2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

A new Self-adaptive PSO based on the identification of planar regions

Eddy Mesa, Juan D. Velásquez and Patricia Jaramillo

***Abstract*— In this paper, we propose a new approach for self-adaptive particle swarm optimization, using the function's topology to adapt the parameters and modifying them when a planar region is identified in the objective function. Particle swarm optimization is a metaheuristic developed to optimize nonlinear problems. This metaheuristic has four parameters to adapt the search for the different optimization problems. However, finding an optimal set of parameters is not a trivial problem. Some strategies to adapt the parameters have been developed, but they are not robust enough to cover all kinds of problems. Function's topology is one of the most decisive factors in order to choose a right set of parameters; i.e. convex functions need more exploitation because this topology offers a clear direction to the minimum point. In the opposite way, a noise function can be trapped in a local minimum for the same level of exploitation. In order to validate and compare our methods, we use the benchmark functions from CEC 2005 to compare the different particle swarm optimization versions. The results show that the proposed version is significant better than the original particle swarm optimization and the standard particle swarm optimization proposed in 2011.**

I. INTRODUCTION

Particle swarm optimization (PSO) is a popular metaheuristic, widely used to solve real-world optimization problems and proposed in 1995 by Kennedy and Eberhart [1]. Inspired in behavior of birds, PSO uses as other metaheuristics parameters to arrange the search. Nevertheless, selecting the parameters for metaheuristics is an optimization problem itself. This problem is usually solved using control parameters techniques [2], [3], [4]. Often classical metaheuristics have some self-adaptive versions. PSO is not an exception. The PSO adaptive version introduces a strategy based on stochastic functions and hybrids with other metaheuristic or gradient methods[5], [4].

Globalization strategies for metaheuristics are often linked with balance between exploration and exploitation [6], [7], [8], and the probability to find attraction area (A) [9]. This area (A) is defined as the region that provides information about optimal point location. For instance, in a convex unimodal function, all feasible solutions give information about the minimum. The parameters of metaheuristics are a possible way to change this balance, because a right set of parameters favors the probability to find a solution from the region A. When a point in A is found, the algorithm gets information about optimal point and exploits it. In many cases, strategies to control the parameters use iterations, increasing the exploitation while exploration is reducing loop by loop [5], [10], [11]. However, landscape of objective function information is used only in a few approaches to calculate the best parameters to optimize the problem. For PSO the approach directed to landscape is called landscape adaptive particle swarm optimizer (LAPSO) [12]. In this case, a new parameter is introduced to improve the step length and force the individual to turn to the slope given by minimum and maximum found by the solutions of the population. It uses the information of slope found by minimum and maximum, but there is not a modification to increase exploration. Conversely, the exploitation is increased for all kinds of functions.

Each topology has a different kind of attraction area A. A general classification can be: large areas, which are highly probable to find (convex functions, quasi convex functions, etc.) and small areas, which have small probability to be found. In any case, big attraction areas are not always easy to optimize. Then there are attraction areas with high-quality information, e.g. convex functions where decedent direction is clear, and attraction areas with low-quality information like planar regions. In the last case, descent direction is not clearly defined, e.g. Rosenbrock function. Notice that, it is important to increase the exploitation in big attraction areas with high-quality information. In contrast, for low-quality areas, increase the exploration is recommended. In a previous work, we present a criterion to determine when the function probably has planar regions that might be problematic to optimize [13].

Regularly, the strategies for control parameters take into account the number of iterations, but if the function to control the parameters would uses: iterations and the shape of the fitness function; probably, the search could improve in terms of results and robustness. In this paper, we propose a self-adaptive strategy for PSO parameters based on the information about topology of the function that is provided from population and iterations. The strategy consists in identifying planar regions using the current population; the solution changes iteration by iteration giving new information about topology. If the landscape shows planar regions, i.e. regions with poor-quality information about optimal solution, then parameters to increase exploration are chosen. In the opposite case (high-quality information), parameters to increase the exploration are selected.

This paper is organized as follows. In section II, we present the materials and methods where we describe the algorithms, criterion and the proposed functions to adapt the parameters. In section III, we describe the experimental design, and the results obtained and discuss them. And finally, conclusions and future work are presented at section IV.

Facultad de Minas, Universidad Nacional de Colombia, Medellín, Colombia (email: {ejmesad,jdvelasq,gpjarami}@unal.edu.co).

II. MATERIALS AND METHODS

A. PSO algorithm

The original version of PSO, inspired by social behavior of birds and proposed in 1995 has evolved to many different versions [1]. Currently, there are standard versions proposed by Clerc [14] based on the improvements described in the literature. Pseudocode 1 describes the general steps of canonical PSO. Each version changes in different parts of the algorithm specially in the lines 1,5 and/or 6 of the Pseudocode 1.

