Particle Swarm Optimization Algorithm with a Bio-Inspired Aging Model

Eduardo Rangel-Carrillo,
Esteban A. Hernandez-Vargas, Nancy Arana-Daniel,
Carlos Lopez-Franco and Alma Y. Alanis

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.71791

Abstract

A Particle Swarm Optimization with a Bio-inspired Aging Model (BAM-PSO) algorithm is proposed to alleviate the premature convergence problem of other PSO algorithms. Each particle within the swarm is subjected to aging based on the age-related changes observed in immune system cells. The proposed algorithm is tested with several popular and well-established benchmark functions and its performance is compared to other evolutionary algorithms in both low and high dimensional scenarios. Simulation results reveal that at the cost of computational time, the proposed algorithm has the potential to solve the premature convergence problem that affects PSO-based algorithms; showing good results for both low and high dimensional problems. This work suggests that aging mechanisms do have further implications in computational intelligence.

Keywords: particle swarm optimization, bio-inspired aging model, evolutionary optimization algorithms

1. Introduction

Bio-inspired optimization algorithms are based on precise observation of natural systems [1–3]. A relevant characteristic of these algorithms is that the biological process had been tested, validated and proven by means of evolution. The mechanisms of self-adaption, self-organizing and self-learning in natural inspired optimization approaches provide means to address challenging problems that cannot be solved by traditional methods [4].

Thus, bio-inspired algorithms become particularly important to tackle complex optimization problems [6–10]. The outstanding performance of bio-inspired optimization algorithms is attributed to their structures which are closely related to one or other features observed in nature [10]. Accuracy and repeatability are the prime objectives of every optimization algorithm. Therefore, modeling biological mechanisms may impact the outcome of the observed system, designing more accurate and efficient heuristic algorithms [13, 16].

Problem-solving algorithms inspired by one or more biological features have been developed after observing the behavior in humans, animals, and cells [10–13]. For instance, Genetic Algorithms (GA) [4] defined the basis for evolutionary computing using the early works of Darwin [14] and Mendel [15]; or the Ant Colony System (ACS) [6, 7] which considers the traveling behavior model of self-organized argentine ants published by Goss [5], solving in a fashion way travel salesman-type problems; or the Particle Swarm Optimization (PSO) [8] inspired on feeding behavior of bird flocks becoming a very popular optimizer nowadays [17].

Particle-based optimizers, like those described in [8]; or those presented in [18–22, 28] are very popular because instead of working with one candidate solution, they offer a subset of individual candidate solutions (particles), which are explored, exploited and improved. A relevant mechanism related to evolution that could play a central role in optimization algorithms is aging [23, 25]. Aging is a natural characteristic whose inclusion in a particle-based optimizer could give a mean of individual control over the particle without highly increasing the complexity of algorithms [26].

To the authors' best knowledge, previous PSO algorithms did not have a measurement to control individual particle existence within the swarm by evaluating each particle performance. In PSO, because of the very nature of the algorithm, an effect called *premature convergence* appears when most (or all) of the particles within a swarm compromises their ability to explore and stay close to a local solution. The Particle Swarm Optimization with an Aging Leader and Challengers (ALC-PSO) [24] was the first approach to include aging processes to alleviate this unwanted effect. However, this was only leadership-oriented and not swarm-related; even more important: the aging dynamics were linear and bounded to static predefined values. In [29], it is used to design a high speed symmetric switching CMOS inverter.

Our PSO variant proposal, the Particle Swarm Optimization with Bio-inspired Aging Model (BAM-PSO) is based on a mathematical model that describes the telomeres shortening observed in the immune system cells, this model includes a form of *aging* effect over all the particles of the swarm; this mechanism provides a mean to control the existence of each particle within the swarm avoiding the premature convergence effect. Therefore, the PSO variant with aging model possesses the potential to outperform current optimization algorithms and have further implications in computational intelligence. Finally, optimization under uncertainty and complex functions is an area of special scientific interest, and many real problems and applications include some form of uncertainty; it is also known that collective-intelligence algorithms perform excellent in this type of scenarios [11]; BAMPSO has been successfully implemented in several optimization applications: in time-series forecasting [30] it was implemented as a training algorithm for an artificial neural network and in [31] it was used over a Geometric Algebra (GA) framework in order to compute the rigid movement on images to improve the accuracy of Structure from Motion (SfM) algorithms, which comprises a family of computer vision algorithms whose paradigm is based on extracting structures when movement is detected or extracting movement when structures are detected in 2D images. Therefore, in this work, BAM-PSO is tested with several popular and well-established benchmark functions and its performance is compared to well-known evolutionary algorithms in both low and high dimensional scenarios.

2. Aging mechanisms to alleviate the premature convergence

Aging is the process of becoming older, which consists on the accumulation of changes over time. This process affects all living systems: humans, cells, unicellular organisms, fruit flies and mammals like rodents [12, 32, 42]. Since the particles of the PSO optimizer algorithm can be treated as a living system, aging could represent a relevant mechanism to alleviate the premature convergence problem in heuristic algorithms. Nowadays, we have better understanding of the lifespan of human cells, which is determined by homoeostatic properties of the immune system. Homeostasis refers to the regulation of the lymphocytes pool in an organism. It is assumed that the number of cells is determined by the capacity of the peripheral immune system.

In the immune system, it is observed that cell death rate accelerates if the immune cells exceed the allocated free space [33]. For instance, in the course of a viral infection, immune system cells can undergo approximately 15–20 divisions. Total proliferative capacity of human T lymphocyte is about 40–45 divisions and depends on the telomere length [41]. Telomeres are the end parts of the chromosomes, which become shorter in every cell division; this can be appreciated in **Figure 1**. The cell can reach its unresponsiveness state when the telomere length completes about one half of its initial value.

