Is Velocity Clamping Necessary for 30-Dimensional Spaces?

SD van Deventer
20273401
20273401@sun.ac.za

***Abstract*—This report describes how velocity clamping influences particle swarm optimization (PSO) and whether it is still necessary. It has been shown that a PSO will converge provided that the control parameters, i.e. w, c_1 and c_2 conform to the convergence condition. Naturally a new hypothesis can be constructed: velocity clamping is no longer necessary. This hypothesis was put to the test by running a gbest PSO against a series of tests. These tests were designed to determine whether velocity clamping approaches provide an improvement over no-clamping approaches for different control parameter values. It was observed that velocity clamping does not provide a significant improvement, provided that the control parameters are chosen in such a way that they satisfy the convergence conditions.**

I. Introduction

This section serves as an introduction to the problem at hand and provides insight as to how a conclusion was drawn.

Velocities have a tendency to explode or grow continuously. In the context of particle swarm optimisation this results in potentially divergent behaviour. Velocity clamping was introduced in an effort to combat this effect by limiting the magnitude of the velocity of every particle in the swarm. Velocity clamping has been a divisive topic within particle swarm research. Many researchers believe it is sufficient to limit the velocity of each particle, but it has proven to be no more than a treatment of the symptoms and does not actually provide a compelling solution. Clamping approaches can still result in divergent behaviour and could even be dependent on the number of iterations that the PSO runs for. More over, clamping introduces more variables that have to be calibrated differently for each problem.

It seems obvious that a different solution was needed. It was derived by Poli [1] that the inertia (w), cognitive (c_1) and social (c_2) weights of a particle could be used to guarantee convergence. The convergence condition was introduced:

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

It was shown that the particle swarm will converge by selecting the control parameters in such a way that they conform to this condition. The question therefore arose, is it still necessary to employ velocity clamping if convergence can be guaranteed?

To answer this question this report evaluates different clamping approaches, including no-clamping, against different control parameter selections. Six different control parameter combinations were used of which only some satisfy the convergence condition. Five different benchmark functions were used in the execution of the PSO. Empirical analysis was then done to evaluate the effects of the velocity clamping techniques. The results show an insignificant difference between approaches provided that the control parameters satisfy the convergence condition.

II. Background

This section gives the reader some background knowledge about the concepts being discussed throughout the rest of the report. It also provides a high level description of the algorithms used within this research.

Particle swarm optimisation is an optimisation technique developed by Kennedy and Eberhart [2]. It is an evolutionary computation technique inspired by natural social behaviour. Much like a flock of birds a swarm consists of particles moving around in n-dimensional space. As these particles move about they attempt to move the swarm to an optimal position (local optima). A pseudo-code representation of a global best PSO is given by algorithm 1.

Algorithm 1 *gbest* PSO

```
repeat
    for each particle i = 1,...,n_s do
        //Set the personal best position
        if f(x_i) < f(y_i) then
            y_i = x_i;
        end if
        //Set the global best position
        if f(y_i) < f(ŷ) then
            ŷ = y_i;
        end if
    end for
    for each particle i = 1,...,n_s do
        Update velocity;
        Update position;
    end for
until stopping condition is true
```

Every particle in the swarm has its own velocity and position in n-dimensional space. A particles velocity consists of three different components [3]:

- Previous velocity, $\omega \mathbf{v}_i(t)$
- Cognitive component, $c_1\mathbf{r}_1(\mathbf{y}_i - \mathbf{x}_i)$
- Social component, $c_2\mathbf{r}_2(\hat{\mathbf{y}}_i - \mathbf{x}_i)$

The constant ω denotes the inertia weight. This is important as it prevents the particle from rapidly changing its direction by continuing in the direction it was previously moving in. The constant c_1 denotes the cognitive weight. This component simulates the memory of the particle as it remembers its personal best position and is always drawn back to it. The constant c_2 denotes the social weight. This means that the particle is also drawn to the neighbourhood best position. Using these components we are left with a velocity that takes into account the direction that the particle was previously moving in, the personal best position of the particle and the neighbourhood best position. The neighbourhood best position refers to the best position produced by a particle within the neighbourhood.