The original algorithm proposed in 1995 [1] uses a random population uniformly distributed the expression in Line 1 will be:

$$\mathbf{v}_i^{(t+1)} = \omega v_i^t + c_1\mathbf{u}_1 \otimes (\mathbf{p}_i^t - \mathbf{x}_i^t) + c_2\mathbf{u}_2 \otimes (\mathbf{g}_i^t - \mathbf{x}_i^t) \quad (1)$$

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

where $\mathbf{x}_i^t$ is the current population and $\mathbf{x}_i^{(t+1)}$ is the new population. $\mathbf{v}_i^{(t+1)}$ is new velocity and $\mathbf{v}_i^t$ is the current velocity. ω is the parameter that controls the amplification in the current direction. $\mathbf{u}_1$ and $\mathbf{u}_2$ are random vectors following uniform distribution. $\mathbf{p}_i^t$ and $\mathbf{g}^t$ are the best particle found for the particle i and the best result found for entire population respectively. Eqs. 1 and 2 are used to calculate velocity and the new population in Lines 5 and 6 respectively.

Pseudocode 1 Canonical PSO

Require: f,LB,UP,C1, C2, w, N, max-iterations;
1: Initial population
2: **while** $i \leq$ max-iterations **do**
3: fitness function
4: select pbest and gbest
5: calculate velocity according to PSO version
6: new population according to PSO version
7: **end while**

Two main failures have been detected for original PSO. The first one, it is getting trapped in local minimums; which is boarded for different versions like: comprehensive-learning PSO (CLPSO) [15] , LAPSO, the hybrid between differential evolution and PSO (DE-PSO), among others [16]. The second one is high computational effort [14]. Computational effort is described as the required calls to objective function. Some versions try to improve computational efficiency, e.g. μ-PSO, coordinate PSO[14], [17]. The generalizable and robustness improvements of PSO are summarized in the last official version of PSO, called standard particle swarm optimization SPSO-2011 [18], [19].

SPSO-2011 version includes some improvements: random topology paradigm(here topology are referring to neighborhood of the particles) and the change of the hyper-cube idea to hyper-sphere idea [20]. The mixture of these improvements change Line 1 of Pseudocode 1. In this case, the velocity is calculated using random topology paradigm [14]. Also, Line 5 is changed by introducing a new term in the Eq. 1 given by

$$\mathbf{G}_i^t = \frac{3\mathbf{x}_i^t + c_1\mathbf{u}_1 \otimes (\mathbf{p}_i^t - \mathbf{x}_i^t) + c_2\mathbf{u}_2 \otimes (\mathbf{g}_i^t - \mathbf{x}_i^t)}{3} \quad (3)$$

Then, hyper-sphere $\mathcal{H}_i(\mathbf{G}_i^t, \|\mathbf{G}_i^t - \mathbf{x}_i^t\|)$, where $\mathbf{G}_i^t$ is the center of hyper-sphere and the radius of the hype-sphere is $\|\mathbf{G}_i^t - \mathbf{x}_i^t\|$

In Line 5 of Pseudocode 1 the equation is modified from Eq. 1 to Eq. 4 and Line 6 does not change. Thus, the version that we use to test and validate the proposed strategy is SPSO-2011.

$$\mathbf{v}_i^{(t+1)} = \omega\mathbf{v}_i^{(t+1)} + \mathcal{H}_i(\mathbf{G}_i^t, \|\mathbf{G}_i^t - \mathbf{x}_i^t\|) - \mathbf{x}_i^t \quad (4)$$

B. PSO parameters and some previous adaptations

PSO has four parameters: Size of population (N), ω, c_1 and c_2 (See Eq. 1). The size of population, as a parameter, is important because it helps to reduce computational effort [17]. However, we focus in local minimum trap. Then, we propose a strategy to control the other three parameters, which are directly related with the balance between exploitation and exploration [1], [10], [14].

ω is the amplification of the current direction of the particle i. If a particle is directed only by $\omega\mathbf{v}_i^t$ this particle follows the line with direction $\mathbf{v}_i^t$ jumping with a step with magnitude of $\omega\mathbf{v}_i^t$. Consequently, big values for ω favor exploration whereas small values favor exploitation.

c_1 and c_2 magnify the influence of the best point found by the particle i and the best point found by the whole population, respectively. if $c_1 = c_2$ the importance of local minimum is equal to the global minimum. The random vectors $\mathbf{u}_1$ and $\mathbf{u}_2$ are not considered parameters, but they affect the search introducing noise that increases exploration.