Telomere dynamics can be interpreted in a mathematical model based on experimental observations. In this work, we consider the mathematical model proposed by [33] to represent the telomere dynamics. This model considers the following equation:

$$\frac{dT}{dt} = \alpha(p^* - T)N \tag{1}$$

where T represents the remaining telomere divisions per cell. α defines the telomere consumption rate per iteration. p^* represents the length of telomere repeats in naive cells produced at the age t (with initial length p_0= 8.3 × 10^3) and N is the number of cells as defined in [33].

Eq. (1) is a differential non-ascending equation that defines the derivative in telomere division per cell depending on the consumption rate of the cell α, telomere capacity of the cell (p^*) and number of cells (N). Eq. (1) describes the dynamic of average telomere length T in the pool of naive cells. The rate of this process depends on $(p^* - T)$, where p^* is the telomere length in the cells and p_0 defines the telomere at initial age. This dynamic describes the self-sustaining process of regulation of total concentration of the T cells and how the telomere is affected by T cells concentration and iterations [33].

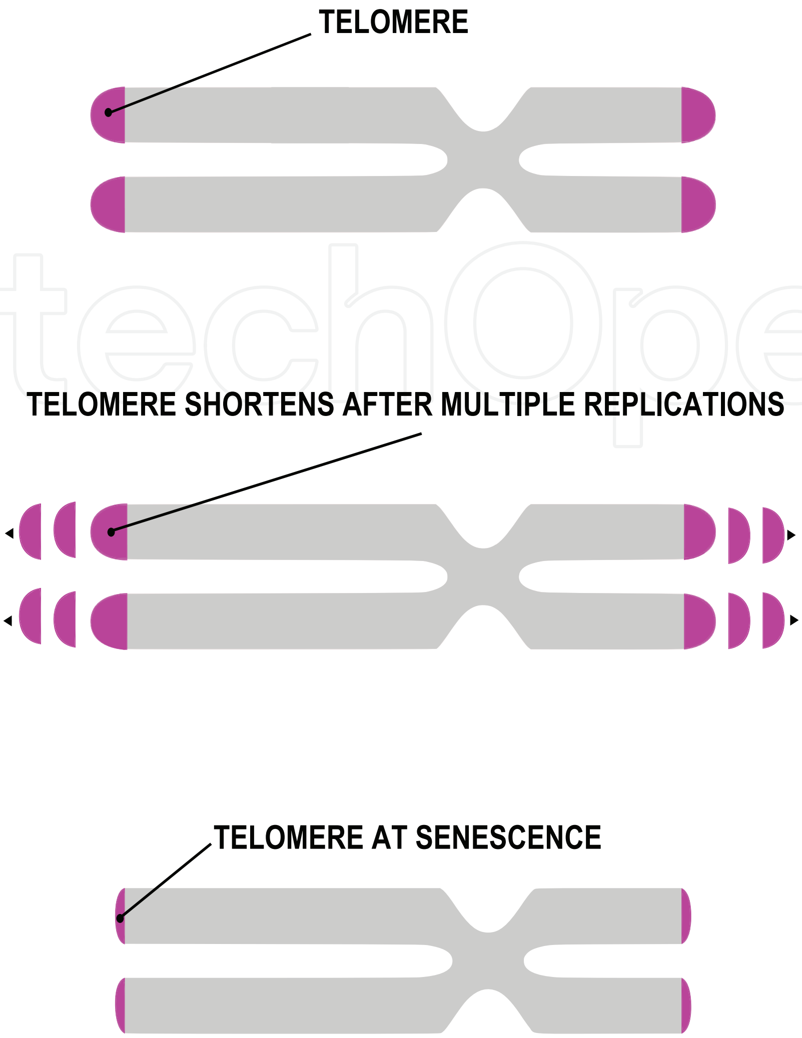

Figure 1. Telomere division process in human *T* cells.

The proposed scheme in Eq. (1) can provide PSO the same self-regulation capabilities during swarm's concentration around a local minimum (known as premature convergence) if we consider the swarm as lymphocytes, and particle concentration around a local minimum as *T* cells concentration. Finally, senescence of the particle will be translated as a new random-generated particle, replacing the unresponsive one. For this to be achieved, a given particle within the swarm will have a limited number of iterations to exist within the swarm, similar to telomere length at p_0, and senescence of this particle will occur when particle's capacity for search space-exploitation approaches to 0 and swarm's concentration around a local minimum has exceeded a given limit (premature convergence indicator), similar to *T* cells in human immune systems [35].

Based on this aging mechanism, it is possible to include senescence to a given particle within the swarm, and the lifespan of each particle will be adjusted according to the error produced by its candidate solution and the premature convergence indicator of the swarm.

3. Particle swarm optimization with a bio-inspired aging model (BAM-PSO)

The ALC-PSO variant [24] suggests an interesting approach using aging factors to define when to remove non-useful characteristics of the algorithm. However, this variant proposes a simple aging model with the following characteristics:

- The lifespan exists only for the *leader* and is controlled by linear means according to lifespan controller shown in [24].
- There is no lifespan controller for the rest of the particles of the swarm, meaning that the particles continue offering candidate solutions even if premature convergence has occurred.
- There is no communication between particles within the swarm about premature convergence and no evaluation is performed about particle concentration around local minima.

Consequently, there are several points that can be improved in ALC-PSO. For instance, including aging mechanisms to the rest of the particles within the swarm may help exploration without affecting the convergence. Moreover, in order to alleviate premature convergence in the PSO, there is an urgent need to include means of measuring the premature convergence in real time allowing the swarm to discard non-useful particles and to explore new candidate solutions without losing the convergence inertia toward the global minimum.