All basic approaches to particle swarm optimisation suffer from velocity explosion. What is meant by this is the behaviour in which the velocity of the particles continuously grow towards infinity. This results in divergent behaviour for the swarm and causes the particles to move into infeasible space. Velocity clamping was introduced in an attempt to keep velocities small enough for the particles to not diverge and stay in feasible space. For a long time this was seen as the best solution to the problem, but divergent behaviour still occurred.

Velocity clamping limits the size of velocities by introducing a maximum value namely V_{max}. Many different approaches to velocity clamping were introduced. Some of these include basic velocity clamping, normalised clamping [5], the exponentially decaying V_{max} strategy, and the dynamically changing V_{max} strategy. Other strategies exist but are not considered in this report.

The following book on computational intelligence is recommended to the interested reader [3].

III. Implementation

This section provides a detailed discussion of the approaches taken to solve the problem at hand. Algorithms and formulae that were used in the implementation will be explored in detail.

The question under consideration is as follows: *Is velocity clamping necessary?* To answer this question a standard global best particle swarm optimisation was implemented. Five equations were selected to be evaluated by the PSO and six different control parameter combinations were used.

A. Particle swarm optimisation

The gbest PSO makes use of the star topology as illustrated in figure 1(a) [4]. This implies that every particle in the swarm is connected to every other particle in the swarm. The neighbourhood best position is therefore called the global best position and is shared among all particles. Unlike the local best PSO, that uses the ring topology, every particles personal best position is taken into account when calculating the new global best position. The global best position is denoted by $\hat{\mathbf{y}}$ and is used when calculating the social component of the velocity of each particle. The velocity of a particle changes after every iteration that the particle swarm runs for and is then used to calculate the particles new position. The velocity is calculated by equation 1. Velocities are initialised to zero.

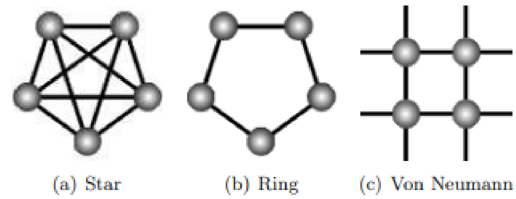

Fig. 1. Topologies

$$\mathbf{v}_i(t+1) = \omega\mathbf{v}_i(t) + c_1\mathbf{r}_1[\mathbf{y}_i(t) - \mathbf{x}_i(t)] + c_2\mathbf{r}_2[\hat{\mathbf{y}}_i(t) - \mathbf{x}_i(t)] \quad (1)$$

It is clear that the velocity is dependent on three main components which are the inertia, social and cognitive components respectively. The social and cognitive components also include random numbers. These random numbers are sampled from the uniform distribution with $\mu = 0$ and $\sigma = 1$. Each dimension receives a new random number. This serves the purpose of adding random noise to the equation. Each of the components have a unique weight namely, ω, c_1 and c_2. These weights are referred to as the control parameters. The relationship between the control parameters will be discussed in a later section.

Each particle also has a position that is calculated by equation 2. $\mathbf{x}_i(0) \sim U(\mathbf{x}_{min}, \mathbf{x}_{max})$.

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t) \quad (2)$$

B. Velocity clamping

There are many different implementations of velocity clamping, but for the purposes of this report only the following were considered: basic clamping, normalised clamping, exponentially decaying V_{max} and dynamically changing V_{max}. V_{max} refers to the maximum size of the velocity.

$$\mathbf{v}_i(t+1) = \begin{cases} \mathbf{v}_i(t) & \text{if } |\mathbf{v}_i(t)| < \mathbf{V}_{max} \\ \text{sgn}(\mathbf{v}_i)\mathbf{V}_{max} & \text{if } |\mathbf{v}_i(t)| \geq \mathbf{V}_{max} \end{cases} \quad (3)$$

Velocity clamping attempts to limit the velocity of a particle to combat velocity explosion which causes particles to move into infeasible space. Equation 3 shows how $\mathbf{V}_{max}$ is used to calculate the new velocity.