Historically, there are a hundred of different versions including new parameters, strategies to control parameters and strategies to determine parameters [4], [12], [11], [14]. For the sake of brevity, we mention only two paradigms for each parameter. we consider the parameters chosen are the the closest and most useful for our approach.

The equations to control ω are:

- $\omega = \omega_{max} - (\omega_{max} - \omega_{min})\frac{t}{T}$ [21]
- $\omega = \frac{0.6}{1+(log(t))^2}$ [11]

where $\omega_{max} = 0.97$ and $\omega_{min} = 0.4$ are the minimum and maximum values recommended for ω [11] and T is the maximum number of iterations.

The equations to control c_1 and c_2 are

- $c_1 = c_2 = constant$ [1]
- $c_1 = 3 - (3 - 0.5)\frac{t}{T}$ and $c_1 = 0.5 + (3 - 0.5)\frac{t}{T}$ [4]
- $c_1 = 1.3$ and $c_2 = 2.8$ [23]

C. Planar regions criterion

Each iteration gives us information about N solution, where N is the size of the population. We use the individual information, e.g. $\mathbf{p}_i^t$ and $\mathbf{g}^t$. However, the information of whole population is not used, i.e. statistics measure as mean, distributions or median. Mesa, Velasquez and Jaramillo proposed a criterion to use the information given by statistical measures to determine when the function has a problematic planar region [13].

The criterion uses mean and median to determine the kind of function that the algorithm is searching for. How does it work? In Fig. 1, the possible cases are defined. For part (a) Range $R_g > 0$ and the mean and median are approximately equal; this means that the distribution of particles over the range is uniform. The uniformity is associated to functions with strong minimums and clear descendent directions. In the case (b), $R_g > 0$ and the mean and median are different values. The difference between both measures shows that there are outlier points. Also means that, the most part of population is near to median. For a random population, this indicates that there is a pretty flat region near to median value.

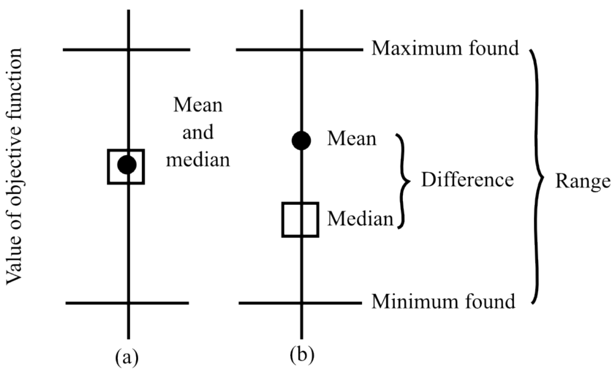

Fig. 1: Cases for criterion. (a). mean and median are approximately equal. (b).mean and median are difference

The concept of criterion was described above, but there are not punctual limits to determine which actually are or not are a problematic region. In [13] the percentage difference between mean and median (m) is defined as:

$$m = \frac{mean(f) - median(f)}{R_g} \quad (5)$$

where mean and median are calculated with value of the objective function for current population and R_g is the range defined as $max - min$. This measure is affected by the dimension. An empirical equation, presented in [13], to delimitate problematic or not problematic function according to dimensions is:

$$\%MM \geq 7.675(D)^{-0.588} \quad (6)$$

where D is the dimensions of the problem. Moreover, if $\%MM$ satisfy eq. 6, then the function presents a flat topology. Also, the range has a meaning. If the range is equal or so close to zero, the region is completely flat. While, the range $R_g > 0$ indicates a different topology of the function. In addition, authors in [13] also conclude that the minimal N is 30 because for N under 30 the mean and median are not consistent for different random populations [13].

D. How do parameters work in PSO?

PSO has four parameters to control the search. We focus on three of them (c_1, c_2 and ω), as mentioned above. They affect the search as follow: when c_1 and ω decrease, and/or c_2 decrease the exploitation of the algorithm increase. When the parameters move in the opposite way the exploration of the algorithm increase.

In Table I, recommended values in the literature for c_1, c_2 and ω are presented. Using these parameters, we plot the behavior of the PSO's population for two sets of parameters with the aim to know the influence of them in the balance between exploration and exploitation. The Figs. 2 and 3 show the influence of parameters for two different functions f_1 and f_6. f_1 is a translated sphere; this function is convex, unimodal and smooth. In contrast, f_6, a translation of Ronsebrock's function, is non-separable, unimodal (only for 2D), uniform and has the banana region, which is planar. The extreme values of the parameters present in Table I are used. The size of the population is 25 and total iteration are 6; Iterations 2, 4, and 6 are shown in the figures mentioned above.

