A Performance Study on Synchronous and Asynchronous Updates in Particle Swarm Optimization

Juan Rada-Vilela Mengjie Zhang Winston Seah

Victoria University of Wellington
School of Engineering and Computer Science
Wellington, New Zealand
{juan.rada-vilela, mengjie.zhang, winston.seah}@ecs.vuw.ac.nz

ABSTRACT

This work provides a further study on the difference between synchronous and asynchronous updates in Particle Swarm Optimization with different neighborhood sizes ranging from local best to global best. Ten well-known functions are used as benchmarks on both variants. Statistical tests performed on the results provide strong evidence to claim that synchronous updates yield in general better results with similar or even faster speed of convergence than its asynchronous counterpart, contrary to observations and conclusions of previous studies based solely on descriptive statistics.

Categories and Subject Descriptors

I.2 [**Computing Methodologies**]: Artificial Intelligence

General Terms

Performance

Keywords

Particle swarm optimization, synchronous and asynchronous updates, speed of convergence

1. INTRODUCTION

Ever since the first appearance of Particle Swarm Optimization (PSO) [3, 6], several variants and modifications have been proposed in order to improve quality of results, speed of convergence, or even just to adapt it to a given problem (see [4] for detailed examples).

Particles in the original PSO algorithm perform synchronous updates, that is, the best particle in each neighborhood is located and then used by the other particles to update their positions. Conversely, there also exists a variant using asynchronous updates, first mentioned in [2]. Each particle in this variant updates its position knowing the current best position found by half of its neighborhood and the previous best one found by the other half of it. These algorithms are referred to as Synchronous PSO (SPSO) and Asynchronous PSO (APSO), and can be regarded as situations of perfect and imperfect information, respectively.

In terms of performance, previous works have suggested that APSO generally yields better results than SPSO [2, 8, 9, 11, 12]. However, most of these works have based their observations and conclusions on the best result obtained after several independent runs. Furthermore, if we perform a Wilcoxon test on the results presented by Carlisle and Dozier [2], we realize that the differences are *not* statistically significant. This scenario leads to question whether APSO actually yields better results than SPSO, especially since the article of Carlisle and Dozier [2] is the reference by default when referring to comparisons between SPSO and APSO (cited in [8, 9, 11, 12, 15], and according to Google Scholar at the time of writing, cited by 300 articles from which 60 include the word *asynchronous*).

At the moment of writing this article, we have no knowledge of any previous works providing strong evidence that APSO actually outperforms SPSO. Consequently and facing this scenario, we decided to perform further comparisons between both variants with different social structures in ten well-known benchmark functions. Most importantly, we support our observations using boxplots and our conclusions using statistical tests on the quality of results and on the speed of convergence in all independent runs.

The overall goal of this paper is to investigate whether the claim on SPSO and APSO can still be supported by statistical significance tests. Specifically, we will focus on the following objectives:

1. Compare the performance of SPSO and APSO with different social network structures in terms of quality of results and speed of convergence on ten well-known benchmark functions.

2. Perform a statistical significance test to measure the importance of the differences in quality and in speed of convergence between SPSO and APSO.

3. Assess the effect of different social network structures on SPSO and APSO in terms of quality of results and speed of convergence.

This article is structured as follows. In Section 2 we present the theoretical background of SPSO and APSO, as well as the most relevant works to our research. In Section 3

GECCO'11, July 12–16, 2011, Dublin, Ireland.

we present the experimental design to achieve the goals proposed and introduce the indicator to measure the speed of convergence. In Section 4 we present and discuss the results obtained. In Section 5 we present our conclusions and suggestions for further research.

2. PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization was invented by Eberhart and Kennedy in 1995 [3, 6] with inspiration on social models (e.g. bird flocking, fish schooling) and swarming theory. It is a population-based algorithm in which its individuals (known as particles) encode potential solutions to n-dimensional optimization problems and explore the search space through cooperation with other particles by communicating the best solutions found so far and moving stochastically towards them.

More concretely, particles have a position vector $\mathbf{x}(t)$ that encodes a potential solution to the problem, and a velocity vector $\mathbf{v}(t)$ that determines the change in the position according to

$$\mathbf{x}(t+1) = \mathbf{x}(t) + \mathbf{v}(t+1) \tag{1}$$

It can be seen that the velocity vector balances the tradeoff between exploration and intensification of the search process: high velocities imply large changes in the positions of the particle and hence favoring exploration, whereas low velocities imply small changes and hence providing intensification. The velocity vector is computed for each dimension i as follows

$$\begin{aligned} v_i(t+1) = wv_i(t) &+ c_1r_1(t)[y_i(t) - x_i(t)] \\ &+ c_2r_2(t)[\hat{y}_i(t) - x_i(t)] \end{aligned} \tag{2}$$

where w is the inertia of the particle [13], c_1 and c_2 are positive acceleration coefficients (a.k.a. cognitive and social components) that weigh the importance of the personal and neighborhood knowledge, $r_1(t)$ and $r_2(t)$ are random values sampled from a uniform distribution, $y_i(t)$ is the best position found in dimension i (from $t = [0, t]$) by the particle itself, and $\hat{y}_i(t)$ is the best one found (from $t = [0, t]$) by its neighborhood.

2.1 Social network structure

The social network structure of the swarm defines the neighborhood of each particle and hence how they interact with each other. Several social structures have been proposed in the literature (see [4] for a survey review), but we are only interested in the **ring** and **star** social network structures.