The first velocity clamping technique considered is basic velocity clamping. $\mathbf{V}_{max}$ is only calculated at the beginning of the simulation and does not change during run time. Equation 4 shows how the clamping is initialised. The constant k has a value between 0 and 1 and denotes a percentage of the search space.

$$\mathbf{V}_{max} = k \times (\mathbf{x}_{max} - \mathbf{x}_{min}) \quad (4)$$

Normalised clamping makes use of the velocity magnitude to limit the velocity. The aim of this approach is to preserve the search direction of the particle [5]. The way in which the velocity is updated is therefore changed and is shown by equation 5.

$$\mathbf{v}_i(t+1) = \begin{cases} \mathbf{v}_i(t+1) & \text{if } ||\mathbf{v}_i(t+1)|| \leq \mathbf{v}_{max} \\ \frac{\mathbf{v}_{max}}{||\mathbf{v}_i(t+1)||}\mathbf{v}_i(t+1) & \text{if } ||\mathbf{v}_i(t+1)|| > \mathbf{v}_{max} \end{cases} \tag{5}$$

Exponentially decaying $\mathbf{V}_{max}$ [3]. The constant α is a problem dependant variable and needs to be tuned for every new problem. Equation 6 shows how $\mathbf{V}_{max}$ is updated.

$$\mathbf{V}_{max}(t+1) = \left(1 - \left(\frac{t}{n_t}\right)^{\alpha}\right)\mathbf{V}_{max}(t) \tag{6}$$

It can be noted that an increase in the value of α will result in a larger percentage of $\mathbf{V}_{max}(t)$ to be used per iteration and as time grows $\mathbf{V}_{max}(t)$ will tend towards zero.

Dynamically changing $\mathbf{V}_{max}$ [3]. When there has been no improvement of the global best score over the past τ runs, $\mathbf{V}_{max}$ is updated. Equation 7 shows how $\mathbf{V}_{max}$ is updated.

$$\mathbf{V}_{max}(t+1) = \begin{cases} \beta\mathbf{V}_{max}(t) & \text{if } f(\hat{\mathbf{y}}(t)) \geq f(\hat{\mathbf{y}}(t-t')) \\ \mathbf{V}_{max}(t) & \text{otherwise} \end{cases} \tag{7}$$

$\forall\ t' = 1, ..., \tau$. β decreases from 1.0 to 0.01.

This has the result of setting $\mathbf{V}_{max}$ to a lower percentage of what it previously was. The percentage decreases linearly with β.

C. Control parameters

As stated earlier the weights used in the calculation of the velocity, namely ω, c_1 and c_2 form the control parameters. It was shown that these parameters have a very useful relationship in terms of convergence. Equation 8 shows this relationship and is called the convergence condition [1].

$$c_1 + c_2 < \frac{24(1-\omega^2)}{7-5\omega} \tag{8}$$

If the control parameters are chosen in such a way that they conform to the convergence condition, the PSO will converge. This could prove useful as it may render velocity clamping redundant.

IV. Empirical procedure

This section will explain to the reader how the aforementioned algorithms and formulae were applied to answer the question at hand.

A. Benchmark functions

To test the effectiveness of the gbest PSO with the different approaches a set of benchmark functions is needed. This report evaluates five different benchmark functions, equations [9-13]. Each of these benchmark functions have differing characteristics and search spaces. This is important as it will also test the scalability of the approaches.

f_1, the absolute value function, defined as

$$f_1(\mathbf{x}) = \Sigma_{j=1}^{n_x}|x_j| \tag{9}$$

with each $x_j \in [-100, 100]$.

f_5, the elliptic function, defined as

$$f_5(\mathbf{x}) = \Sigma_{j=1}^{n_x}(10^6)^{\frac{j-1}{n_x-1}}x_j \tag{10}$$

with each $x_j \in [-100, 100]$.

f_6, the griewank function, defined as

$$f_6(\mathbf{x}) = 1 + \frac{1}{4000}\Sigma_{j=1}^{n_x}x_j^2 - \prod_{j=1}^{n_x}\cos\frac{x_j}{\sqrt{j}} \tag{11}$$

with each $x_j \in [-600, 600]$.