In Fig. 2, we show the behavior for function 1 and 6. The left column of the Fig. 2 presents the population behavior for $\omega = 0.4$. On this occasion, population covered a small part of a feasible area. Quickly, the best point is followed and other parts are not covered by the search. On the other hand, the opposite situation is shown in the right column of the Fig. 2 where a larger area is covered for the population. In this case, the exploration for f_6 is bigger than the exploration for f_1; this means that, for a function with a descendent direction defined the ω influence is smaller than in flatter functions.

In Fig. 3, the behavior for functions 1 and 6 is presented as in the previous figure. The left column of the Fig. 3 shows the population when c_2 and ω are constant and c_1 change from 2 to 2.8. For the function f_1, there is not a clear change in the exploration balance of the population. In contrast, f_6 exploration is increased proportional to c_1; i.e., when descendent direction is not defined, an increment of c_1 augments the exploration of the algorithm.

TABLE I: RECOMMENDED VALUES FOR PARAMETERS OF PSO

Parameter	Value	Reference
ω	$\omega_{max} = 0.9$and $\omega_{min} = 0.4$	[11]
c_1	$c_1 = 2$ or $c_1 = 1.193$	[1], [14]
c_2	$c_2 = 2$ or $c_2 = 1.193$	[1], [14]
$c_1$and c_2	$c_1 + c_2 \leq 4$	[14]

E. Proposed self-adapting strategy

In this section, we propose a strategy to adapt c_1, c_2 and ω. The current version SPSO2011 presents a good performance for the most part of unimodal functions (translate sphere, Schwefel 1.2 function, [22]). The traditional parameters achieve a good balance between exploration and exploitation. Therefore, these parameter should remain in our strategy.

The strategy has two parts, the first one is to keep the current behavior which is good for convex and unimodal functions. The second one is to adapt the parameter to improve the performance for the functions which have a low-quality information for the attraction area. Following the criterion presented in section 2.C., m is calculated classifying the function in problematic planar or non planar. When the function is non-planar, we remain the parameters. In contrast, for planar regions the parameters must change.

The exploration of the algorithm increase by:

- Increase ω
- Increase c_1 for planar regions
- Decrease c_2

when the exploitation increase is desired, the parameters must move in the opposite way. Also, it is important to remind that when exploration increases exaggeratedly, the algorithm may not converge.

Thinking in the last items, the proposed strategy have two main points. First, reduce the risk of an infinite growth of exploration. And second, increase the exploration for functions which are consider as planar regions following the criterion.

ω will decrease according to the number of iterations with the aim of stopping exploration growth. In this paper, we defined such control function for the parameter ω the following sigmoid function:

$$\omega = \frac{1}{1.8 + e^{\frac{8t}{(T-1)-4}}} + 0.4 \tag{7}$$

where T and t are maximum number of iterations and current iteration respectively. This function takes the values of 0.4 for $t = T$ and 0.9 for $t = 1$. From $t = 1$ to $t = T$ the value of ω decrease smoothly, preventing the failure because of convergence.

In any case, the exploration should be augmented for planar regions and kept the current parameter for non-planar region. For problematic planar regions detected c_1 should increase and c_2 should decrease. It is important to notices that the difference between mean and median can be positive or negative. If it is positive, outlier points are bigger than median and the exploration should increase. Moreover, negative difference means that the outlier point is smaller than median; then, exploitation should augment. The control equation due increase c_2 for negative values and decrease it for positive values and c_1 should move in the opposite way. A piecewise function is defined as follows:

$$\begin{aligned} c_1 &= \begin{cases} m < -\%MM & \frac{1}{0.56+e^{m+1}} + 1 \\ -\%MM \leq m \leq \%MM & 1.193 \\ m > \%MM & \frac{1}{0.56+e^{m-1}} + 1 \end{cases} \\ c_2 &= \begin{cases} m < -\%MM & \frac{1}{0.56+e^{-m-1}} + 1 \\ -\%MM \leq m \leq \%MM & 1.193 \\ m > \%MM & \frac{1}{0.56+e^{-m+1}} + 1 \end{cases} \end{aligned} \tag{8}$$

where $\%MM$ is the limit proposed in the Eq.6 and m is the calculation of the difference between mean and median in the current population, presented in Eq. 5. Eq. 8 is represent in the Fig. 4. In addition, the parameters for non-planar regions are the current parameter for SPSO2011.

