSO and fails to improve on these.

c) Percentage of Particles to Leave the Search Space

Fig. 7. Shows that the percentage of particles that leave the search space per iteration decreases to 0% when the control parameters are chosen to be convergent. Comparable results were found for all benchmark functions when PSO was performed using control parameter choices B and D.

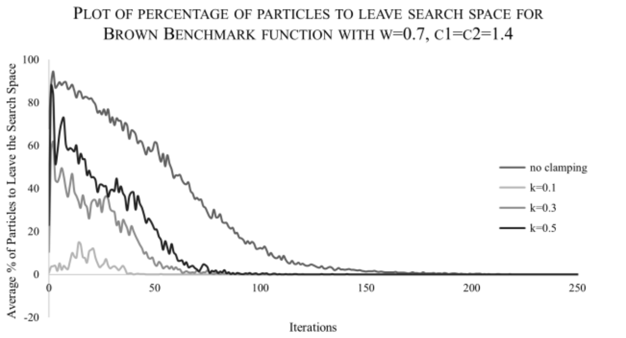

Fig. 7. Average Percentage of Particles to Leave the Search Space over time for Brown benchmark function (Averaged over 20 Independent Runs, n=30, j=30)

For both control parameter choices B and D, PSO with velocity clamping caused the percentage of particles that left the search space per iteration to decrease more rapidly. However, for all benchmark functions, the percentage of particles to leave the search space approached a number close to 0% in under 800 iterations, which indicates that the improvement achieved by velocity clamping is negligible. It is also worth noting that the worst quality results were achieved when velocity clamping caused the percentage of particles leaving the search space to decrease to 0% almost immediately.

Since the percentage of particles that leave the search space approaches 0% with time, the standard deviation of the percentage of particles also approaches 0. This confirms that PSO performs consistently over all independent runs of the simulation. The oscillatory nature of the standard deviation as shown in Fig. 8. reflects the explorative behaviour of the swarm during early iterations of PSO for which there is more variation between independent runs of the simulation. However, as the swarm starts to behave more exploitatively, the percentage of particles leaving the search space approaches 0% for all independent runs and thus the standard deviation also converges to 0.

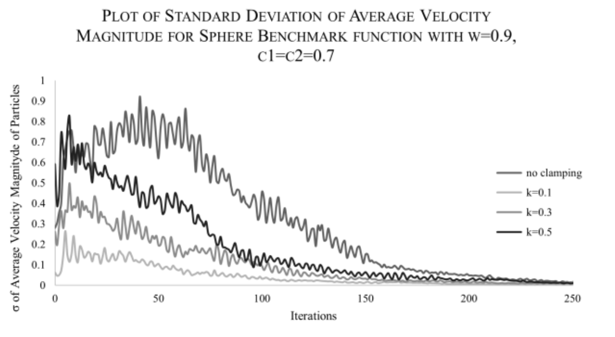

Fig. 8. Standard Deviation of Average Velocity Magnitude of Particles over time for Sphere benchmark function (Averaged over 20 Independent Runs, n=30, j=30)

d) Average Velocity Magnitude

Fig. 9. Shows that the velocity magnitude converges to 0 for all variations of velocity clamping on PSO when the control parameters are chosen to be convergent.

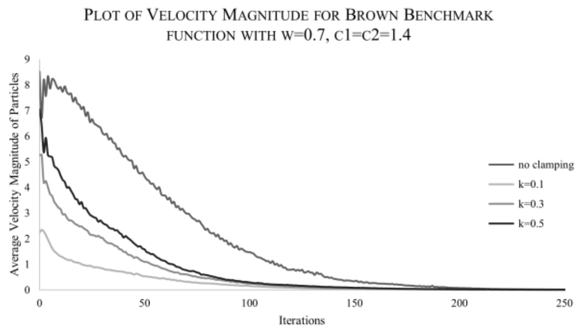

Fig. 9. Average Velocity Magnitude of Particles over time for Brown benchmark function (Averaged over 20 Independent Runs, n=30, j=30)

These findings indicate that the particles converge to a single point which is consistent with the findings of Poli and Broomhead [6]. The velocity magnitude converges to 0 fastest for large values of k and slowest when no velocity clamping is applied. However, larger velocity magnitudes indicate better explorative abilities of the swarm and so it is not desirable to have velocity magnitudes that decrease to 0 too quickly.

The velocity magnitude of a particle approaches 0 as particles converge to a global best solution. Consequently, the standard deviation of average velocity magnitude also approaches 0. The standard deviation of average velocity magnitude increases during early iterations of PSO with convergent control parameters and then decreases to 0. There is more variance among independent runs of the PSO simulation during the explorative phase since particles need to have larger position updates and need larger variance in velocity magnitudes to do this. During the exploitative phase, the velocity magnitude of all particles in a swarm approaches 0, resulting in more consistency among independent PSO simulations.

VI. CONCLUSION

This paper aims to establish whether velocity clamping is necessary in particle swarm optimisation (PSO) when the control parameters w, c_1 and c_2 are chosen such that the convergence conditions in equation (5) are satisfied.

This research found that when control parameters are chosen such that particles do not converge to a single point, as with control parameter choices A and C, velocity clamping is necessary. However, when control parameters are chosen according to (5) such that particles converge to a single solution, as with control parameter choices B and D, velocity clamping produces negligible, if any, improvements on PSO.

When velocity clamping is not applied to PSO and control parameters are chosen to be non-converging, the velocity magnitude of particles explodes to infinity, causing the diversity of the swarm to explode to infinity and almost all particles to leave the search space at each iteration. This hinders the exploitative ability of the swarm and results in the swarm finding poor quality global best positions. In this case, applying velocity clamping prevents the explosion of velocity magnitudes, decreasing the step size of particles and preventing the explosion of swarm diversity. Consequently, fewer particles leave the search space with every iteration and the swarm has a higher probability of finding a good quality global best position.

When PSO is conducted with control parameters that result in the convergence of the swarm, the velocity magnitude of particles does not explode to infinity and the diversity of the swarm eventually converges. This means that the percentage of particles that leave the search space per iteration decreases with time, and the swarm has a high probability of identifying a good quality global best position.

It should be noted that the quality of solutions found for all five benchmark functions was better when control parameters were chosen to converge, and no velocity clamping was applied than when non-converging control parameters were chosen, and velocity clamping was applied. Based on these findings, this research recommends that PSO be implemented with a control parameter choice that converges according to (5), rather than with velocity clamping.

The results did not indicate any clear correspondence between performance of PSO with velocity clamping and the modality of the benchmark function.

REFERENCES

[1] A. P. Engelbrecht, "Particle Swarm Optimization," in *Computational Intelligence. An Introduction.*, Pretoria, John Wiley & Sons, Ltd, 2007, pp. 289 - 341.

[2] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory.," in *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 1995.

[3] Y. Shi and R. Eberhart, "A modified particle swarm optimizer.," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, 1998.

[4] R. Poli and D. Broomhead, "Exact Analysis of the Sampling Distribution for the," in *GECCO07: Genetic and Evolutionary Computation Conference*, London, 2007.

[5] A. Thevenot, "Optimization & Eye Pleasure: 78 Benchmark Test Functions for Single Objective Optimization," 31 12 2020. [Online]. Available: https://towardsdatascience.com/optimization-eye-pleasure-78-benchmark-test-functions-for-single-objective-optimization-92e7ed1d1f12. [Accessed 17 03 2022].