f_{12}, the elliptic function, defined as

$$f_{12}(\mathbf{x}) = 10n_x + \Sigma_{j=1}^{n_x}(x_j^2 - 10\cos(2\pi x_j)) \tag{12}$$

with each $x_j \in [-5.12, 5.12]$.

f_{24}, the vincent function, defined as

$$f_{24}(\mathbf{x}) = -\left(1 + \Sigma_{j=1}^{n_x}\sin(10\sqrt{x_j})\right) \tag{13}$$

with each $x_j \in [0.25, 10]$.

These benchmark functions are used to calculate the quality of a particles position.

B. Control parameters

Six different control parameter combinations are used to evaluate each of the aforementioned benchmark functions. These combinations are listed in table I as follows: the value for ω, the value for c_1, the value for c_2 and whether or not the combination conforms to the convergence condition. For example, using the first combination: $\omega = 1.0$, $c_1 = 2.0$ and $c_2 = 2.0$. Calculating the left hand side of equation 8 results in the addition of c_1 and c_2 which is equal to 4.0. Substituting the ω weight into the right hand side of the equation results in zero. Now comparing the left and right hand sides of the equation it should be clear that $4.0 \nless 0.0$ and therefore this combination does not conform to the convergence condition. The combinations were chosen such that three would cause the PSO to converge and three might cause it to diverge. It can however be expected that the non convergent combinations will converge or at least attempt to converge if the PSO is subjected to velocity clamping.

ω	c_1	c_2	Convergent
1.0	2.0	2.0	False
0.7	1.4	1.4	True
0.9	2.0	2.0	False
0.9	0.7	0.7	True
0.5	0.5	0.5	True
0.5	0.5	1.0	False

TABLE I
TABLE OF CONTROL PARAMETERS

C. Velocity clamping

The velocity clamping approaches taken are detailed in section III, subsection B. For the basic velocity clamping (equation 4) and normalised clamping (equation 5) the values of k where chosen as $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. No clamping is also included as a control measure to evaluate the effect of convergent control parameters. Each of these approaches are tested with the six different control parameter combinations and against the five different benchmark functions. It is expected that velocity clamping techniques will perform better than a no-clamping technique while the control parameters do not guarantee convergence. It is however also expected that there will be a negligible difference between the approaches provided that the convergence condition is satisfied.

D. Performance measures

To evaluate the performance of an approach four different performance measures are specified:

1) Diversity of the swarm over time.
2) Quality of the global best position.
3) The percentage of particles that leave the search space.
4) Average velocity magnitude over time.

The diversity of the swarm was calculated as the average Euclidean distance from the center of mass of the swarm. The center of mass of the swarm was viewed as the average particle position. This measure was chosen as it directly tracks how far apart particles are from each other at any given time. Divergent behaviour will result in an increase of this value over time as particles are moving away from each other. Convergent behaviour will result in a decrease of this value towards zero as this is indicative of particles moving closer to each other.

The quality of the global best position of the swarm is simply the evaluation of the global best position by the selected benchmark function. The purpose of a PSO is to improve the quality of the global best position over time. As this is a minimisation problem we can expect this value to decrease over time. If this value stays constant at a value higher than expected it can point towards early divergent behaviour or be an indication that the swarm is getting stuck in local optima.

The percentage of particles that leave the search space was calculated as the amount of particles that leave the search space over the amount of particles that are in the swarm. This is a useful measure as it shows the percentage of particles that are searching in infeasible space. This is wasted computational power as the results obtained from these particles are infeasible and should not be considered. We can expect this percentage to increase as particles diverge and to be close to zero if they converge.

The Average velocity magnitude over time is calculated. The reason for this measure is to track the velocity at which particles move at a given time. If this value becomes close to zero it can be deduced that the particles are slowing down and converging. If this value rises it shows that the particles are diverging and could show the explosive behaviour that is being studied. It is expected of the velocity magnitude to be very high within the first few iterations and then decrease over time for convergent behaviour.

