Advances in Electrical and Computer Engineering **Volume 15, Number 2, 2015**

Incorporating the Avoidance Behavior to the Standard Particle Swarm Optimization 2011

Okkes Tolga ALTINOZ[1], Asim Egemen YILMAZ[1], Anton DUCA[2], Gabriela CIUPRINA[2]
[1]*Department of Electrical and Electronics Engineering, Ankara University, Turkey*
[2]*Department of Electrical Engineering, Politehnica University of Bucharest, Romania*
taltinoz@ankara.edu.tr, aeyilmaz@eng.ankara.edu.tr, anton.duca@upb.ro, gabriela@lmn.pub.ro

***Abstract*—Inspired from social and cognitive behaviors of animals living as swarms; particle swarm optimization (PSO) provides a simple but very powerful tool for researchers who are dealing with collective intelligence. The algorithm depends on modeling the very basic random behavior (i.e. exploration capability) of individuals in addition to their tendency to revisit positions of good memories (cognitive behavior) and tendency to keep an eye on and follow the majority of swarm members (social behavior). The balance among these three major behaviors is the key of success of the algorithm. On the other hand, there are other social and cognitive phenomena, which might be useful for improvement of the algorithm. In this paper, we particularly investigate "avoidance from the bad" behavior. We propose modifications about modeling the Standard PSO 2011 formulation, and we test performance of our proposals at each step via benchmark functions, and compare the results of the proposed algorithms with well-known algorithms. Our results show that incorporation of "Social Avoidance" behavior into SPSO11 improves the performance. It is also shown that in case the Social Avoidance behavior is applied in an adaptive manner at the very first iterations of the algorithm, there might be further improvements.**

***Index Terms*—particle swarm optimization, social factors, cognitive informatics, performance evaluation.**

I. INTRODUCTION

Swarm intelligence constitutes a very significant portion of the literature regarding nature inspired methods. The term "swarm intelligence", since its introduction by Beni and Wang [1] in 1989 with the context of cellular robotic systems, has been a major multidisciplinary attraction center for researchers dealing especially with complex inverse (e.g. design and synthesis) problems. Typically, swarm intelligence systems consist of a population with members having some characteristic behaviors and interacting locally with each other within their environment. In these systems, the members individually behave freely to a certain extent and interact with each other. Even though there is no dictating centralized mechanism, these interactions yield a global behavior, which is more organized and directive than that of a stand-alone individual.

As of today, many nature inspired optimization algorithms (like Ant Colony Optimization, Artifical Bee Optimization, Cuckoo Search, Firefly Search etc.) have been proposed. Among them, two main algorithms come into mind first, when the phrase "swarm intelligence" is mentioned. These are the Ant Colony Optimization (ACO) (considered initially in 1992 by Dorigo in his Ph.D. Thesis [2], and later formalized by Dorigo et al. in [3]), and the Particle Swarm Optimization (PSO) (developed by Kennedy and Eberhart in 1995 [4]) methods. Both algorithms were developed by observing the behaviors of animals living as swarms/colonies and getting inspired by them; and in more than a decade, they proved to be successful in solving various complex problems due to their intelligent and systematic metaheuristic approaches. Originally, ACO was designed for combinatorial optimization problems; whereas PSO was designed for continuous ones. However, by the time, successful versions of continuous ACO (e.g. [5-7]), and discrete PSO (e.g. [8-10]) have been developed.

Between these two methods, since it is originally defined and quite appropriate for continuous problems, the latter constitutes the main subject of our interest and this paper. Similar to the members of the swarms individually searching for the best place for nutrition in 3-dimensional space, the original PSO algorithm depends on motions of particles (swarm members) searching for the global best in *N*-dimensional continuous space. In addition to this exploration capability (i.e. the tendency for random search throughout the domain), each particle has a cognitive behavior (i.e. remembering its own good memories and having tendency to return there); as well as a social behavior (i.e. observing the rest of the swarm and having tendency to go where most other particles go). A pictorial description (based on Robinson and Rahmat-Samii [11]) of the basic idea behind the algorithm in 2-*D* (*N*=2) is given in Fig. 1.