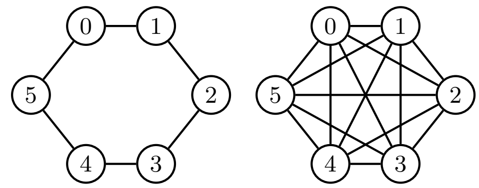
The **ring** social structure defines the neighborhood of each particle p_i according to Equation 3,

$$\mathcal{N}_i = \bigcup_{j=i-m}^{i+m} p_j \text{ with } m = \left\lfloor \frac{n}{2} \right\rfloor \bmod |\mathcal{S}| \tag{3}$$

where *mod* refers to the modulo operator using the Euclidean definition (for a convenient handling of negative j-values), n is the number of neighbors, and $\mathcal{S}$ refers to the swarm. Notice that each particle belongs to its own neighborhood and to those of the $\lfloor \frac{n}{2} \rfloor$ adjacent particles (neighborhoods overlap). Also notice that we consider neighborhoods to be symmetrical, that is, each particle has the same number of neighbors on both sides.

A particular case of the **ring** social structure is when the neighborhood of each particle is the whole swarm ($n = |\mathcal{S}|$), and then the social structure is known as **star**. The **star** social structure makes the whole swarm to be fully connected and hence all particles stochastically follow one particle on each iteration. PSO algorithms using the **star** social structure are often referred to as the *gbest* or Global Best PSO, and when the **ring** social structure is used instead with $n = 2$, the PSO algorithms are then referred to as *lbest* or Local Best PSO [4].

The **ring** and **star** social structures are represented in Figure 1, where particles (represented as circles) transmit their position and fitness to their neighbors and receive their respective information.



(a) $n = 2$ (b) $n = |\mathcal{S}|$

Figure 1: Ring and Star social structures.

The size of the neighborhoods affects the speed of convergence of the swarm. In large neighborhoods, the best solution found in it is used by more particles to update their positions and hence the change is greater than that in small neighborhoods. Conversely, in small neighborhoods the best solution does not propagate as fast, causing the magnitude of change in position of the particles to be more subtle.

The size of the neighborhoods is inversely correlated to the diversity of the swarm. Since the best solution is more widely propagated in large neighborhoods, more particles are going to be attracted to such a solution and hence the diversity of the swarm decreases rapidly across iterations. Contrarily, in small neighborhoods particles are attracted towards different solutions (i.e. the best within each neighborhood), and hence the swarm converges more slowly but providing a greater diversity.

In summary, the implications of using small or large neighborhoods come down to: a) small neighborhoods make the swarm to be more resilient to stagnation in local optima, but at the cost of a slow convergence, and b) large neighborhoods make the swarm to converge faster, but also more prone to stagnation in local optima. These implications make the choice of neighborhood size to be problem-dependent: in complex problems (i.e. multimodal functions) the swarm may benefit from small neighborhood sizes, whereas in simpler problems (i.e. unimodal functions) a large neighborhood size might be preferred. Experiments increasing and decreasing the neighborhood size have also been performed in the past [14, 10].

2.2 Synchronous updates

In the original PSO algorithm, all particles have perfect information about the neighborhood: the fitness of all particles is computed and shared within their respective neighborhoods. Only then, particles update their velocities considering the current best position found so far by their neighbors. We refer to this algorithm as the Synchronous PSO (SPSO), and it is summarized in Algorithm 1.

```
while not stopping condition do
    foreach Particle p in Swarm do
        p.evaluate();
    end
    foreach Particle p in Swarm do
        p.update();
    end
end
```

Algorithm 1: Synchronous PSO (SPSO).

It is important to remark that the action `evaluate` of each particle computes the fitness of its current position and, if it is better than the fitness of its best personal position, then the particle updates it and sends a message to the neighborhood communicating the discovery of such position and its respective fitness. Otherwise, no message is sent since either way all particles in the neighborhood will still use the best position previously found in their respective neighborhood. Regarding the action `update`, it performs the velocity and the position updates of the particle according to Equations 2 and 1, respectively.

2.3 Asynchronous updates

In this variant of the PSO algorithm, particles update their velocity immediately after computing their fitness and, as a consequence, they update it having imperfect information about the neighborhood. We refer to this algorithm as the Asynchronous PSO (APSO), and it is summarized in Algorithm 2.

```
while not stopping condition do
    foreach Particle p in Swarm do
        p.evaluate();
        p.update();
    end
end
```

Algorithm 2: Asynchronous PSO (APSO).

In this algorithm, a particle performs the action `evaluate` and `update`s immediately after. At this point, the particle uses the current best position found so far by the neighbors previously evaluated but not knowing that of the neighbors next to evaluate, instead, all they know about them is the best position found until the previous iteration (hence the term *asynchronous*). In other words, each particle knows the best solution found by half of its neighbors in the current iteration and the best solution found by the other half in the previous iteration.

2.4 Related work

Carlisle and Dozier [2] observed that APSO generally finds the best solution to the benchmark functions considered (two unimodal and three multimodal) faster than SPSO regardless of the neighborhood size, and they concluded that APSO is generally less costly. Their observations and conclusions are based on the median number of iterations in 20 independent runs with an upper-limit of 100,000 iterations each.[1]

Shortly after, Schutte [12] performed further comparisons between SPSO and APSO with different variations: constant inertia (CI), linear inertia with and without velocity clamping (LIV, LI), constriction factor (C), and dynamic inertia plus velocity clamping (DIV). His observations and conclusions, which are based on the results obtained from the best run over 50 independent runs, found that synchronous updates are less costly and more reliable in 12 benchmark functions (unimodal and multimodal). It is therefore unclear why he claims that the results support the findings in Carlisle and Dozier [2] since both sets of results are contradictory in this aspect.[2]

Luo and Zhang [9] compared SPSO and APSO on the `Rosenbrock` (unimodal) and `Griewank` (multimodal) benchmark functions. They observed and concluded from the best result found (and its respective evolutionary curve) in 25 independent runs of 8000 iterations each that APSO yields better results and with a faster convergence than SPSO. No statistical test was performed on the results.