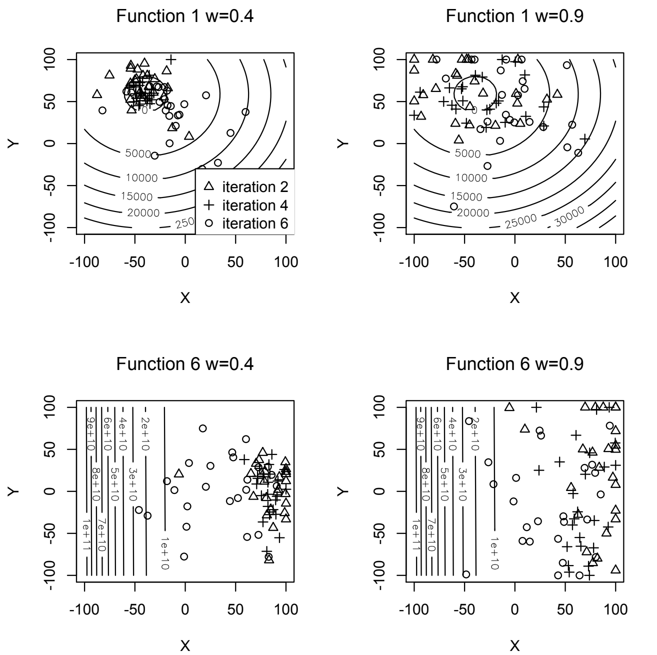

Fig. 2: Population for PSO with $c_1 = c_2 = 1.193$, $K = 25$ iterations 2, 4 and 6 for $\omega = 0.4, 0.65$ and 0.9

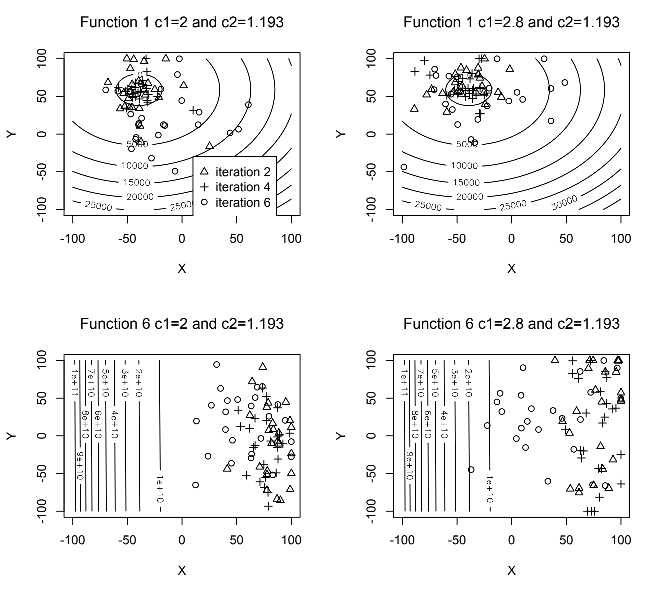

Fig. 3: Population for PSO with $\omega = 0.4$,$c_2 = 1.193$, $K = 25$ iterations 2, 4 and 6 for and $c_1 = 2$ and 2.8

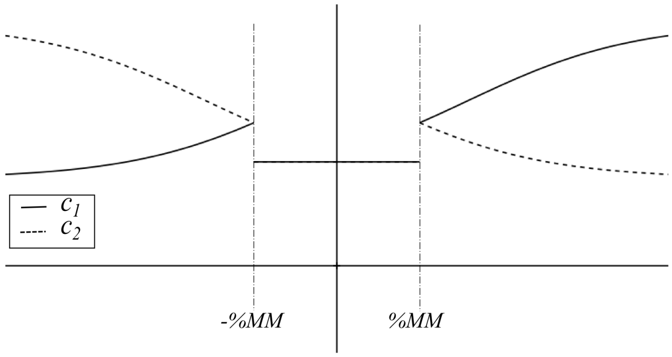

Fig. 4: Function to control c_1 and c_2

III. Results

A. Convergence test

The first implementation was tested using two function f_1 and f_3 for 10 dimensions. Convergence curves are shown in Fig 5. Dashed lines represent the behavior for SPSO2011 version, and solid lines are the performance for PRPSO for the same functions. For both functions algorithms converge, but PRPSO needs fewer iterations than SPSO2011 to achieve the same minimum.