Based on previous observations, we propose a variant of the PSO named Particle Swarm Optimization with a Bio-inspired Aging Model (BAM-PSO). Our proposed algorithm considers the aging leader and challengers in the same fashion as ALC-PSO, but it applies senescence to each particle within the swarm by using the mathematical model that describes aging dynamics in Eq. (1).

For BAM-PSO to implement senescence efficiently, it is necessary to implement a mechanism that allows the algorithm to interpret when the swarm has reached a local minimum; this can be achieved by means of premature convergence measurement.

In the aging model represented by Eq. (1), the number of cells can be interpreted as a measure of the particles numbers around the same one-dimensional location. In this sense, measuring the standard deviation among the swarm in each particle dimension can be computed as follows:

$$k_j = \frac{k_{min}}{\sqrt{\frac{\sum_{i=1}^{N}\sum_{j=1}^{D}\left(x_{ij}-\overline{x_j}\right)^2}{(D-1)}}} \tag{2}$$

where $D \in \mathbb{R}$ is the dimension of the problem; k_{min} represents the deviation minimum for all dimensions; k_j is the premature convergence around *j-th* element of the dimension D. Note that x_{ij} is the particle within the swarm and $\overline{x_j}$ represents the mean value of all *j-th* elements of the swarm. This scheme provides a premature convergence measurement mechanism around each element of the problem dimension.

3.1. Lifespan controller

The BAM-PSO algorithm considers Eq. (2) to evaluate swarm's efficiency, and to control the lifespan of each particle, the aging mechanism proposed by [33] is adapted to the algorithm, satisfying the next criteria:

$$L_{ij} = L_{ij} - \alpha k_j \tag{3}$$

$$\text{With}: \ 0 \le L_{ij} \le L_{max}$$

where L_{ij} is the lifespan of the *i-th* particle with j-*th* element of dimension D. L_{max} represents the maximum lifespan of any particle within the swarm.

The error improvement of the particle with respect to the iteration t is calculated by:

$$\alpha = \frac{e_i(t)}{e_i(t-1)} \tag{4}$$

The error of the *i-th* particle $e_i(t)$ is computed within the swarm at iteration t.

This scheme completes the bio-inspired, population-broad aging mechanism and will allow us to propose the final algorithm.

The steps involved in the BAM-PSO algorithm are as follows:

Step 1: Initialization. The initial positions of all particles are generated randomly within the *n*-dimensional search space, with velocities initialized to *0*. The best particle among the swarm is selected as the *leader.* The age of the *leader* and all particles within the swarm is initialized to *0*.

Step 2: Velocity and position updating. Every particle follows the velocity update rule and the position update rule presented in [8]:

$$v_{ij}(t+1) = v_{ij}(t) + c_1 R_1 \left(p_{ij}(t) - x_{ij}(t)\right) + c_2 R_2 \left(p_{gj}(t) - x_{ij}(t)\right) \tag{5}$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \tag{6}$$

with:

$$i = 1, 2, \dots, S \qquad j = 1, 2, \dots, D$$

where i is the *ith* particle of a swarm that satisfies $S \in \mathbb{R}^D$, and j is the *jth* element of dimension problem D. Also t represents the iteration counter, R_1 and R_2 are random, normalized and uniformly distributed values. c_1, c_2 represents the social and cognitive parameter, $x_{ij}(t)$ is the particle *ij* position for t iteration, $x_{ij}(t+1)$ is the particle *ij* position for $t+1$ iteration, $v_{ij}(t)$ is the particle's *ij* velocity for t iteration. $p_{ij}(t)$ represents the local best position for particle *ij* in iteration t and $p_{gj}(t)$ represents the global best position for entire swarm in iteration t.

Step 3: Evaluate the *leader* or generate new challengers for leadership according to leadership term according to lifespan controller defined in [24]:

$$\delta_g Best = f\left(p_{gj}(t)\right) - f\left(p_{gj}(t-1)\right) \le 0 \tag{7}$$

$$\delta_l Best = \sum_{i=1}^{M} f\left(p_{ij}(t)\right) - \sum_{i=1}^{M} f\left(p_{ij}(t-1)\right) \le 0 \tag{8}$$

$$\delta_{Leader} = f(leader(t)) - f(leader\ (t-1)) \le 0 \tag{9}$$

with leadership term: $\theta = 1, 2, \dots, \Theta$.