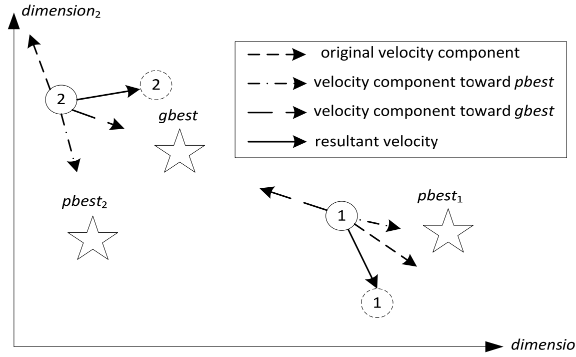
dimension₂
dimension₁
original velocity component
velocity component toward pbest
velocity component toward gbest
resultant velocity
gbest
pbest₂
pbest₁
2
2
1
1

Figure 1. 2D pictorial description of the PSO method (based on the illustration by Robinson and Rahmat-Samii [11]).

Mathematical expression of these behaviors will be presented in the upcoming paragraphs. But first, let us introduce our nomenclature and notation. For a variable *a*, in the representation $a_i^{(d)}[k]$:

- the subscript *i* denotes the particle index (i.e. the symbol

---

This work was supported in part by a joint grant from TUBITAK (with Grant Nr. 112E168) and ANCS-UEFISCDI (with Grant Nr. 605/01.01.2013)..

Digital Object Identifier 10.4316/AECE.2015.02007

i addresses the *i*th particle),

- the superscript with parenthesis denotes the dimension index (i.e. the symbol (*d*) addresses the *d*th dimension),
- and the term inside the bracket denotes the iteration index (i.e. the symbol [*k*] addresses the *k*th iteration).

Hence, for a *D*-dimensional problem, we denote position of the *i*th particle at *k*th iteration as $x_i[k]=\{x_i^{(1)}[k], x_i^{(2)}[k],..., x_i^{(D)}[k]\}$, and eventually $x_i^{(d)}[k]$ is the position of that particle in the *d*th dimension at *k*th iteration. Similarly, the velocity vector of the *i*th particle at *k*th iteration can be expressed as, where $v_i^{(d)}[k]$ is the corresponding velocity component in the *d*th dimension at *k*th iteration. By these very definitions, in conventional PSO, the velocity and position update of each particle in each dimension at each iteration can be expressed as:

$$\begin{aligned} v_i^{(d)}[k+1] = {} & w \cdot v_i^{(d)}[k] \\ & + c_1 \cdot \text{rand}() \cdot \left(p_{best,i}^{(d)}[k] - x_i^{(d)}[k]\right) \\ & + c_2 \cdot \text{rand}() \cdot \left(g_{best}^{(d)}[k] - x_i^{(d)}[k]\right) \end{aligned} \tag{1}$$

$$x_i^{(d)}[k+1] = x_i^{(d)}[k] + v_i^{(d)}[k+1] \cdot \Delta \tag{2}$$

where:

- $x_i^{(d)}[k]$ is the position of the *i*th particle in the *d*th dimension at the *k*th iteration,
- $v_i^{(d)}[k]$ is the velocity component of the *i*th particle in the *d*th dimension at the *k*th iteration,
- $p_{best,i}^{(d)}[k]$ is the best position experienced by the *i*th particle in the *d*th dimension up to the *k*th iteration (where the letter *p* is for personal best),
- $g_{best}^{(d)}[k]$ is the best position experienced by the whole swarm in the *d*th dimension up to the *k*th iteration (where the letter *g* is for global best),
- w is the so-called inertial weight factor, which prevents the particles to get excessively-speedy (usually initialized at 0.9 and decreased throughout the iterations down to 0.4),
- c_1 is a measure of how much the particle has tendency to revisit the positions in its personal good memories (used for representing the cognitive behavior and usually chosen to be 1.494),
- c_2 is a measure of how much the particle keeps an eye to the other swarm members and has tendency to go there (used for representing the social behavior and usually chosen to be 1.494),
- rand() is a pseudo-random number generated from a uniform distribution in the [0,1] interval,
- Δ is the time step or time difference between two successive iterations (usually assumed to be unity for simplicity, and will be omitted throughout the formulations from now on)

The parametric representation of all these tendencies and the balance among them are the keys for the success and the power of the method. Therefore, PSO has been successfully applied to various multidimensional continuous and discontinuous problems, so far. A review article by Poli [12] demonstrates how wide the application spectrum of the method currently is.

PSO depends on the behaviors of creatures living in forms of swarms. The main cognitive and social behaviors of these swarm members have already been modeled and successfully incorporated into the algorithm. The main theme of our ongoing research is investigation of the effects of more complicated social phenomena (such as avoidance from the bad, social exclusion, etc.) to the success of the algorithm (in terms of finding the global optimum, convergence rate, etc.). In this paper, we will try to investigate how to incorporate the "avoidance from the bad" type of behavior to the Standard PSO 2011 [13] (will be referred to as SPSO11 from now on) and observe the impacts on the performance of the algorithm in a quantitative manner.