Perez and Basterrechea [11] compared SPSO and APSO on five benchmark functions (`Griewank`, `Rosenbrock`, `Sphere`, `Rastrigin` and `Schaffer's f6`) and on an electromagnetic problem entitled *Antenna far-field pattern reconstruction*. They performed 20 independent runs with an upper-limit of 10000 iterations each on the benchmark functions, and different configurations on the electromagnetic problem. They observed from their results on the benchmark functions that APSO is able to find solutions faster and with a similar accuracy as SPSO. They based such observations on the mean number of iterations for each algorithm to find the optimal solution and on the success rate. They finally conclude that APSO offers the best trade-off between accuracy and computational time.

These are the most relevant works to this research. All of them have compared favorably APSO to SPSO using descriptive statistics, but none of them has provided strong evidence to support their observations and conclusions.

3. EXPERIMENTAL DESIGN

3.1 Benchmark functions

Ten well-known benchmark functions [4] (unimodal and multimodal) are used for testing both variants in different scenarios. All functions are minimization problems and are detailed below:

1. The `Quadric` function (unimodal)
   $f_1(\mathbf{x}) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$ where $-100 \leq x_j \leq 100$.

[1] We performed a Wilcoxon test using paired samples on their results and we could not find strong evidence at a significance level $\alpha = 0.95$ to support their conclusions and observations (i.e. the differences are not statistically significant).

[2] We performed a Wilcoxon test with paired samples on their results at a confidence level $\alpha = 0.95$, and we found no significant differences between APSO and SPSO in terms of cost and reliability with most variations: CI, CIV, C (in cost), and CIV, LI, LIV, C (in reliability). We only found strong evidence of APSO being more reliable with CI and DIV, and also more costly with LI, LIV, and DIV.

2. The `Quartic` function (unimodal)
$f_2(\mathbf{x}) = \sum_{i=1}^{n} i x_i^4$ where $-1.28 \leq x_i \leq 1.28$.

3. The `Rosenbrock` function (unimodal)
$f_3(\mathbf{x}) = \sum_{i=1}^{n-1} 100 \left(x_i^2 - x_{i+1}\right)^2 + (x_i - 1)^2$
where $-2.048 \leq x_i \leq 2.048$.

4. The `Spherical` function (unimodal)
$f_4(\mathbf{x}) = \sum_{i=1}^{n} x_i^2$ where $-5.12 \leq x_i \leq 5.12$.

5. The `HyperEllipsoid` function (unimodal)
$f_5(\mathbf{x}) = \sum_{i=1}^{n} i x_i^2$ where $-5.12 \leq x_i \leq 5.12$.

6. The `Ackley` function (multimodal)
$$f_6(\mathbf{x}) = 20 + e - 20 \exp\left[-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right] - \exp\left[\frac{1}{n}\sum_{i=1}^{n} \cos\left(2\pi x_i\right)\right]$$
where $-32.768 \leq x_i \leq 32.768$.

7. The `Griewank` function (multimodal)
$f_7(\mathbf{x}) = 1 + \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right)$
where $-600 \leq x_i \leq 600$.

8. The `Rastrigin` function (multimodal)
$f_8(\mathbf{x}) = 10n + \sum_{i=1}^{n} x_i^2 - 10\cos\left(2\pi x_i\right)$
where $-5.12 \leq x_i \leq 5.12$.

9. The `Salomon` function (multimodal)
$f_9(\mathbf{x}) = 1 - \cos\left(2\pi\sqrt{\sum_{i=1}^{n} x_i^2}\right) + 0.1\sqrt{\sum_{i=1}^{n} x_i^2}$
where $-600 \leq x_i \leq 600$.

10. The `EggHolder` function (multimodal)
$$f_{10}(\mathbf{x}) = \sum_{i=1}^{n-1} -(x_{i+1} + 47)\sin\left(\sqrt{\left|x_{i+1} + \frac{x_i}{2} + 47\right|}\right) - x_i \sin\left(\sqrt{\left|x_i - (x_{i+1} + 47)\right|}\right)$$
where $-512 \leq x_i \leq 512$.

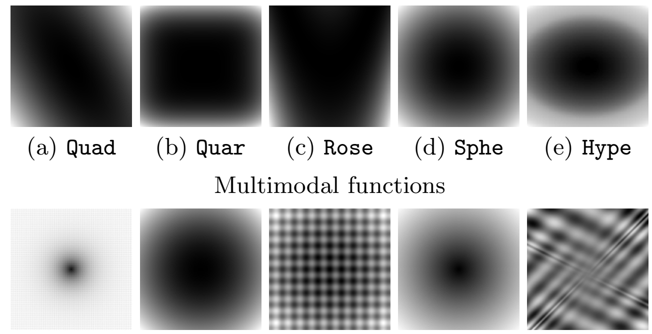
Figure 2 shows the complexity of these functions in two dimensions. These functions are plotted using the range previously defined in each for coordinates x and y of the image, and the result of $f(x, y)$ is represented in grayscale where darker tones mean lower values and lighter tones mean higher values. The coordinate $(0, 0)$ is located in the middle of the image, and it represents the global minima in all functions except for `EggHolder` which global minimum is located approximately at $f(512, 404) = -959.57$ (near the upper-right corner). Notice that `Ackley`, `Griewank` and `Salomon` look unimodal in the figure due to the resolution of the image, but in proper resolution the multimodality caused by the cosine functions can be noticed.