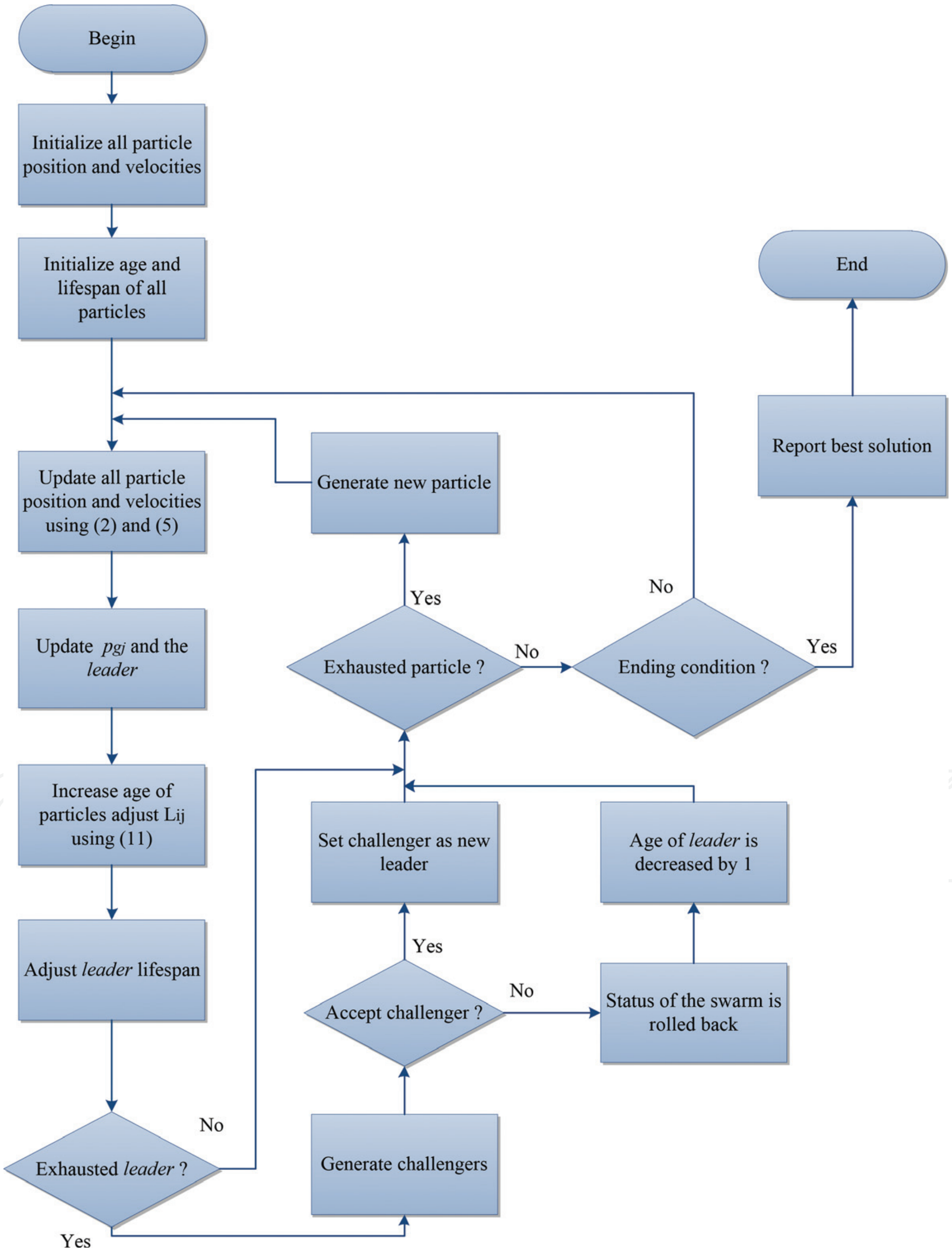
Begin
Initialize all particle position and velocities
Initialize age and lifespan of all particles
Update all particle position and velocities using (2) and (5)
Update p_{gj} and the *leader*
Increase age of particles adjust Lij using (11)
Adjust *leader* lifespan
Exhausted *leader* ?
No
Yes
Generate challengers
Accept challenger ?
Yes
No
Status of the swarm is rolled back
Set challenger as new leader
Age of *leader* is decreased by 1
Exhausted particle ?
Yes
No
Generate new particle
Ending condition ?
No
Yes
Report best solution
End

Figure 2. BAM-PSO flow diagram.

where $f(^*)$ represents the objective function value for the best candidate solution, θ the remaining leadership's term, Θ represents the maximum leadership term, δ_g *Best* defines the entire swarm improvement factor, δ_l *Best* represents the individual particle improvement factor, *leader* represents the particle within the swarm that is the acting leader (not necessarily $p_{gj}(t)$) and whose all particles will follow according to Eqs. (5) and (6); finally, δ_{Leader} represents the *leader's* individual improvement factor.

Eqs. (7), (8) and (9) indicate the leading performance of the *leader*. The lifespan controller utilizes these performance evaluations to adjust the leading term of *leader* according to the following decision tree:

if $\delta_g Best < 0$: $\theta = \theta + 2$ $(up\ to\ \Theta)$, else :

if $\delta_l Best < 0$: $\theta = \theta + 1$ $(up\ to\ \Theta)$.else :

if $\delta_{Leader}\theta < 0$: $\theta = \theta$ $(no\ increase)$, else :

$$\theta = \theta - 1\ (leader\ term's\ reduction).$$

When the leading term of *leader* reaches $\theta = 0$ the *leader* is considered exhausted and replaced by newly generated challengers as described in [24].

Step 4: Adjust lifespan of all particles within the swarm according to Eqs. (2)–(4) and replace particles with random ones for every depleted lifespan.

Step 5: Terminal condition check. If the number of iterations is larger than the predefined or the error has reached a minimum expected value, the algorithm terminates. Otherwise go to Step 2 for a new round of iteration.

Figure 2 shows the flow chart for BAM-PSO algorithm.

4. Results

The proposed BAM-PSO algorithm is compared with five different biologically inspired algorithms: PSO with inertial vector and boundaries [27], ant colony system (ACS) [6], differential evolution (DE) [31], simplified swarm optimization (SSO) [20] and particle swarm optimization with aging leader and challengers (ALC-PSO) [24]. These algorithms are selected because of several factors: first, PSO is the base algorithm for BAM-PSO, so it is natural to compare performance with the original optimizer, SSO and ALC-PSO are other well-known variants of PSO that in some way, claim to alleviate the premature convergence problem and, specifically, ALC-PSO is related in many ways to BAM-PSO. Finally, while ACS and DE are not related closely to BAM-PSO, they are swarm-based and evolution-based optimization algorithms respectively and thus, were considered as good candidates for performance comparison.

To test optimization performance of these algorithms, well-established benchmark functions are selected in low and high dimensionality [33]. These selected functions help evaluate algorithm's performance over a broad type of problems, because they possess multiple local minima, complex non-linear structure, or have bowl-shaped/plate-shaped structure [36, 37]; even some of them have a steep ridge and drops structure. From the literature, a list of 18 functions was considered relevant enough to test BAM-PSO performance. The selected benchmark functions are shown in **Table 1**.