E. PSO configuration

The global best PSO discussed in algorithm 1 is initialised with 30 particles. Each particle is initialised with 30 dimensions. The initialisation for the particles are depicted in algorithm 2.

Algorithm 2 Particle initialisation

for *each particle* i = 1,...,n_s **do**
  for *each dimension* $j = 1, ..., n_x$ **do**
    //Set the position and velocity
    $x_j \sim U(x_{j,min}, x_{j,max})$
    $v_j = 0$
  end for
  //Set the personal best position
  $\mathbf{y} = \mathbf{x}$
end for

In total 390 independent simulations were run to evaluate all combinations of the benchmark functions, control parameters and velocity clamping approaches. A single simulation consists of 5000 iterations that the PSO is executed for. Each simulation is executed 30 times and the average of the performance measures are taken over the 30 executions. The resulting data is captured in a *csv* file for ease of access when viewing the data graphically.

V. RESEARCH RESULTS

In this section the results obtained during the empirical process will be conveyed and discussed. It is important to note that these results are averages taken over 30 runs per simulation.

Figures 2 and 3 show the divergent and convergent behaviour of equation 9 respectively. Figure 2 is the result of a no-clamping approach with control parameters $\omega = 1, c_1 = c_2 = 2$ and show the diversity of the swarm over time. It should be noted that the y axis is $\times$1e151. This causes the diversity to seem constant for lower time intervals. Note the very sudden and significant increase in diversity as time moves past 3500 iterations. This shows that the diversity of the swarm is increasing rapidly and showcases the result of the explosive behaviour of the PSO as particles move further away from each other. This behaviour is shared by the velocity magnitude and follows the same trajectory. This result is expected when taking the convergence condition in equation 8 into account. The control parameters do not conform to this condition and is expected to diverge as the PSO is not subjected to velocity clamping.

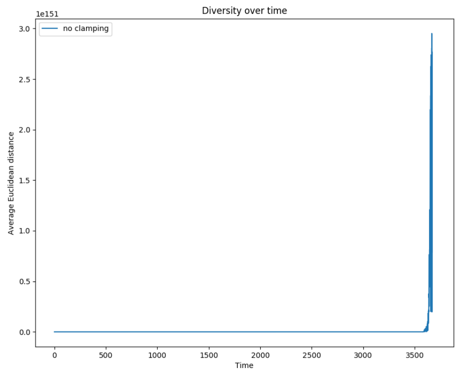

Fig. 2.

In contrast to figure 2, figure 3 showcases the convergent behaviour of the PSO due to the control parameter combination. The parameters were $\omega = 0.7, c_1 = c_2 = 1.4$ and no clamping was involved. These control parameters meet the convergence condition and is therefore expected to cause the PSO to converge.

It can be noted that the function converges quickly as illustrated by the diversity's rapid decent towards zero. This shows that the particles are moving closer together. Note also the difference in the behaviour of particles at different time intervals. At an early stage all particles are far away from each other. This indicates that more of the search space is being *explored*. This behaviour is desired at this stage as it increases the possibility of finding more local optima. Then as time progress the particles move closer together, indicating that they are starting to *exploit* more. This behaviour is necessary to find more optimal solutions within local optima. This change from *exploring* to *exploiting* is the desired behaviour of a PSO.

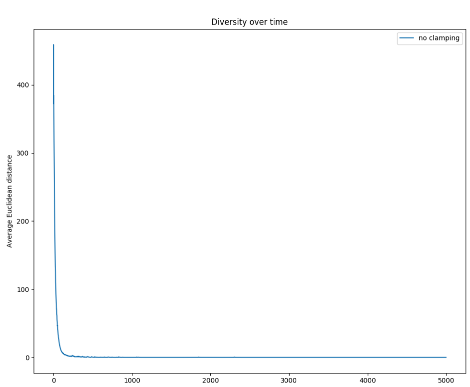

Fig. 3.