II. RELATED WORKS

"Avoidance from the bad" is a common type of behavior for all creatures. Even though the term "bad" has a very wide scope and meaning in different aspects (e.g. the predator for an animal, an unethical activity, group or individual for a human being, etc.), it can be simply modeled just as the antonym term "good" which has already been modeled and incorporated in the original PSO formulation

Modeling and incorporation of the "avoidance from the bad" behavior was previously been proposed by Yang and Simon [14]. Yang and Simon modified the velocity update of the traditional inertial weight PSO (i.e. Eq. (1)) formulation as follows:

$$\begin{aligned} v_i^{(d)}[k+1] = {} & w.v_i^{(d)}[k] \\ & + c'_1.\text{rand}().\left(x_i^{(d)}[k] - p_{worst,i}^{(d)}[k]\right) \\ & + c'_2.\text{rand}().\left(x_i^{(d)}[k] - g_{worst}^{(d)}[k]\right) \end{aligned} \tag{3}$$

where:

- $p_{worst,i}^{(d)}[k]$ is the worst position experienced by the *i*th particle in the *d*th dimension up to the *k*th iteration,
- $g_{worst}^{(d)}[k]$ is the worst position experienced by the whole swarm in the *d*th dimension up to the *k*th iteration,
- c'_1 is a measure of how much the particle has tendency to avoid the positions in its personal bad memories (not to be confused with c_1 of Eq. (1) since it serves in the contrary manner),
- c'_2 is a measure of how much the particle keeps an eye to the bad positions of other swarm members and has tendency to avoid there (not to be confused with c_1 of Eq. (1) since it serves in the contrary manner)

The idea is pictorially depicted in Fig. 2. As can be seen from Eq. (3) and Fig. 2, the proposal of Yang and Simon does not include any attraction to the good memories or positions; it only considers avoidance from the bad memories or positions. As mentioned by the authors themselves, the experiments in that study were limited to a very narrow setting, and they could not have performed a comprehensive or definitive concluding discussion. Their proposal did not result in an explicit improvement of the performance of PSO. However, to our belief, the conceptual proof of this idea (i.e. the idea of "avoidance from the bad") requires more research. That constitutes the main motivation of our study, which can be considered as a highly empirical study based on quantification of the performances of the proposed formulations evaluated via numerous independent executions for benchmark functions.

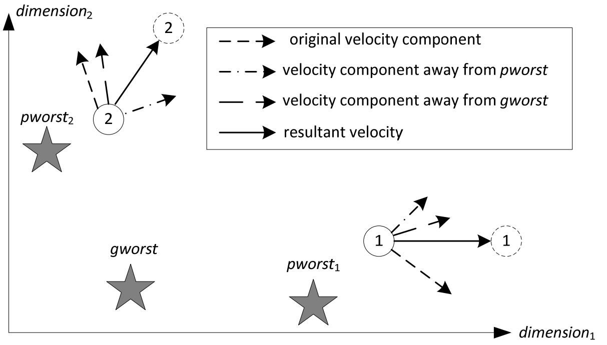
dimension₂
dimension₁
original velocity component
velocity component away from pworst
velocity component away from gworst
resultant velocity
pworst₂
gworst
pworst₁
1
2

Figure 2. 2D pictorial description of the proposal of Yang and Simon [16].

The idea of "avoidance from the bad" was also mentioned previously by Ciuprina et al. [15] in the so-called Intelligent Particle Swarm Optimization (IPSO) algorithm [16]; where a tabu list is constructed via bad experiences. But since IPSO is an advanced algorithm with many other intelligent features, it is not possible to observe the stand-alone impact of the "tabu list" idea explicitly.

Biswas et al. [17] proposed a similar idea, but in their proposal:

- They introduced only cognitive avoidance (but not social avoidance),
- They applied the idea to the conventional inertial weight PSO; and
- Their analyses were limited to only 4 conventional benchmark functions with aforementioned disadvantages.

The remainder of this paper is organized as follows: In the next section, we will revisit the recently proposed and standardized PSO formulation, called SPSO11 [13]. In Section 3, will include modeling the "Avoidance from the Bad" behavior. Proposed formulation will be presented in this section we will present benchmark functions, which are used for performance evaluation of optimization algorithms. In this study, comparisons between SPSO11 and modified SPSO11 variants, which are our proposals, are based on these benchmark functions given in Section 4, and experimental results will be commented. The last section will be the conclusions of the study

III. STANDARD PARTICLE SWARM OPTIMIZATION 2011

SPSO11 was proposed, and compared with other variants of Standard PSO by Clerc [13]. SPSO11 contains generally accepted improvements such as wall reflection, however it is still modifiable. Those features make this algorithm the best preference for our study. In this section, this algorithm is explained in four steps.