3.2 Experimental setup

In both PSO variants we consider the ring social structure with different neighborhoods ranging from $n = 2$ to $n = 30$. The swarms in all configurations have 30 particles with 30 dimensions each, and the initial position of the particles is randomly chosen from a uniform distribution. The velocity of the particles is constrained using the hyperbolic tangent function [4]. The values regarding acceleration coefficients and inertia are chosen according to the guidelines in [16]. Table 1 summarizes the parameters used in all independent runs.

We designed the experimental study based on two performance metrics: quality of results and speed of convergence.

Unimodal functions



(a) `Quad` (b) `Quar` (c) `Rose` (d) `Sphe` (e) `Hype`

Multimodal functions

(f) `Ackl` (g) `Grie` (h) `Rast` (i) `Salo` (j) `EggH`

Figure 2: Benchmark functions in two dimensions.

Table 1: Algorithm parameters.

Parameter	Value
Iterations	300
Independent runs	50
Number of particles	30
Number of dimensions	30
Social structure	Ring with $n = \{2, 6, 14, 22, 30\}$
Acceleration	Static with $c_1 = c_2 = 1.49618$
Inertia	Static with $w = 0.729844$
Velocity clamping	hyperbolic tangent
Maximum velocity	$0.25 \cdot \lvert x_{\max} - x_{\min}\rvert$

3.2.1 *Quality of results*

In order to assess the quality of results each variant is capable of delivering, we record the best result (i.e. best fitness) obtained in each independent run on each benchmark function. Then, we use boxplots to display the distribution of the recorded results and observe which configuration is more likely to yield better results. Finally, we perform a statistical test on the results between the variants to determine the significance of the differences.

3.2.2 *Speed of convergence*

The speed of convergence between two evolutionary curves can be easily assessed when one curve strictly dominates the other, however, this is not always the case since these curves can intersect each other in different iterations and even converge to different values, thus, there can be many criteria to measure the speed of convergence.

In this paper we introduce the Area Under the Curve (AUC) as an indicator of speed of convergence. This indicator considers the descent of the evolutionary curve and the final result as equally important factors. Thus, in minimization problems, the smaller the AUC the faster the convergence. Notice that the AUC indicator is equivalent to the hypervolume indicator [1] in two dimensions and, as such, it is important to choose an adequate point of reference in order to avoid unexpected values as was shown in [7] when arbitrary points were chosen.

Now, to measure the speed of convergence in each variant, we record the evolutionary curve depicted by the best particle over the iterations during each run. Thus, we have 50 evolutionary curves for each algorithm on each benchmark function. Then, we proceed to compute the AUC on all evolutionary curves using $(0, 0)$ as the point of reference and, in order to increase the accuracy of the indicator, for each benchmark function we translate all the evolutionary curves by subtracting the minimum fitness found with any algorithm. This way we minimize the AUC of all evolutionary curves but we preserve the relation between them. Afterwards, we display in boxplots the distribution of the indicator to observe which configuration is more likely to deliver a faster speed of convergence. Finally, we perform a statistical test on the results to determine the significance of the differences.

3.2.3 Statistical test

The statistical test we use to measure the significance of the differences between both variants on the quality of results and on the speed of convergence is the Wilcoxon test. The reasons for such choice are that 1) it does not assume the normality of the samples, and 2) it has already demonstrated to be helpful analyzing the behavior of evolutionary algorithms [5].

On the quality of results, we perform the Wilcoxon test between variants in each configuration. That is, we use the 50 results of one SPSO configuration and the 50 results of the APSO respective configuration. A similar approach is used regarding the speed of convergence.

Finally, the significance level we use is $\alpha = 0.95$.

4. RESULTS AND DISCUSSION

4.1 Quality of results

4.1.1 Observations

The quality of results of both variants is shown as boxplots in Figure 3. Each boxplot represents the distribution of the 50 results (best result in each independent run) obtained by each algorithm. Notice that by best result we mean the fitness of the best particle, i.e. the minimum function value obtained by the swarm.

Notice that the variants are denoted by letters from a to e according to the neighborhood size ($n_a = 2$, $n_b = 6$, $n_c = 14$, $n_d = 22$, $n_e = 30$) and when the letter is starred (*) it refers to the APSO variant. These boxplots do not include the outliers to allow a better visualization, and also the results from boxplots with gray background were scaled down to better visualize the other boxplots. The following scale factors were applied to both variants: $\dot{a} = \frac{1}{500}$ and $\dot{b} = \frac{1}{20}$ in `Quartic`, $\dot{a} = \frac{1}{20}$ and $\dot{b} = \frac{1}{5}$ in `Spherical`, and $\dot{a} = \frac{1}{10}$ in `HyperEllipsoid`.

Figure 3 shows that the SPSO yields better results than the APSO in most of the functions, being the differences more marked in unimodal than in multimodal functions.