For comparison purposes, all algorithms were configured in similar vein when it was possible, e.g. the ACS algorithm [6] uses *ant-type* vectors which can be considered particles in a *swarm* like those found in the PSO [8, 21], ALC-PSO [24] and the proposed BAM-PSO algorithms. Nevertheless, the behavior and setting are very different, since *ant-type* vectors behavior is determined by a mathematical model that simulates the pheromone attraction between biological ants. The DE algorithm [34] does not have a *swarm*-based mathematical model for the dynamics of particles, but instead the mathematical model used to simulate *evolution* is based on vectors and mutation factors. Finally, SSO algorithm [20] does not consider linear equations

f_1	Griewank	$1+\frac{1}{4000}\ \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right)$
f_2	Ackley	$-20e^{\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right)} - e^{\left(\frac{1}{n}\sum_{i=1}^{n} \cos 2\pi x_i\right)} + 20 + e$
f_3	Sphere	$\sum_{i=1}^{n} x_i^2$
f_4	Rastrigin	$A_n + \sum_{i=1}^{n} \left[x_i^2 + A\cos 2\pi x_i\right]$
f_5	Zakharov	$\sum_{i=1}^{n} x_i^2 + \left(\sum_{i=1}^{n} 0.5ix_i\right)^2 + \left(\sum_{i=1}^{n} 0.5ix_i\right)^4$
f_6	Trid	$\sum_{i=1}^{n} (x_i-1)^2 - \sum_{i=1}^{n} x_i x_{i-1}$
f_7	Sum of squares	$\sum_{i=1}^{n} ix_i^2$
f_8	Sum of exponential squares	$\sum_{i=1}^{n} \left[\left(\sum_{j=1}^{n} x_j^i\right) - b_i\right]^2$
f_9	Styblinski-Tang	$\frac{1}{2}\sum_{i=1}^{n} \left(x_i^4 - 16x_i^2 + 5x_i\right)$
f_{10}	Shifted Rastrigin	$\sum_{i=1}^{n} \left[A + \sum_{i=1}^{n} \left[x_i^2 - A\cos 2\pi x_i\right]\right.$
f_{11}	Schwefel 1.2	$\sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j\right)^2$
f_{12}	Elliptical	$\sum_{i=1}^{n} 10^{6\frac{i-1}{d-1}x_i^2}$
f_{13}	Rosenbrock	$\sum_{i=1}^{n} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i-1)^2\right]$
f_{14}	Schwefel	$418.9829d - \sum_{i=1}^{n} x_i sen\left(\sqrt{\lvert x_i \rvert}\right)$
f_{15}	Perm D, 0, β-function	$\sum_{i=1}^{n} \left(\sum_{j=1}^{n} \left(j^i + \beta\right)\left(\left(\frac{x_i}{j}\right)^i - 1\right)\right)^2$
f_{16}	Michalewicz	$-\sum_{i=1}^{n} sen(x_i) sen^{2m}\left(\frac{ix_i}{\pi}\right)$
f_{17}	Levy	$sen^2(\pi\omega_1) + \sum_{i=1}^{n-1} (\omega_i - 1)^2\left[1 + 10sen^2(\pi\omega_i + 1)\right.$ $\left. +(\omega_n - 1)^2[1 + sen2\pi x_n]\right]; \omega = 1 + \frac{x_i - 1}{4}$
f_{18}	Dixon-Price	$(x_i-1)^2 + \sum_{i=2}^{n} i\left(2x_i^2 - x_i - 1\right)^2$

Table 1. Benchmark functions used in algorithm performance comparison for BAM-PSO.

to update the information of the particles, instead a probability function is considered to decide the next particle position based on previously defined settings.

4.1. Evaluating the algorithms in low dimensional settings

The swarm size *S* for every algorithm is set to *20*, dimension *D* for every function is set to 2, and total iterations are set to *10,000* for each objective function. **Table 2** reveals the performance for the different selected algorithms in a low dimension, the results show the best possible solution offered by the algorithm after terminal condition was reached. As we can see, both BAM-PSO and ALC-PSO algorithms show improved performance in comparison to the other algorithms. Meaning that BAM-PSO provides good results in low dimensional problems for all the benchmark functions, outperforming most of the other tested algorithms. It is important to note, that results marked in Bold are the best solution obtained for each case.

4.2. Evaluating the algorithms in high dimensional settings

Our second simulation scenario consists in evaluating the performance of the BAM-PSO with high dimensional problems. In this case, the total of 18 benchmark functions from **Table 1** was considered and the function dimension *D* was configured to *30*.

Based on the previous results, ALC-PSO, SSO, and PSO algorithm were selected to compare results with the BAM-PSO because of their shared origin. However, ACS was also included due to its swarm nature.

At first glance, the results shown in **Table 3** suggest that the BAM-PSO provides the best performance of all compared algorithms in highly-dimensional problems for several benchmark functions. It is important to note, that results marked in Bold are the best solution obtained for each case.

4.3. Non-parametrical statistical analysis of BAM-PSO performance results

In order to conclude whether or not BAM-PSO outperforms the other selected algorithms, more accurate means of comparison other than simple observation of benchmark results are required; for this reason, some of the most popular non-parametric statistical tests were employed. This type of analysis is widely accepted as a metric of performance comparison between algorithms in a pair-wise configuration [43]. To this end, using the statistical procedures defined by [38–40], the Signed Test and the Wilcoxon Test statistical analysis were selected.