Step 1: Initialization

The first step naturally is the initialization phase. As in other PSO variants, initial position of each particle is assigned randomly via uniformly distributed numbers having intervals coinciding with the borders of the solution space (Eq. (4)),

$$x_i^{(d)}[0] = \varphi(X_{\min}^{(d)}, X_{\max}^{(d)}) \tag{4}$$

where lower and upper bounds are $X_{min}= \{X_{min}^{(1)}, X_{min}^{(2)},..., X_{min}^{(D)}\}$ and $X_{max}=\{X_{max}^{(1)}, X_{max}^{(2)},..., X_{max}^{(D)}\}$ and $\varphi()$ denotes the pseudo-random number generation function. However, unlike its predecessors, SPSO11 has a different velocity initialization approach. In previous methods, particle's velocities are initialized in a similar manner with their positions. Such an assignment might cause most particles to leave the search space immediately after the position update, especially when the problem dimension is high. Hence, is SPSO11 the initial particle velocities are selected from a narrowed space (which is constructed via subtraction of the initial position of every particle from the search space boundary), the updated position hardly exceeds the search space boundaries as [13]:

$$v_i^{(d)}[0] = \varphi\left(\left(X_{\min}^{(d)} - x_i^{(d)}[0]\right), \left(X_{\max}^{(d)} - x_i^{(d)}[0]\right)\right) \tag{5}$$

In SPSO11, two vectors are defined for determining the previous personal best position ($p_{best,i}[0]$), and the previous best position in the neighborhood ($l_{best,i}[0]$). These two vectors are initially defined in a similar manner as follows:

$$p_{best,i}[0] = x_i\,[0],\ \text{ i.e. for } \forall d = 1,...,D : p_{best,i}^{(d)}[0] = x_i^{(d)}[0] \tag{6}$$

Step 2: Topology Construction

SPSO11 prefers "local neighborhood best" approach rather than the "global best" approach. This means that, instead of information sharing among all particles (same link among each particle); only the neighbors have the information about their fellow particles. SPSO11 applies "adaptive random topology" as an information link. Before each iteration, *N*x*N* matrix is created as an indicator of the information link. The column and row are the swarm indices, and they indicate which particle is linked to other particles. However, these links are not static; they are modified at the beginning of each iteration in case there is no improvement on the best fitness value. In other words, the matrix is modified randomly based on the probability ρ given in Eq. (7), where N is the swarm size and L is the link size.

$$\rho = 1 - \left(\frac{N-1}{N}\right)^L \tag{7}$$

Step 3: Velocity and Position Updates

At each iteration, both velocity and position of each particle are updated. Traditional velocity update rule seen in Eq. (1) causes biases; which means that this rule is efficient only for problems whose optima located close to the center of the search space [14]. In SPSO11, this rule is modified in such a manner that the velocity update equation does not depend on system coordinates.

In SPSO11, the velocity update is based on the center of gravity (G_i) obtained via three positions, which are:

- The current position: $x_i^{(d)}[k]$
- A position between the current position and relative best previous position: $z_i^{(d)}[k]$, as seen in Eq. (8):

$$z_i^{(d)}[k] = x_i^{(d)}[k] + c_1\left(p_{best,i}^{(d)}[k] - x_i^{(d)}[k]\right) \tag{8}$$

- And a position between the current position and relative best previous position in the neighborhood: $q_i^{(d)}[k]$, as seen in Eq. (9):

$$q_i^{(d)}[k] = x_i^{(d)}[k] + c_2\left(l_{best,i}^{(d)}[k] - x_i^{(d)}[k]\right) \tag{9}$$

For the *i*th particle, the position of the Center of Gravity in the *d*th dimension at the (*k*+1)'st iteration is calculated as in Eq. (10) where the cognitive and social behavior factors are chosen to be identical (i.e. $c_1=c_2=c$):

$$G_i^{(d)}[k+1] = \frac{3x_i^{(d)}[k] + c\left(l_{best,i}^{(d)}[k] + p_{best,i}^{(d)}[k] - 2x_i^{(d)}[k]\right)}{3} \tag{10}$$

Definition of the center of the gravity changes, in case the best previous position $p_{best,i}{}^{(d)}[k]$ equals to best previous position in the neighborhood $l_{best,i}{}^{(d)}[k]$. In this case, Eq. (10) is substituted with Eq. (11).