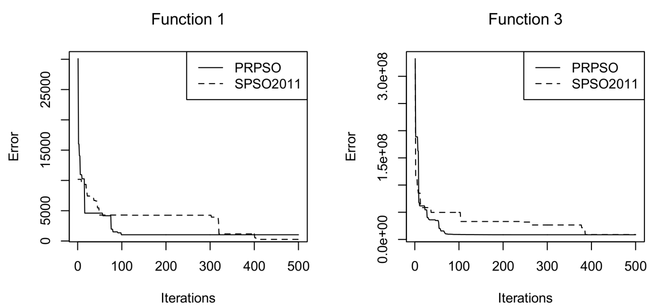

Fig. 5: Convergence curves for f_1 and f_3 for SPSO2011 and PR PSO

B. Benchmark's functions

We test the proposed strategy using CEC-2005 benchmark functions [22]. These 25 functions are divided as: the first 5 functions ($f_1 - f_5$) are unimodal, the next seven functions are basic multimodal functions ($f_6 - f_{12}$), the following two functions (f_{13} and f_{14}) are expanded functions. The remaining functions are hybrid functions ($f_{15} - f_{25}$). This test is characterized by the high level of complexity, all functions are modify to be asymmetrical and translate to have minimums different than zero.

C. Experimental design

We validate the proposed self-adaptive PSO version comparing with other two versions of PSO. Long SentenceThe first one is the original version of PSO [1]. The results for this classical version were presented by Derrac *et al.* in [23]. The second algorithm is the standard version SPSO2011 which corresponds to the last version proposed [20], [19]. In this case, we ran the test using R language according to version available in the free access code [24]. The parameters for each version are summarized in Table II. Other adaptive PSO versions have not been public code and are tested with other benchmark functions and other conditions, which are not comparable.

TABLE II: Parameters used to run the different version of PSO

parameter	PSO	SPSO2011	PRPSO
c_1	2.8	1.193	Self-adative Eq. 8
c_2	1.3	1.193	Self-adative Eq. 8
ω	$\omega = 0.9 - (0.5)\frac{t}{T}$	0.7213	Self-adative Eq. 7
population size	100	40	40

For each function, the three algorithms run 50 times. The optimizers stop when 1000 iterations was achieved or the error found was equal or less than 1×10^{-8}. Table III presents the average of the minimum found for the 50 independent runs of each 25 functions.

D. Discussion

In Table III, The columns present each algorithm compared. The rows show the functions results for each function which corresponds to the average of the best minimum found for each 50 runs.The bold letter corresponds to the minimal values of each function. In the last row of the table, the Friedman's test rank is presented.

We compare the results posing a set of hypothesis for Friedman's test. Thereby, we can determine if there is a significant difference in the performance of the algorithms [23]. The null hypothesis (H_0) correspond to non-significant difference between algorithms' behavior. Whereas, alternative hypothesis (H_a) is associated with a significant difference between the three algorithms. To calculate the Friedman's test statistic we use Iman and Davenport approximation described in [23] and the equation is:

$$F_F = \frac{12n}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right] \quad (9)$$

and

$$F_{ID} = \frac{(n-1)F_F}{n(k-1) - F_F} \quad (10)$$

where n is the number of functions, k the number of algorithms and R_j is the average rank of each algorithm. The calculated value for this rank is presented in last row of the Table III. Using the F distribution with $(k-1)$ and $(k-1)(n-1)$ degree of freedom we find a $p - value = 3.98 \times 10^{-7}$. According to the p-value, we reject H_0 with high level of probability, i.e. there is a significant difference between the performance of the algorithms.

With the Friedman's test, we can conclude that there is a significant difference between the algorithms. Now we

TABLE III: AVERAGE ERROR FOR DIFFERENT VERSIONS OF PSO USING CEC 2005 BENCHMARK FUNCTIONS SET AND RANKING FOR FRIEDMAN'S TEST