In unimodal functions we observe that results improve as larger neighborhoods are considered. This behavior is expected because unimodal functions have no local minima but just one global minimum. Hence, once a particle compares to any other particle with better or worse fitness, the sign of the difference between their positions will determine the direction towards the global minimum, and from then on the particle will start to continuously increase the velocity to achieve it. Now, what makes larger neighborhoods yield better results in these functions is that particles have a better chance of finding a better neighbor in larger neighborhoods, and hence their velocity will be higher. This behavior is problem-dependent in multimodal problems given the risk of stagnation in local minima.

In multimodal problems, particularly in `Ackley`, `Rastrigin` and `EggHolder`, we observe that results do not improve much by increasing the neighborhood size, in fact, results are sometimes worse. This behavior is expected given the complexity of such functions (see Figure 2) and especially considering that larger neighborhoods are more prone to stagnate in local minima. The other functions do not present such complexity and the behavior of both variants is similar as that in unimodal problems.

4.1.2 Significance of the results

Tables 2 and 3 show the significance of the difference between the best results obtained with both variants in the unimodal and multimodal functions. The symbol $=$ indicates that the results obtained are not significantly different, whereas symbols $+$ and $-$ indicate that results are significantly greater or less, respectively. Notice that since the functions are to be minimized, the symbol $-$ indicates better results.

Table 2 shows that SPSO achieved significantly better results than APSO in most unimodal functions: in `Quartic` and `Spherical` regardless of the neighborhood size, in `HyperEllipsoid` for all neighborhood sizes except for $n = 30$, and in `Rosenbrock` except for $n = \{22, 30\}$. The differences were not statistically significant in `Quadric` for all neighborhood sizes.

In Table 3 we are particularly interested in the significance of the differences in multimodal problems where the boxplots in Figure 3 suggest that the results from APSO are better than the those from SPSO: when $n = \{22, 30\}$ in `Ackley`, when $n = \{14, 22, 30\}$ in `Rastrigin`, and when $n = \{14\}$ in `EggHolder`. In these cases, Table 3 shows that APSO is significantly better only when $n = 30$ in `Rastrigin` and when $n = 14$ in `EggHolder`, otherwise the differences are not statistically significant. However, notice that `Rastrigin` and `EggHolder` have a more complex search space than the other functions (see Figure 2).

Now, in general, the differences are either favorable to SPSO (`Griewank` regardless of the neighborhood size, `Salomon` when $n = \{6, 14, 22\}$, and `Ackley` when $n = 2$) or just not statistically significant.

4.1.3 Average results

We decided to include the average of the best results of each algorithm on each benchmark function to show how counterintuitive the results from this statistic might be, and how its sensitivity to outliers may lead to incorrect assumptions about the general behavior of the algorithm. Consider the following cases in Tables 4 and 5 where APSO is better than SPSO *but* their respective differences (in Tables 2 and 3) are *not* statistically significant

- In `Quadric`: $\{\bar{a}/\bar{a}^*, \bar{b}/\bar{b}^*, \bar{e}/\bar{e}^*\}$
- In `HyperEllipsoid`: $\{\bar{e}/\bar{e}^*\}$
- In `Ackley`: $\{\bar{c}/\bar{c}^*, \bar{d}/\bar{d}^*, \bar{e}/\bar{e}^*\}$
- In `Rastrigin`: $\{\bar{a}/\bar{a}^*, \bar{b}/\bar{b}^*, \bar{d}/\bar{d}^*\}$
- In `EggHolder`: $\{\bar{a}/\bar{a}^*, \bar{b}/\bar{b}^*, \bar{d}/\bar{d}^*, \bar{e}/\bar{e}^*\}$

Now, consider the case of $\bar{c}/\bar{c}^*$ in `EggHolder` (Table 3)