$$G_i^{(d)}[k+1] = \frac{2x_i^{(d)}[k] + c\left(p_{best,i}^{(d)}[k] - x_i^{(d)}[k]\right)}{2} \tag{11}$$

The new velocity is determined by using a randomly selected arbitrary point ($\tilde{x}$) within the circle having a radius r equal to the difference between the current position and the center of the gravity. By using this point, the new velocity is obtained as in Eq. (12); and the position is updated according to Eq. (13).

$$v_i^{(d)}[k+1] = wv_i^{(d)}[k] + \left(\tilde{x}_i^{(d)}[k] - x_i^{(d)}[k]\right) \tag{12}$$

$$x_i^{(d)}[k+1] = wv_i^{(d)}[k] + \tilde{x}_i^{(d)}[k] \tag{13}$$

Step 4: Boundary Reflection

Optimization algorithms aim to determine the optimum solution for the optimization problem inside the borders of the search space. The solutions outside the search space are meaningless. Therefore, SPSO11 also has the motivation to keep the particles inside the search space. To ensure this, the borders of the search space are treated as physical walls, as defined in Eq. (14).

$$\begin{gathered}\text{if } x_i^{(d)}[k+1] < X_{\min}^{(d)}, \text{ then} \begin{cases} x_i^{(d)}[k+1] = X_{\min}^{(d)} \\ v_i^{(d)}[k+1] = -v_i^{(d)}[k+1] \end{cases} \\ \text{if } x_i^{(d)}[k+1] > X_{\max}^{(d)}, \text{ then} \begin{cases} x_i^{(d)}[k+1] = X_{\max}^{(d)} \\ v_i^{(d)}[k+1] = -v_i^{(d)}[k+1] \end{cases}\end{gathered} \tag{14}$$

In this study, we first come up with the proposal approach and the proposal of Yang and Simon [17]. In other words, the particles will be attracted toward the best positions and memories; meanwhile they will avoid the bad ones as depicted in Fig. 3.

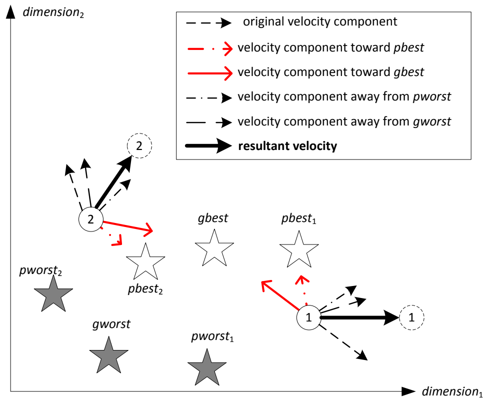
dimension₂
original velocity component
velocity component toward pbest
velocity component toward gbest
velocity component away from pworst
velocity component away from gworst
resultant velocity
2
2
gbest
pbest₁
pworst₂
pbest₂
gworst
pworst₁
1
1
dimension₁

Figure 3. 2D pictorial description of our proposal.

IV. PROPOSED APPROACH: THE "AVOIDANCE FROM THE BAD" BEHAVIOR

The method "Avoidance from the bad" is actually incorporated to SPSO11 via expanding the relevant equations with new factors and components seen in Eq.s (15) and (16):

$$c_3.\left(x_i^{(d)}[k] - p_{worst,i}^{(d)}[k]\right) \tag{15}$$

$$c_4.\left(x_i^{(d)}[k] - l_{worst,i}^{(d)}[k]\right) \tag{16}$$

- $p_{worst,i}{}^{(d)}[k]$ is the worst position experienced by *i*th particle in the *d*th dimension up to the *k*th iteration,
- $l_{worst,i}{}^{(d)}[k]$ is the worst position experienced by the whole particles in the local neighborhood of the *i*th particle in the *d*th dimension up to the *k*th iteration.

In this study, it is desired to investigate the effect of the movement of particles not only towards to the best particle positions but also tendency to moves away from the worst positions. By this way, it is expected that the particles much faster and accurately detects global optima. At this point, we propose three different alternatives by adding the factors in Eq.s (15) and (16) to Eq.s (7) and (8), respectively:

- Case 1: Cognitive Avoidance (will be referred to as CA from now on); which corresponds to avoidance from personal bad experiences only. By this way it is expected that new particle selected in a distinct region from particles worst position. Defined by Eq. (17):

$$\text{CA:}\begin{cases} z_i^{(d)}[k] = x_i^{(d)}[k] + c_1.\left(p_{best,i}^{(d)}[k] - x_i^{(d)}[k]\right) \\ \qquad + c_3.\left(x_i^{(d)}[k] - p_{worst,i}^{(d)}[k]\right) \\ q_i^{(d)}[k] = x_i^{(d)}[k] + c_2.\left(l_{best,i}^{(d)}[k] - x_i^{(d)}[k]\right) \end{cases} \tag{17}$$