Function	PSO	SPSO2011	PRPSO
f_1	1.230×10^{-04}	$\mathbf{1.000 \times 10^{-08}}$	$\mathbf{1.000 \times 10^{-08}}$
f_2	2.600×10^{-02}	$\mathbf{1.000 \times 10^{-08}}$	1.284×10^{-07}
f_3	$5.170 \times 10^{+04}$	$6.275 \times 10^{+04}$	$\mathbf{1.563 \times 10^{+04}}$
f_4	$2.490 \times 10^{+00}$	$\mathbf{1.000 \times 10^{-08}}$	4.288×10^{-05}
f_5	$4.100 \times 10^{+02}$	$4.669 \times 10^{+01}$	$\mathbf{2.046 \times 10^{+01}}$
f_6	$7.310 \times 10^{+02}$	$\mathbf{1.357 \times 10^{+02}}$	$1.389 \times 10^{+02}$
f_7	$2.680 \times 10^{+01}$	$1.209 \times 10^{+03}$	$\mathbf{1.167 \times 10^{+03}}$
f_8	$2.040 \times 10^{+01}$	$\mathbf{1.931 \times 10^{+01}}$	$1.937 \times 10^{+01}$
f_9	$1.440 \times 10^{+01}$	$6.283 \times 10^{+00}$	$\mathbf{5.972 \times 10^{+00}}$
f_{10}	$1.400 \times 10^{+01}$	$5.277 \times 10^{+00}$	$\mathbf{4.987 \times 10^{+00}}$
f_{11}	$5.590 \times 10^{+00}$	$\mathbf{4.861 \times 10^{+00}}$	$5.592 \times 10^{+00}$
f_{12}	$6.360 \times 10^{+02}$	$2.110 \times 10^{+02}$	$\mathbf{5.381 \times 10^{+01}}$
f_{13}	$1.500 \times 10^{+00}$	$1.093 \times 10^{+00}$	$\mathbf{1.021 \times 10^{+00}}$
f_{14}	$3.300 \times 10^{+00}$	$\mathbf{2.711 \times 10^{+00}}$	$3.027 \times 10^{+00}$
f_{15}	$\mathbf{3.400 \times 10^{+02}}$	$4.000 \times 10^{+02}$	$3.566 \times 10^{+02}$
f_{16}	$1.330 \times 10^{+02}$	$\mathbf{1.001 \times 10^{+02}}$	$1.065 \times 10^{+02}$
f_{17}	$1.500 \times 10^{+02}$	$\mathbf{1.219 \times 10^{+02}}$	$1.262 \times 10^{+02}$
f_{18}	$8.510 \times 10^{+02}$	$8.000 \times 10^{+02}$	$\mathbf{3.001 \times 10^{+02}}$
f_{19}	$8.500 \times 10^{+02}$	$8.000 \times 10^{+02}$	$\mathbf{3.000 \times 10^{+02}}$
f_{20}	$8.510 \times 10^{+02}$	$8.000 \times 10^{+02}$	$\mathbf{3.003 \times 10^{+02}}$
f_{21}	$9.140 \times 10^{+02}$	$8.000 \times 10^{+02}$	$\mathbf{5.000 \times 10^{+02}}$
f_{22}	$8.070 \times 10^{+02}$	$7.541 \times 10^{+02}$	$\mathbf{7.438 \times 10^{+02}}$
f_{23}	$1.030 \times 10^{+03}$	$9.705 \times 10^{+02}$	$\mathbf{5.495 \times 10^{+02}}$
f_{24}	$4.120 \times 10^{+02}$	$2.048 \times 10^{+02}$	$\mathbf{2.000 \times 10^{+02}}$
f_{25}	$5.100 \times 10^{+02}$	$2.381 \times 10^{+02}$	$\mathbf{2.002 \times 10^{+02}}$
R_j	2.76	1.78	1.46

use Wilcoxon's test, a pairwise comparison, to determine which one is better. The set of hypothesis posed are: null hypothesis the performances is non-significant different and the alternative hypothesis there is a significant difference between both algorithms. In this case, we use one-tailed test to guarantee that the performance is better. In Table IV, we present the ranks calculated from Table III. The negative ranks show a better performance for PRPSO. The absolute value of the smallest rank between positive and negative ranks should be equal or less than the critical value of the last column ([25] Table A.5) to be significant different. In both cases, the difference between algorithms is significant and the negative rank is greater. Therefore, PRPSO performance is significant better than the other two algorithms.

TABLE IV: RANKING FOR WILCOXON'S TEST

Algorithms	Positive Rank	Negative Rank	Critical Value $\alpha = 0.05$
PRPSO-PSO	67	-258	100
PRPSO-SPSO2011	79,5	-220,5	91

Besides, functions $(f_6 - f_{25})$ are multimodal. One of the failures described in literature for SPSO2011 is multimodal functions[19]. For hybrid and rotated functions $(f_{15} - f_{25})$ PRPSO shows better average for the most part of these functions.

IV. CONCLUSIONS

PSO and SPSO2011 are well established techniques for optimizing nonlinear complex functions; however, some problems appear when functions have not a well-defined descendent direction. The first contribution of this work is applying a criterion for controlling the parameters according to the characteristics of the topology of the optimized function; the criterion allows us to define when the population is placed in a region that needs more exploration with the aim to find the optimum. The second contribution is the proposal of specific equations for the self-adaptation of the three parameters of the PSO algorithm based on the previous criterion. The proposed version of self-adaptive PSO is called PRPSO. The main strategy of adaptive parameters is the increment of exploration for objective functions that have planar regions. The performance of PRPSO was validated comparing it with another two version, original PSO and SPSO2011. PRPSO is, in statistical terms, significant better than the other two versions.