Dimension = 2	BAM-PSO	ALC-PSO	DE	SSO	ACS	PSO
f_1	**0.000000E + 00**	**0.000000E + 00**	7.800000E-03	4.250000E-02	4.880000E-09	**0.000000E + 00**
f_2	**8.880000E-16**	**8.880000E-16**	2.120000E + 00	4.300000E-02	4.160000E-02	**8.880000E-16**
f_3	**0.000000E + 00**	**0.000000E + 00**	2.800000E-01	5.460000E-06	5.460000E-08	3.550000E-43
f_4	**0.000000E + 00**	**0.000000E + 00**	2.610000E + 00	4.300000E-03	7.300000E-01	**0.000000E + 00**

Table 2. Optimization results comparison for D = 2.

Dimension = 30	BAM-PSO	ALC_PSO	PSO	SSO	ACS
f_1	3.420000E-01	4.810000E-02	**1.664995E-02**	1.017972E + 00	1.009513E + 00
f_2	1.710000E + 00	2.950000E + 00	**6.179332E-01**	1.936833E + 01	1.575243E + 01
f_3	9.720000E-04	1.060000E + 00	**1.054778E-05**	3.628362E + 02	2.766939E + 02
f_4	**3.830000E + 00**	1.420000E + 03	2.003442E + 02	1.421872E + 06	4.577661E + 05
f_5	**2.701825E-07**	1.943493E-01	1.314056E + 01	6.189616E + 02	5.320685E + 05
f_6	**1.000012E + 00**	1.347625E + 04	5.670018E + 03	2.350724E + 06	3.796814E + 06
f_7	**1.838826E-25**	1.846162E-01	8.309639E-01	5.267949E + 03	7.131057E + 03
f_8	**1.693373E-15**	5.924127E-08	2.569830E-10	5.672124E-02	2.042905E-01
f_9	1.591141E + 00	**4.989256E-03**	1.084663E + 03	6.528900E + 02	4.572974E + 02
f_{10}	**4.265668E-03**	2.575325E-01	6.748519E + 01	2.362575E + 01	6.768979E + 01
f_{11}	1.000007E + 00	**0.000000E + 00**	8.449218E + 120	3.61590E + 115	1.11140E + 121
f_{12}	**1.00008E + 119**	9.432255E + 119	2.858556E + 125	1.96354E + 124	3.06278E + 125
f_{13}	**1.000053E + 00**	9.837181E + 00	5.033260E + 01	3.015314E + 03	3.832691E + 03
f_{14}	**1.005478E + 00**	2.327770E + 03	9.314781E + 03	8.149385E + 03	1.062061E + 04
f_{15}	2.039775E-23	**0.000000E + 00**	8.520128E-06	4.815499E-04	9.508960E + 02
f_{16}	**9.177340E + 00**	1.276359E + 01	1.683612E + 02	1.086427E + 02	4.458898E + 01
f_{17}	**1.000155E + 00**	6.796335E + 00	1.001832E + 00	1.026017E + 02	1.676198E + 02
f_{18}	**9.112266E-01**	2.056725E + 02	4.960351E + 00	7.179904E + 05	8.280520E + 05

Table 3. Optimization results comparison for D = 30.

In **Table 4**, we can observe that BAM-PSO outperforms the other algorithms with an accepted level of significance using this procedure. However, this test is a simple first-line procedure and to uncover more evidence over the results, we rely on a more robust and sensitive procedure, which is the Wilcoxon Test.

The Wilcoxon test results shown in **Table 5** shed light over the fact that BAM-PSO algorithm can go beyond the results provided by the PSO, SSO and ACS with great statistical significance ($P < 0.001$), but the procedure finds not enough evidence to conclude that the BAM-PSO can outperform the ALC-PSO at this level of statistical significance; however, the results are good enough to show that BAM-PSO can outperform ALC-PSO with good statistical significance ($P < 0.01$), and considering the literature claim that: if the resulting P-value is small enough ($P < 0.05$), then it can be accepted that the median of the differences between the paired

BAM-PSO vs.	ALC_PSO	PSO	SSO	ACS
Positive results	14	15	18	18
Negative results	4	3	0	0
Significant difference? (P < 0.05)	**Yes**	**Yes**	**Yes**	**Yes**

Table 4. Non-parametrical sign test for benchmark results at D = 30.

BAM-PSO	(R+) positive ranks obtained	(R−) negative ranks obtained	(Rm) maximum negative ranks	Accepted significance? ($P < 0.05$)
vs ALC-PSO	147.0	24.0	27 for ($P < 0.01$)	**Yes**
vs PSO	156.0	15.0	17 for ($P < 0.001$)	**Yes**
vs SSO	171.0	0.0	17 for ($P < 0.001$)	**Yes**
vs ACS	171.0	0.0	17 for ($P < 0.001$)	**Yes**

Table 5. Non-parametrical Wilcoxon test for benchmark results at D = 30.

observations is statistically significantly different from 0 [44]. We can conclude then, that BAM-PSO has a greater performance over a broad set of benchmark functions over all other selected algorithms with statistical relevance, including ALC-PSO.

The performance of BAM-PSO can be explained by its senescence mechanism: after particles falls into local minimum, they offer less improvement; then, the senescence mechanism starts acting by producing *senescence* on the swarm; then, exhausted particles are replaced with random ones through the search space. This favors exploration after premature convergence without completely eliminating exploitation of search space near the local minimum, which in the end provides better optimization results than other PSO variants.

5. Conclusions

In this chapter, we introduced a PSO variant algorithm called Particle Swarm Optimization with Bio-inspired Aging Model (BAM-PSO) which was compared with other five popular bio-inspired optimizers. This test was performed using popular benchmark functions with low and high dimensionality configuration.

We observed that the BAM-PSO algorithm has the potential to solve the premature convergence problem of PSO showing good results for both low and high dimensional problems with statistical relevance according to several non-parametric analyses. Furthermore, according to results shown in Section 4, BAM-PSO performs better than the selected PSO variants.