- Case 2: Social Avoidance (will be referred to as SA from now on); which corresponds to avoidance from local neighbor's bad experiences only. Since SPSO11 uses local best instead of global best, in this formulation also local worst is considered. Since the neighborhood is not static in the algorithm and neighbors are arranged in an adaptive matrix formation, the effect of moving away from worst is a hard concept to be visualized. The formulation is defined by Eq. (18):

$$\text{SA:}\begin{cases} z_i^{(d)}[k] = x_i^{(d)}[k] + c_1.\left(p_{best,i}^{(d)}[k] - x_i^{(d)}[k]\right) \\ q_i^{(d)}[k] = x_i^{(d)}[k] + c_2.\left(l_{best,i}^{(d)}[k] - x_i^{(d)}[k]\right) \\ \qquad + c_4.\left(x_i^{(d)}[k] - l_{worst,i}^{(d)}[k]\right) \end{cases} \tag{18}$$

- Case 3: Cognitive Social Avoidance (will be referred to as CSA from now on); which corresponds to avoidance from personal bad experiences as well as local neighbor's bad experiences. Since the particles moves all away from all worst positions, it is expected that new positions are selected from more narrow area any of the previous cases. However, this idea can increase the search ability by selecting positions closer to the best particles. The formulation is defined by Eq. (19):

$$\text{CSA:} \begin{cases} z_i^{(d)}[k] = x_i^{(d)}[k] + c_1.\left(p_{best,i}^{(d)}[k] - x_i^{(d)}[k]\right) \\ \qquad\qquad + c_3.\left(x_i^{(d)}[k] - p_{worst,i}^{(d)}[k]\right) \\ q_i^{(d)}[k] = x_i^{(d)}[k] + c_2.\left(l_{best,i}^{(d)}[k] - x_i^{(d)}[k]\right) \\ \qquad\qquad + c_4.\left(x_i^{(d)}[k] - l_{worst,i}^{(d)}[k]\right) \end{cases} \tag{19}$$

In summary, we investigate the relative performances alternatives such as Cognitive Avoidance Only, Social Avoidance Only, Cognitive and Social Avoidance, which will be referred to as CA, SA and CSA, respectively.

V. EXPERIMENTS AND RESULTS

Next, in this section, there will be four sub-sections to separate analyses for assessment of the quality of the proposed method, which are

A) Benchmark problems: Six composition benchmark problems are selected as test problems.

B) Comparison Results-1: In the first analysis, the aim is to show the efficiency of each proposed case. Therefore, they are compared with each other with respect to fitness value of the six benchmark problems. At the same time, these results are also compared with SPSO11 and with the best results obtained in [18], which contains a comparison among PSO, Cooperative PSO, Comprehensive PSO, Evolution strategy with covariance matrix adaptation, G3 model with PCX crossover and Differential Evolution. By this way, not only the best candidate among the three cases can be selected, but also efficiency of the proposed methods will be compared with their ancestor SPSO11 as well as other well-known methods presented in [18].

C) Wilcoxon Rank Sum Test: Results obtained from selected case and SPSO are compared via a statistical test, which is the Wilcoxon Rank Sum Test. This test presents a comparison between the means of two statistical distributions. By this way, we will be sure about the significance of the differences between mean values of the proposed case and SPSO11.

D) Comparison Results-2: In the second analysis, by taking account of the results obtained in the first analysis, the selected case is applied only for pre-determined number of iterations. The aim of this second analysis is to determine and discuss whether it is possible to obtain much better performance.

TABLE I. ALGORITHM PARAMETERS FOR SPSO11 AND OUR PROPOSAL.

$w = \dfrac{1}{2\log(2)}$ (Algorithm parameter)	$N = 20$ (Population Size)
$c_1 = c_3 = \dfrac{1}{2} + \log(2)$ (Algorithm parameter)	$M = 20$ (Number of independent Monte Carlo runs)
$c_2 = c_4 = \dfrac{1}{2} + \log(2)$ (Algorithm parameter)	$FE_{\max} = 5\text{x}10^4$ (Number of maximum function evaluation)

All algorithm parameters for SPSO11 and our proposals are selected as the same. On the other hand, only the parameters common with [15] (i.e. the population size, the number of independent Monte Carlo runs, and the number of maximum fitness evaluations) are kept identical with those given in [18]; other parameters cannot be same due to the differences among algorithm definitions. The parameters given in Table 1 seem to be sufficient for a fair and consistent comparison. Other unspecified parameters (if any) can be seen in [18].