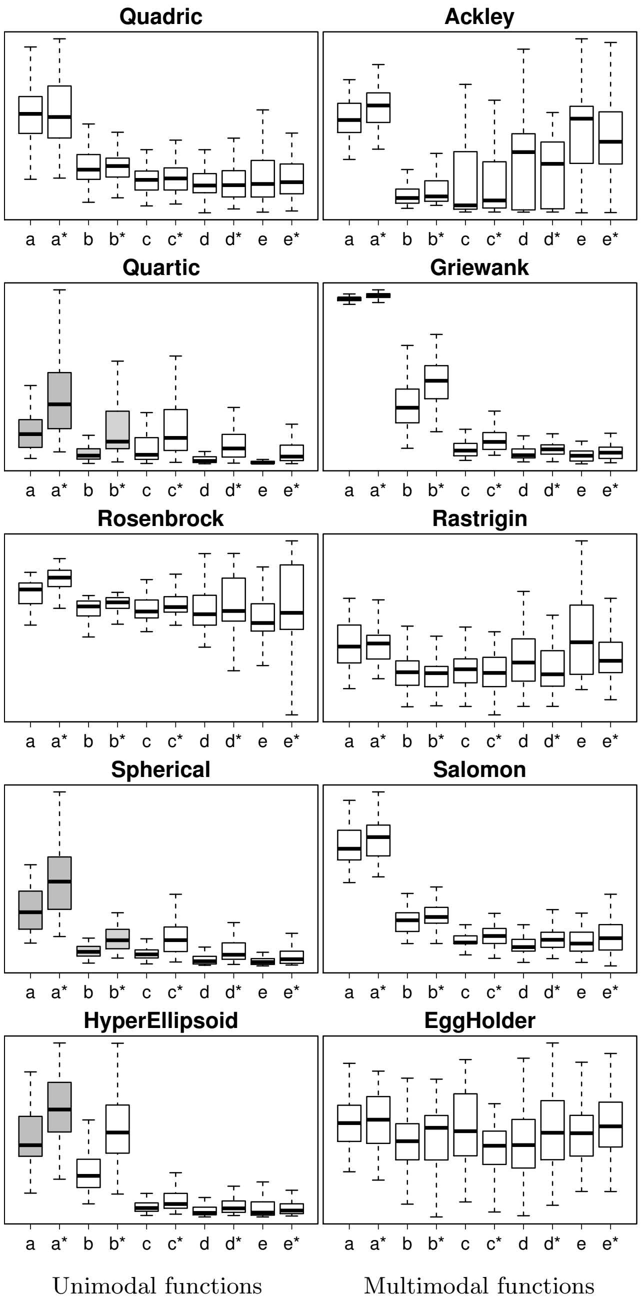


Figure 3: Quality of results in benchmark functions.

where APSO is significantly better than SPSO. If we examine the average result in Table 5 we see that the average result from APSO is worse than that of SPSO.

Other examples include the cases where the average result of SPSO is slightly better (e.g. `Rosenbrock` or `HyperEllipsoid`, and most multimodal functions) and the significance test shows that such difference in results is actually statistically significant in most cases favoring SPSO. All these comparisons show how disrupting the effect of outliers may be when using the mean as a statistic indicator on results from evolutionary algorithms such as PSO, leading to potentially false assumptions about the general behavior of the algorithm.

Table 2: Significance on unimodal functions.

	Quad	Quar	Rose	Sphe	Hype	n
a/a^*	=/=	−/+	−/+	−/+	−/+	2
b/b^*	=/=	−/+	−/+	−/+	−/+	6
c/c^*	=/=	−/+	−/+	−/+	−/+	14
d/d^*	=/=	−/+	=/=	−/+	−/+	22
e/e^*	=/=	−/+	=/=	−/+	=/=	30

Table 3: Significance on multimodal functions.

	Ackl	Grie	Rast	Salo	EggH	n
a/a^*	−/+	−/+	=/=	=/=	=/=	2
b/b^*	=/=	−/+	=/=	−/+	=/=	6
c/c^*	=/=	−/+	=/=	−/+	+/−	14
d/d^*	=/=	−/+	=/=	−/+	=/=	22
e/e^*	=/=	−/+	+/−	=/=	=/=	30

Table 4: Averaged best results on unimodal functions.

	Quad	Quar	Rose	Sphe	Hype
$\bar{a}/\bar{a}^*$	2.57/2.53	2.85/5.86	2.75/2.82	5.04/7.58	0.14/0.18
$\bar{b}/\bar{b}^*$	1.39/1.37	0.97/3.07	2.70/2.74	1.39/2.44	0.08/0.15
$\bar{c}/\bar{c}^*$	1.02/1.09	1.71/4.97	2.72/2.74	1.22/2.49	0.02/0.03
$\bar{d}/\bar{d}^*$	0.94/1.14	0.60/1.98	2.68/2.83	0.76/1.88	0.01/0.02
$\bar{e}/\bar{e}^*$	1.42/1.32	0.51/1.77	2.87/3.01	0.58/0.89	5.26/0.01
	$\times 10^3$	$\times 10^{-8}$	$\times 10^{-1}$	$\times 10^{-4}$	$\times 10^{-1}$

Table 5: Averaged best results on multimodal functions.

	Ackl	Grie	Rast	Salo	EggH
$\bar{a}/\bar{a}^*$	1.77/2.01	1.02/1.05	6.43/6.42	3.91/4.08	1.19/1.18
$\bar{b}/\bar{b}^*$	0.45/0.47	0.36/0.52	4.80/4.63	1.89/2.03	1.27/1.25
$\bar{c}/\bar{c}^*$	0.65/0.58	0.10/0.16	4.91/4.99	1.39/1.51	1.21/1.29
$\bar{d}/\bar{d}^*$	0.96/0.75	0.06/0.10	5.48/5.13	1.30/1.45	1.27/1.23
$\bar{e}/\bar{e}^*$	1.56/1.40	0.06/0.09	6.95/5.71	1.44/1.50	1.22/1.20
	$\times 1$	$\times 1$	$\times 10^{-1}$	$\times 1$	$\times -10^4$

4.1.4 Key findings

We have provided strong evidence to show that SPSO generally yields better results than APSO in unimodal functions. Regarding the multimodal functions, SPSO yields similar or even better results than APSO. In both cases, our key findings contradict the observations and conclusions from previous works [2, 8, 9, 11, 12, 15].

4.2 Speed of convergence

4.2.1 Observations

The results of the AUC indicator for all evolutionary curves are shown as boxplots in Figure 4. Similar to the quality of results, each boxplot represents the distribution of the AUC indicator on the evolutionary curves depicted by the best particle in all 50 independent runs.

Notice that the median and quartiles in the boxplots regarding SPSO tend to be lower than their respective APSO, showing a similar behavior as the boxplots on the best results in Figure 3 but with less marked differences. These boxplots suggests that SPSO has a faster speed of convergence according to the AUC indicator.

The differences in speed of convergence are more marked for neighborhoods with $n = \{2, 6\}$, but they become less clear as the neighborhood size increases. In fact, it seems that for $n = \{22, 30\}$ the speed of convergence is not as fast since the respective boxplots show the median slightly above the median of neighborhoods with $n = 14$. This might be due to the fact that particles in larger neighborhoods tend to achieve a higher velocity which allows them to have a fast descent, but it also causes a slower convergence due to their accumulated velocity.