As shown in results section, BAM-PSO outperforms all other compared swarm-based algorithms with at least a confidence factor as high as $P < 0.01$. However, the cost of this improved accuracy is found in computation complexity due to the introduction of Eq. (2) and all the lifespan control for the particles; which in turn translates to computing time; this time increase was found to be approximately of at least 9 times the required computation time for the original PSO on the conducted experiments of section 5 and 1.5 times the required computation time for ALC-PSO. However, this increase in time is not fixed, as it depends on how early the premature convergence occurs and how many particles are replaced after senescence.

Finally, these experimental results provide support on the important role of aging mechanisms during the selection process in bio-inspired optimization algorithms, because the population-broad aging mechanism implemented in BAM-PSO allows the algorithm to provide better results than some other popular optimizers that does not implement aging.

Acknowledgements

The authors thank the support of CONACYT Mexico, through Projects CB256769 and 340 CB258068 (Project supported by Fondo Sectorial de Investigación para la Educación).

Author details

Eduardo Rangel-Carrillo[1], Esteban A. Hernandez-Vargas[2], Nancy Arana-Daniel[1], Carlos Lopez-Franco[1] and Alma Y. Alanis[1]*

*Address all correspondence to: almayalanis@gmail.com

1 CUCEI, Universidad de Guadalajara, Guadalajara, Jalisco, Mexico

2 Frankfurt Institute for Advanced Studies, Frankfurt am Main, Germany

References

[1] Von Neumann J. Zur theorie der gesellschaftsspiele. Mathematische Annalen. 1928;**100**(1):295-320. DOI: 10.1007/BF01448847

[2] Maynard J, Price R. The logic of animal conflict. Nature. 1973;**5427**(246):15-18. DOI: 10.1038/246015a0

[3] Henson S, Hayward J. The mathematics of animal behavior: An interdisciplinary dialogue. Notices of the AMS. 2010;**57**(10):1248-1258

[4] Barricelli N. Esempi numerici di processi di evoluzione. Methodos. 1954;**6**(21–22):45-68

[5] Goss S, Aaron S, Deneubourg J, Pasteels J. Self-organized shortcuts in the angentine ant. Naturwissenschaften. 1989;**76**(12):579-581. DOI: 10.1007/BF00462870

[6] Colorni A, Dorigo M, Maniezzo V. Distributed optimization by ant colonies. Proceedings of ECAL91. 1991;**1**(1):134-142. DOI: 10.1007/BF00462870

[7] Socha K, Dorigo M. Ant colony optimization for continuous domains. European Journal of Operational Research. 2008;**1**(1):1155-1173. DOI: 10.1016/j.ejor.2006.06.046

[8] Kennedy J, Eberhart R. Particle swarm optimization. Proceedings of the IEEE International Joint Conference on Neural Networks. 1995;**4**(1):1942-1948. DOI: 10.1109/ICNN.1995.488968

[9] Vrahatis M. Particle Swarm Optimization and Intelligence, Advances and Applications. 1st ed. Hershey, PA, USA: Information Science Publishing (IGI Global); 2010. p. 328. DOI: 10.13140/2.1.3681.1206

[10] Castro L. Fundamentals of natural computing: An overview. Physics of Life Reviews. 2007;**4**(1):1-36. DOI: 10.1016/j.plrev.2006.10.002

[11] Simon D. Evolutionary Optimization Algorithms. 1st ed. 1. New Jersey, USA: John Wiley & Sons; 2013. 772 p. DOI: 978-0-470-93741-9

[12] Gershon H, Gershon D. The budding yeast, saccharomyces cerevisiae, as a model for aging research: A critical review. Mechanisms of Ageing and Development. 2000;**120**(1–3): 1-22. DOI: 10.1016/S0047-6374(00)00182-2

[13] Fisher R. The Genetical Theory of Natural Selection. 1st ed. New York, USA: Nabu Press; 1930. p. 360. DOI: 0-19-850440-3/ISBN 9780198504405

[14] Darwin C. The Origin of Species. 1st ed. United Kingdom of Great Britain and Ireland: Fellow of the Royal, Geological, Linnaean, Etc., Societies; 1859. p. 502. DOI: 10.5962/bhl.title.46292

[15] Mendel G. Verhandlungen des naturforschenden ver-eines in Brunn. Versuche Uber Plflanzen-hybriden. 1865;**IV**(1):3-47. DOI: 10.5962/bhl.title.61004

[16] Grosan C, Abraham A. Hybrid evolutionary algorithms: Methodologies, architectures, and reviews. Studies in Computational Intelligence. 2007;**75**(1):1-17. DOI: 10.1007/978-3-540-73297-6-1

[17] Van Den Bergh F, Engelbrecht A. A cooperative approach to particle swarm optimization. IEEE Transactions on Evolutionary Computation. 2004;**8**(3):225-239. DOI: 10.1109/TEVC.2004.826069

[18] Meissner M, Schmuker M, Schneider G. Optimized particle swarm optimization (OPSO) and its application to artificial neural network training. BMC Bioinformatics. 2006;**7**(1): 1-11. DOI: 10.1186/1471-2105-7-125

[19] Jiang B, Wang N, Li X. Particle swarm optimizer with aging operator for multimodal function optimization. International Journal of Computational Intelligence. 2014;**6**(5):862-880. DOI: 10.1080/18756891.2013.807108

[20] Changseok B, Wei-Chang Y, Noorhanizan W, Yuk-Ying C, Liu Y. A new simplified swarm optimization. International Journal Of Innovative Computing Information And Control. 2012;**8**(6):4391-4406. DOI: 1349-4198