When the convergence condition is not met, velocity clamping can in some cases lead to more promising results. Figure 4 shows the difference in the global best position over time between a no-clamping approach and a dynamic clamping approach. The control parameters were chosen as $\omega = 0.9, c_1 = c_2 = 2.0$. The figure clearly shows that the dynamic clamping approach followed an almost linear decent of the global best position where as the no-clamping approach did not improve its global best position as it immediately started diverging. It can therefore be assumed that velocity clamping can have a positive effect on the quality of the global best position if the convergence condition is not met. It should however be noted that this report only considers 5000 iterations and therefore may not be consistent with tests run on more iterations.

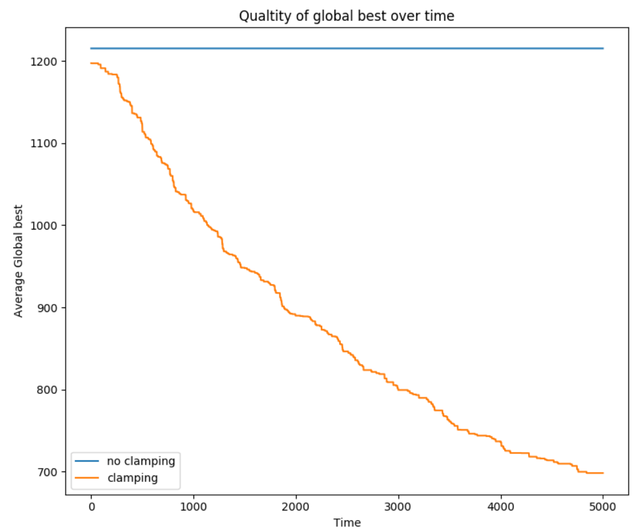

Fig. 4.

Table II shows the final average quality of the global best position, diversity and velocity magnitude of the PSO for a convergent selection of control parameters. Notice how low the diversity of the swarm becomes for all clamping approaches. There is almost no difference between the no-clamping approach and the velocity clamping approaches. Figure 5 is a graph of the diversity of the swarm as seen in table II. As expected the no clamping approach converged just as well as the velocity clamping approaches and no divergent behaviour was detected. The difference between the approaches are negligible. Similar results were obtained from the remaining benchmark functions.

Clamping approach	Global best	Diversity	Velocity
no clamping	4.77286e-20	3.60335e-11	2.62928e-11
basic clamping $k = 0.3$	9.32261e-22	2.71245e-12	1.94479e-12
basic clamping $k = 0.9$	3.08181e-21	1.06346e-11	8.07745e-12
normalised clamping $k = 0.3$	4.4788e-21	8.05522e-12	6.05397e-12
normalised clamping $k = 0.9$	5.26715e-22	6.98041e-12	5.40239e-12
dynamic clamping	1.28381e-20	2.23932e-11	1.56261e-11
exponential clamping	0.4167	0.00334	0.0

TABLE II
AVERAGE RESULTS FOR $f_5, \omega = 0.9, c_1 = c_2 = 0.7$

The velocity magnitude resulted in the same graph as figure 5. We can deduce from this that a PSO will display convergent behaviour when the diversity and velocity magnitudes of the swarm decrease over time. Figure 5 also depicts how particles start off by *exploring* as they are var from each other, but then move to a state in which *exploitation* is preferred as they move closer together. This is the desired behaviour of a particle swarm optimisation.

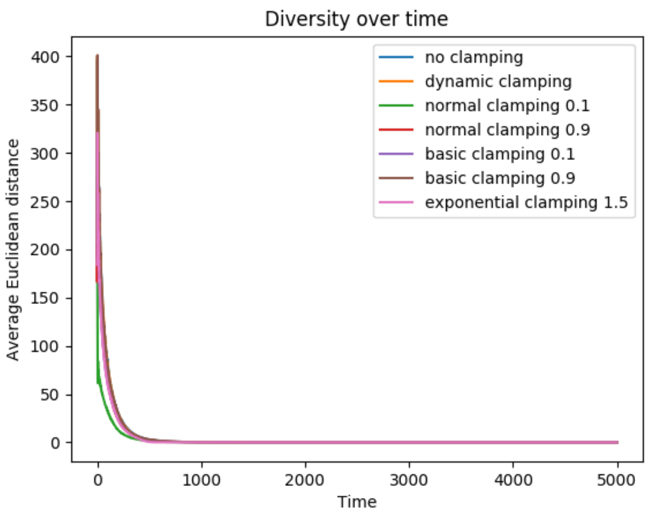

Fig. 5.