4.2.2 *Significance of the results*

Results from the Wilcoxon test on the AUC indicators are shown in Tables 6 and 7. The parentheses in some of these results mean that the significance regarding the speed of convergence between both variants is different to the significance in quality of results between the same variants in Tables 2 and 3, respectively.

In unimodal functions we find that SPSO is significantly better in `Quartic` and `Rosenbrock` for neighborhood sizes $n = \{2, 6, 14\}$, in `Spherical` for $n = \{2, 6, 14, 22\}$, and in `HyperEllipsoid` for $n = \{2, 6\}$. The speed of convergence was not significantly different for the rest of neighborhood sizes, especially in `Quadric` where both variants had a similar speed of convergence for all neighborhood sizes.

In multimodal functions SPSO was significantly better in `Ackley` for $n = \{2, 6, 14\}$, in `Griewank` for $n = \{2, 6\}$, in `Salomon` for $n = \{2, 6, 14, 22\}$, and in `EggHolder` for $n = \{6, 22\}$. There were no significant differences regarding the rest of the neighborhood sizes and functions except for two cases where APSO was significantly better: in `Rastrigin` for $n = 30$ and in `EggHolder` for $n = 14$.

Notice that the significance of the difference between SPSO and APSO regarding the speed of convergence coincides, in most cases, with the significance of the quality of results (there are few results in parentheses). This is also true when APSO outperforms SPSO: in `Rastrigin` when $n = 30$, and in `EggHolder` when $n = 14$.

Table 6: Significance of AUC on unimodal functions.

	`Quad`	`Quar`	`Rose`	`Sphe`	`Hype`	n
a/a^*	=/=	−/+	−/+	−/+	−/+	2
b/b^*	=/=	−/+	−/+	−/+	−/+	6
c/c^*	=/=	−/+	−/+	−/+	(=/=)	14
d/d^*	=/=	(=/=)	=/=	−/+	(=/=)	22
e/e^*	=/=	(=/=)	=/=	(=/=)	=/=	30

Table 7: Significance of AUC on multimodal functions.

	`Ackl`	`Grie`	`Rast`	`Salo`	`EggH`	n
a/a^*	−/+	−/+	=/=	(−/+)	=/=	2
b/b^*	(−/+)	−/+	=/=	−/+	(−/+)	6
c/c^*	(−/+)	(=/=)	=/=	−/+	+/−	14
d/d^*	=/=	(=/=)	=/=	−/+	(−/+)	22
e/e^*	=/=	(=/=)	+/−	=/=	=/=	30

4.2.3 *Key findings*

From our results we conclude that SPSO generally has a similar or even faster speed of convergence than APSO, and not the other way around as observed in previous works [2, 9, 11, 15].

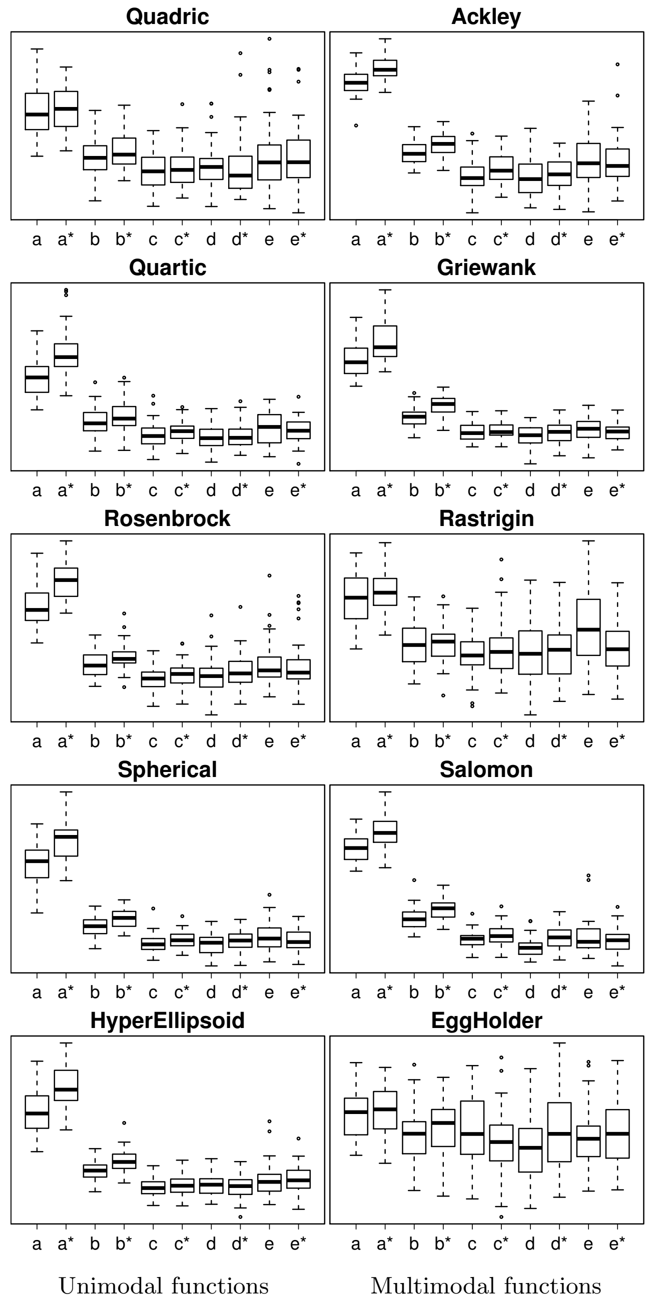


Figure 4: Speed of convergence.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have compared SPSO and APSO with different social network structures in ten well-known benchmark functions. The results we obtained suggest that SPSO is actually better than APSO, contrary to what previous studies have claimed. We performed a statistical test on these results to assess the significance of the differences, and obtained strong evidence to support our observations: SPSO generally yields better results than APSO. We also showed how using the mean as a statistic to analyze these results might lead to incorrect assumptions about the general behavior of the PSO algorithms.