[21] Chen G, Huang K. Online parameter identification of an induction motor using improved particle swarm optimization. In: IEEE, editor. IEEE Control Conference; 26–31-07-2007; Hunan, China. Hunan, China: IEEE; 2007. pp. 745-749. DOI: 10.1109/CHICC.2006.4347151

[22] Liu C, Du W, Wang W. Particle swarm optimization with scale-free interactions. PLoS One. 2014;**9**(5):1-8. DOI: 10.1109/4235.985692

[23] Rahman H. Ainul Hayat by Muhammad Ibn Yusuf Al-Harawi. 1st ed. India: Ibn Sina Academy of Medieval Medicine and Sciences; 2007

[24] Wei-Neng C, Jun Z, Ying L, Chen N, Zhi-Hui Z, Henry S, et al. Particle swarm optimization with aging leader and challengers. IEEE Transactions on Evolutionary. 2013;**17**(2): 241-258

[25] Williams G. Pleiotropy, natural selection, and the evolution of senescence. Evolution. 1957;**11**(4):398-411. DOI: 10.2307/2406060

[26] Clerc M, Kennedy J. The particle swarm – explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation. 2002;**6**(1):58-73. DOI: 10.1109/4235.985692

[27] Eberhart R, Shi Y. Comparison between genetic algorithms and swarm optimization. Evolutionary programming VII. 1998;**1447**(1):611-616. DOI: 10.1007/BFb0040812

[28] Van den Bergh F, Engelbrecht A. A new locally convergent particle swarm optimiser. In: IEEE, editor. IEEE International Conference on Systems, Man and Cybernetics; 6–9 Oct 2002; Yasmine Hammamet, Tunisia, Tunisia. Yasmine Hammamet, Tunisia, Tunisia: IEEE; 2002. DOI: 10.1109/ICSMC.2002.1176018

[29] Prasad B, Mandal D, Ghoshal SP. PSO with aging leader and challengers for optimal design of high speed symmetric switching CMOS inverter. International Journal of Machine Learning and Cybernetics. 2016;**8**(4):1403-1422. DOI: 10.1007/s13042-016-0517-z

[30] Rangel E, Alanis AY, Ricalde LJ, Arana-Daniel N, Lopez-Franco C. Bio-inspired aging model particle swarm optimization neural network training for solar radiation forecasting. Progress in pattern recognition, image analysis, computer vision, and applications. CIARP 2014. Lecture Notes in Computer Science. 2014;**8827**(1):682-689. DOI: 10.1007/978-3-319-12568-8_83

[31] Arana-Daniel N, Villaseñor C, Lopez-Franco C, Alanis AY. Bio-inspired aging model-particle swarm optimization and geometric algebra for structure from motion. Progress in pattern recognition, image analysis, computer vision, and applications. CIARP 2014. Lecture Notes in Computer Science. 2014;**8827**(1):762-769. DOI: 10.1007/978-3-319-12568-8_83

[32] Partridge L, Gems D. Benchmarks for aging studies. Nature. 2007;**450**(7167):165-167. DOI: 10.1038/450165a

[33] Romanyukha A, Anatoli I. Age related changes in population of peripheral T cells: Towards a model of immunesenescence. Mechanisms of Ageing and Development. 2013;**124**(1):433-443. DOI: 10.1016/S0047-6374(03)00019-8

[34] Storn R. Kenneth. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization. 1997;**11**(4):341-359. DOI: 10.1023/A:1008202821328

[35] Cibelli J, Blackwell C, Cristofalo V, Francis M, Baerlocher G, al e. Extension of cell life-span and telomere length in animals cloned from senescent somatic cells. Science. 2000;**288**(1):665-669. DOI: 10.1126/science.288.5466.665

[36] Liang J, Qu B, Suganthan P, Hernández-Díaz A. Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session and Competition on Real-Parameter Optimization. Technical Report 2012 on Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University. 2013;**1**(1)

[37] Wei-Hong L, Ashidi N, Isa M. Adaptive division of labor particle swarm optimization. Expert Systems with Applications: An International Journal. 2015;**42**(14):5887-5903. DOI: 10.1016/j.eswa.2015.03.025

[38] Ghosh S, Das S, Kundu D, Suresh K, Abraha A. Inter-particle communication and search-dynamics of lbest particle swarm optimizers: An analysis. Informative Science. 2012;**182** (1):156-168. DOI: 10.1016/j.ins.2010.10.015

[39] Derrac J, García S, Molina D, Herrera F. A practical tutorial on the use of nonparametric statistical tests as a methodology. Swarm and Evolutionary Computation. 2011;**1**(1):3-18. DOI: 10.1016/j.swevo.2011.02.002

[40] Blaise L. Technische Universität Carolo-Wilhelmina zur Braunschweig. Master thesis [thesis]. Germany; 2015

[41] Poblete-Castro I, Binger D, Rodrigues A, Becker J, Martins Dos Santos V, Wittmann C. In-silico-driven metabolic engineering of pseudomonas putida for enhanced production of polyhydroxyalkanoates. Metabolic Engineering. 2013;**15**(1):113-123. DOI: 10.1016/j.ymben.2012.10.004

[42] Hernandez-Vargas E, Esther W, Canini L, Toapanta F, Binder S, Uvarovskii A, al e. Effects of aging on influenza virus infection dynamics. Journal of Virology. 2014;**88**(8):4123-4154. DOI: 10.1128/JVI.03644-13

[43] Johnson D. The insignificance of statistical significance testing. Journal of Wildlife Management. 1999;**63**(3):763-772. DOI: 10.2307/3802789

[44] Conover W. Practical Nonparametric Statistics. 3rd ed. New Jersey, USA: Wiley & Sons; 1999. p. 584. DOI: 978-0471160687