A. Benchmark Problems

Benchmark functions are preferred for performance evaluation of the optimization algorithms. These functions are defined intentionally to have traps and challenges for optimization algorithms. However, conventional benchmark functions have some common features making them easy to be solved by using some pre-defined information or ideas. Generally, the global optima for most of them are at the same position for all dimensions, which are usually at the origin or the center of the search space, or on the boundaries; and their local optima are generally located on the coordinate axis. Such facts might sometimes behave like hints and clues for the optimization algorithms and ease their jobs. To get rid of these weaknesses, benchmark functions can be rotated, shifted, or rescaled randomly. Therefore, six benchmark problems which are defined by Liang et al. [18] their Matlab codes [18] are integrated and solved by using proposed algorithms.

The benchmark problems are based on five fundamental benchmark functions (Sphere, Rastrigin, Weierstrass, Griewank, and Ackley's functions); and they combined these functions in order to obtain more challenging functions. Therefore, six composition benchmark functions are obtained via Eq.s (20) and (21).

$$\begin{aligned} w_i &= \exp\left(-\frac{\sum_{k=1}^{D} (x_k - o_k)^2}{2D{\sigma_i}^2} \right) \\ w_i &= \begin{cases} w_i & w_i = \max(w_i) \\ w_i\left(1 - \left(\max(w_i)^{10}\right)\right) & w_i \neq \max(w_i) \end{cases} \\ w_i &= \frac{w_i}{\sum_{i=1}^{n} w_i} \end{aligned} \tag{20}$$

$$F(x) = \sum_{i=1}^{n} \left(w_i \left[f_i\left(\frac{x - o_i}{\lambda_i} M_i \right) + b_i \right] \right) + fb \tag{21}$$

The first elementary function $f_1(x)$ is always the function with the global optimum, as its bias is zero always. o_1, o_2, o_3,…,o_9 are all generated randomly in the search range, except o_{10} is set [0, 0, …, 0] for trapping algorithms which have a potential to converge to the center of the search range. M_1, M_2,...,M_n are DxD orthogonal rotation matrixes obtained by using Salomon's method. Detailed information about composition functions, and how can be formed discussed in [18].

B. Comparison Results - 1

In the first analysis, three blocks of results are considered in order comment on the performance of the proposed method:

TABLE II. COMPARISON BETWEEN SPSO11 AND PROPOSED METHODS.

		SPSO11	CA - Eq. (20)	SA - Eq. (21)	CSA - Eq. (22)
CF 1:	Mean	4.01e+000	1.08e+002	**4.89e-003**	1.83e+002
	Std.	1.79e+001	3.46e+001	3.20e-003	6.38e+001
CF2:	Mean	2.96e+001	2.16e+001	**8.61e+000**	4.98e+001
	Std.	5.08e+001	1.28e+001	2.66e+001	1.69e+001
CF3:	Mean	8.51e+001	2.78e+002	**6.98e+001**	3.86e+002
	Std.	5.04e+001	5.39e+001	4.54e+001	4.46e+001
CF4:	Mean	**2.03e+002**	3.59e+002	2.07e+002	4.45e+002
	Std.	1.04e+002	3.43e+001	9.45e+001	4.26e+001
CF5:	Mean	1.03e+001	1.25e+002	**5.35e-003**	1.63e+002
	Std.	3.18e+001	8.01e+001	2.87e-003	7.86e+001
CF6:	Mean	6.01e+002	5.64e+002	**5.21e+002**	5.35e+002
	Std.	2.05e+002	1.54e+002	1.97e+002	1.14e+002

TABLE III. COMPARISON RESULTS, WHERE BOLD ONES ARE THE BEST RESULT AMONG OTHER CONTRASTED METHODS.

		Best Result in [18]	SPSO11	CA - Eq. (17)	SA - Eq. (18)	CSA-Eq. (19)
CF 1:	Mean	**5.73e-008**	4.01e+000	1.08e+002	4.89e-003	1.83e+002
	Std.	1.03e-007	1.79e+001	3.46e+001	3.20e-003	6.38e+001
CF2:	Mean	1.91e+001	2.96e+001	2.16e+001	**8.61e+000**	4.98e+001
	Std.	1.47e+001	5.08e+001	1.28e+001	2.66e+001	1.69e+001
CF3:	Mean	1.32e+002	8.51e+001	2.78e+002	**6.98e+001**	3.86e+002
	Std.	2e+001	5.04e+001	5.39e+001	4.54e+001	4.46e+001
CF4:	Mean	3.14e+002	**2.03e+002**	3.59e+002	2.07e+002	4.45e+002
	Std.	2e+001	1.04e+002	3.43e+001	9.45e+001	4.26e+001
CF5:	Mean	5.37e+000	1.03e+001	1.25e+002	**5.35e-003**	1.63e+002
	Std.	2.6e+000	3.18e+001	8.01e+001	2.87e-003	7.86e+001
CF6:	Mean	**4.9e+002**	6.01e+002	5.64e+002	5.21e+002	5.35e+002
	Std.	3.94e+001	2.05e+002	1.54e+002	1.97e+002	1.14e+002