The percentage of particles that leave the search space did not return any significant insight. As expected, divergent behaviour resulted in higher percentages of particles moving into infeasible space. This is due to the high velocities of the particles. Conversely, convergent behaviour resulted in significantly lower percentages. There was no significant difference between approaches except in cases where one approach resulted in diversion and another resulted in conversion.

It is noteworthy that for a no-clamping approach there is consistent divergent or convergent behaviour depending on the selection of the control parameters. This is to be expected and is consistent with the findings of Poli [1]. The conclusion can be made that the PSO will always converge if the control parameters are chosen such that the convergence condition is satisfied and that there is no significant difference between the convergence of no-clamping and velocity clamping approaches.

VI. Conclusion

This section will convey the conclusions that were made by analysing the data that was gathered during the empirical procedure. Future research opportunities will also be discussed.

Is velocity clamping still necessary for 30-dimensional spaces? This is the question that this report attempts to answer. A global best particle swarm optimisation was implemented with six unique control parameter combinations. Three of these combinations conformed to the convergence condition, set by equation 8, and the other three did not. Each combination was tested against five different benchmark functions and five approaches to velocity clamping, including no clamping. As a measurement the quality of the global best position, diversity of the particle swarm, average velocity magnitude and the percentage of particles that leave the search space were considered. To obtain these measurements 30 simulations of 5000 iterations each were run for all unique combinations and the average was calculated.

According to the results, as seen in section V, there is an insignificant difference in convergence between no velocity clamping and various velocity clamping approaches given that the control parameters satisfy the convergence condition. It should be noted that only four velocity clamping approaches were considered, but the outcome of other approaches are expected to display the same behaviour. Some velocity clamping techniques like normalised clamping with a k value of 0.1 produced better quality global best positions than that of no-clamping, but there was no dissimilarity in the convergent behaviour.

Future research

- *Variable dimensionality*. This report only considers a 30-dimensional approach to the particle swarm. There may be additional insights to gain from considering larger and smaller dimensions. This would allow for a more general conclusion to be drawn.

- *Increased iterations*. The number of iterations considered in this report is limited to 5000. The behaviours of the different approaches analysed may change when the number of iterations per simulation is increased. This may cause additional conclusions to be drawn from the possible change in behaviour.

- *Variable swarm size*. This report only considers a swarm size of 30 particles. Increasing or decreasing this parameter may influence the convergent behaviour of the particle swarm. It is worth investigating whether this affects the results obtained in this report.

- *Benchmark functions*. Only five benchmark functions were examined. There are however a plethora of these benchmark functions and additional insights could be gained by examining these under the same conditions.

REFERENCES

[1] Poli, Riccardo. (2009). Mean and Variance of the Sampling Distribution of Particle Swarm Optimizers During Stagnation. Evolutionary Computation, IEEE Transactions on. 13. 712 - 721. 10.1109/TEVC.2008.2011744.

[2] Kennedy, J. & Eberhart, R.C., Particle swarm optimization. Proceedings of the IEEE International Conference on Neural Networks, Piscataway, NJ, pp. 1942–1948, 1995.

[3] Andries P. Engelbrecht. *Computational Intelligence: An Introduction*, Second Edition, John Wiley & Sons, University of Pretoria, South Africa, 2007. pp. 290-293, 303-306

[4] Competitive co-evolution of trend reversal indicators using particle swarm optimisation, E. Papacostantis. p. 47 `http://hdl.handle.net/2263/23929`

[5] E. T. Oldewage, A. P. Engelbrecht and C. W. Cleghorn, "The merits of velocity clamping particle swarm optimisation in high dimensional spaces," 2017 IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, 2017, pp. 1-8.