In the future, we expect to improve the control function testing other kinds of functions different to the sigmoidal function or changing the parameters of the current function for increasing the level of exploration according to the number of iterations. Moreover, a new parameter for the sigmoidal function can introduce a smoothly increasing of the exploration when function presents low-quality regions(planar regions) repeatedly. Futhermore, we expect to control the size of population to reduce the computational effort. Finally, The proposed strategy should extend to another metaheuristics.

REFERENCES

[1] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in Proc. IEEE Int. Conf. Neural Netw., Perth, Australia, vol. 4, pp. 1942-1948, 1995.

[2] P. Angeline, "Adaptative and Self-Adaptative Evolutionary Computations," in *IEEE, Computational Intelligence*, pp. 152-163, 1995.

[3] O. Kramer, *Self-adaptative Heuristics for Evolutionary Computation*, First Ed. Berlin: springer, p. 175, 2008.

[4] D. Sedighizadeh and E. Masehian, "Particle Swarm Optimization Methods, Taxonomy and Applications," *International Journal of Computer Theory and Engineering*, Vol. 1, No. 5., 2009,pp 486-502.

[5] X.-M.Wang, Y.-Z. Guo and G.-J. Liu, "Self-adaptive particle swarm optimization algorithm with mutation operation based on K-Means" *Advanced Materials Research*, 760-762, pp. 2194-2198,2013.

[6] E. G. Talbi, *Metaheuristics: from design to implementation*. Wiley, 2009.

[7] T. Back, *Evolutionary Algorithms in Theory and Practice*. Oxford University Press,1996.

[8] A.Torn and Zilinkas, "Global Optimization, in *Lecture Notes in Computer Science* vol. 350, G. Goos and J. Hartmanis, Eds. Berlin:Springer, 1989, p. 255.

[9] I.B. Mohd, "Identification of region of attraction for global optimization problem using interval symmetric operator," in*Applied Mathematics and Computation* 110 (2000) pp.121-131.

[10] Z. H. Zhan, J. Zhang, Y. Li and H.S. Chung, "Adaptive Particle Swarm Optimization," in *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* , vol.39, no.6, 2009, pp.1362-1381.

[11] Z. Hou, "Hammerstein Model Identification Based on Adaptive Particle Swarm Optimization," in *Intelligent Information Technology Application, Workshop on* , Washington, 2007,pp.137-140.

[12] J. Yisu, J. Knowles, L. Hongmei, L. Yizeng and D. B. Kell, "The landscape adaptive particle swarm optimizer," *Applied Soft Computing*, Vol. 8, No. 1., 2008,pp 295-304

[13] E. Mesa, J. D. Velasquez and G. P. Jaramillo, "Nonlinear Optimization in Landscapes with Planar Regions" in *Studies in Computational Intelligence*, Vol. 512,*Nature Inspired Cooperative Strategies for Optimization (NICSO 2013)*, G. Terrazas, F. E.B. Otero and A.D. Masegosa, Eds, Switzerland:Springer International Publishing,2014, pp 203-215.

[14] M. Clerc, *Particle swarm optimization*, ISTE,2006
[15] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," in *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, 2006 pp. 281295.
[16] W.-J. Zhang; X.-F. Xie, "DEPSO: hybrid particle swarm with differential evolution operator," *Systems, Man and Cybernetics, 2003. IEEE International Conference on* , vol.4, Washington, 2003 pp.3816-3821.
[17] T.Huang, A. S. Mohan "Micro-particle swarm optimizer for solving high dimensional optimization problems (μ-PSO for high dimensional optimization problems)"
[18] M. Clerc, "Back to random topology, Open access archive, http://clerc.maurice. free.fr/pso/random topology.pdf.
[19] M. Zambrano, M. Clerc and R. Rojas, "Standard Particle Swarm Optimisation 2011 at CEC-2013: A baseline for future PSO improvements. in *Evolutionary Computation (CEC), 2013 IEEE Congress on* , Cancún, 2013 pp.2337-2344.
[20] M. clerc. "Standard particle swarm optimization ",Open access archive HAL: http://hal.archives-ouvertes.fr/hal-00764996, 2012.
[21] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," *in Proc. IEEE World Congr. Comput. Intell.*, 1998, pp. 6973.
[22] P.N. Suganthan, et al., Problem de definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. Technical Report KanGAL N2005005, Nanyang Technological University, Singapure (May 2005)
[23] J. Derrac, S. Garca, D. Molina, F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," in *Swarm and Evolutionary Computation*, Vol. 1,2011, pp 3-18.
[24] M. Clerc, "SPSO-2011 code for matlab",Available in http://clerc.maurice.free.fr/pso/
[25] D. Sheskin,*Handbook of parametric and nonparmetric statistical procedures*, 3rd ed. Chapman and Hall,2004, p.1187