We have also introduced the Area Under the Curve (AUC) as an indicator to measure the speed of convergence. This indicator equally weighs the importance of the descent in the evolutionary curve and the quality of the result to which it converges. Using the evolutionary curves depicted by the best results obtained, we computed their AUC and observed that the speed of convergence is faster in general for SPSO. We proceeded to perform a statistical test on these results to assess the significance of the difference in speed of convergence between SPSO and APSO. The results from this test provided strong evidence in several cases that SPSO has a faster speed of converge than APSO, but in several other cases the results were not conclusive and suggested that both variants had a similar speed of convergence. Therefore, from a broad point of view, we conclude that SPSO generally has a similar or even faster speed of convergence than APSO.

Regarding the social network structures considered, the results depicted in general the expected behavior when increasing the neighborhood size: faster speeds of convergence and better results in unimodal problems, and faster speeds of convergence but not necessarily better results in multimodal problems. Nevertheless, the most important result in this matter is the significant difference in quality of results and respective speed of convergence when using neighborhoods of $n = 2$ and $n = 6$. SPSO and APSO using the ring social structure with $n = 6$ yielded better results than when $n = 2$ in both unimodal and multimodal functions.

Finally, this work highlights the importance of analyzing results and drawing conclusions considering all the results obtained in all independent runs instead of just using the best one found, since it might just be circumstantial and not the product of the general behavior of the algorithm. Furthermore, it also demonstrates the importance of using statistical tests on the results to better support observations and conclusions about the general behavior and performance of the algorithms in question.

In the future, this work could be extended by

- Comparing both variants on other benchmark functions as well as on those already tackled but analyzed without robust statistics or statistical significance tests.
- Comparing both variants on functions with a similar or greater complexity as `Rastrigin` or `EggHolder` where APSO managed to significantly outperform SPSO with two configurations ($n = 30$ and $n = 14$, respectively) and achieve similar results with the rest.
- Analyzing the behavior of both algorithms when using different numbers of particles.
- Using different indicators to assess speed of convergence.
- Finding an *ideal* neighborhood size in which improvements on results are still significant with respect to a smaller neighborhood in both unimodal and multimodal functions and without risking stagnation in local minima.

6. REFERENCES

[1] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler. Theory of the hypervolume indicator: optimal μ-distributions and the choice of the reference point. In *Proceedings of the tenth ACM SIGEVO workshop on Foundations of genetic algorithms*, pages 87–102. ACM, 2009.

[2] A. Carlisle and G. Dozier. An off-the-shelf PSO. In *Workshop on Particle Swarm Optimization*, 2001.

[3] R. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *Sixth International Symposium on Micro Machine and Human Science*, pages 39–43, 1995.

[4] A. Engelbrecht. *Computational Intelligence: An introduction*. John Wiley & Sons Ltd, second edition, 2007.

[5] S. García, D. Molina, M. Lozano, and F. Herrera. A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special Session on Real Parameter Optimization. *Journal of Heuristics*, 15(6):617–644, 2008.

[6] J. Kennedy and R. Eberhart. Particle Swarm Optimization. In *IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948, 1995.

[7] J. Knowles and D. Corne. On metrics for comparing nondominated sets. In *IEEE Congress on Evolutionary Computation*, volume 1, pages 711–716, 2002.

[8] B.-I. Koh, A. George, R. Haftka, and B. Fregly. Parallel asynchronous particle swarm optimization. *International Journal for Numerical Methods in Engineering*, 67:578–595, January 2006.

[9] J. Luo and Z. Zhang. Research on the Parallel Simulation of Asynchronous Pattern of Particle Swarm Optimization. *Computer Simulation*, 22(6):78–70 (in chinese), 2006.

[10] M. Montes de Oca, T. Stutzle, M. Birattari, and M. Dorigo. Frankenstein's PSO: A Composite Particle Swarm Optimization Algorithm. *IEEE Transactions on Evolutionary Computation*, 13(5):1120–1132, 2009.

[11] J. R. Perez and J. Basterrechea. Particle swarm optimization and its application to antenna farfield-pattern prediction from planar scanning. *Microwave and optical technology letters*, 44(5):398–403, 2005.

[12] J. Schutte. Particle Swarms in Sizing and Global Optimization. Master's thesis, University of Pretoria, South Africa, 2001.

[13] Y. Shi and R. Eberhart. A modified particle swarm optimizer. In *IEEE World Congress on Computational Intelligence*, pages 69–73, 1998.

[14] P. Suganthan. Particle swarm optimiser with neighbourhood operator. In *IEEE Congress on Evolutionary Computation*, volume 3, pages 1958–1962, 1999.

[15] C. Sun, C. Chiu, and C. Li. Time-Domain inverse scattering of a two-dimensional metallic cylinder in slab medium using asynchronous particle swarm optimization. *Progress In Electromagnetics Research M*, 14:85–100, 2010.

[16] F. van den Bergh. *An analysis of particle swarm optimizers*. PhD thesis, University of Pretoria, South Africa, 2002.