- We take the first block of results from Liang et al. [18]; and we select only the best result among Conventional Particle Swarm Optimization (PSO), Cooperative PSO (CPSO), Comprehensive Learning PSO (CLPSO), Evolution Strategy with Covariance Matrix Adaptation (CMA-ES), G3 model with PCX crossover (G3-PCX), Differential Evolution (DE).
- The second block of results is taken from SPSO11; and
- The last block is from our 3 proposed cases (CA, SA, and CSA).

The aim of this sub-section is to obtain the answer of the question "Which method should be chosen among all cases?" Therefore, proposed Eq.s (17) to (19) are applied separately, and relevant results are presented in three tables:

- The first table (Table 2) is aimed for outlining the difference among the proposed cases and their ancestor SPSO11. By this way, improvement on the algorithm, SPSO11, is clearly seen.
- In the second table (Table 3), all cases are compared to SPSO11 and the best results obtained in [18].
- In the last table (Table 4), contrast of the mean values of SPSO11 with the selected case is presented by using Wilcoxon Sum Rank test. Table 2 presents the first comparison only between SPSO11 and all proposed cases. Among all benchmark problems, only for composition benchmark function four (CF4) SPSO11 presents lower statistical results when compared to proposed cases. Even in that problem, the performance of SPSO and case SA can be regarded as similar since the difference in statistical results between SPSO11 and case SA is very small. The problem CF6 is the only test function that all proposed cases outperform SPSO11. However, for problems CF1, 3, and 5 cases CA and CSA (for CF2 only case CSA) present higher statistical performance, but still case SA shows better results than SPSO for these five test problems. It is clear that "SA" has better overall performance results for all benchmark functions except the composition benchmark function four. Table 3 presents comparison between all results including the best results from [18]. For two problems CF1 and CF2 the results from [18] outperform against cases and SPSO11. But for the other four problems CF2-5 show lower statistical problems for case SA except CF3 that also SPSO11 shows better result. The table indicates that, "CA" and "CSA" never obtain better results for the selected benchmark functions. The competition exists only for "SA". Hence, it is clear to consider only "SA" for the following analysis. For composition benchmark functions two, three, and five, "SA" presents improved results among all methods. Only the results obtained from [18] (i.e. Comprehensive Learning PSO for the first composition benchmark function and from Differential Evolution for the last composition function) are superior against SPSO11 and its variants. The results obtained from Table 2 and 3 indicate that only SA (defined via Eq. (18)) can be used to increase the performance of the SPSO11. Besides that, "SA" shows better performance against other well-known methods for composition benchmark functions two, three, and five. It should be noted that only 20 independent Monte Carlo runs are executed for this analysis. Therefore, in order to show the improvement in the mean values; a non-parametric statistical test, Wilcoxon Rank Sum Test will be executed in the next sub-chapter.

TABLE IV. APPLICATION OF THE PROPOSED METHOD FOR LIMITED NUMBER OF ITERATIONS.

		First 10 iterations	First 10^2 iterations	First 10^3 iterations	First 10^4 iterations	All iterations
CF 1:	Mean	4.95e-003	4.00e+000	4.00e+000	8.00e+000	**4.89e-003**
	Std.	3.11e-003	1.79e+001	1.79e+001	2.46e+001	3.20e-003
CF2:	Mean	1.36e+001	1.71e+001	1.36e+001	9.61e+000	**8.61e+000**
	Std.	3.32e+001	3.52e+001	4.42e+001	2.96e+001	2.66e+001
CF3:	Mean	7.46e+001	**4.50e+001**	6.24e+001	5.73e+001	6.98e+001
	Std.	4.03e+001	4.24e+001	5.28e+001	4.43e+001	4.54e+001
CF4:	Mean	2.03e+002	2.02e+002	**1.62e+002**	2.11e+002	2.07e+002
	Std.	1.04e+002	1.08e+002	3.16e+001	9.98e+001	9.45e+001
CF5:	Mean	5.68e-003	2.08e+001	**4.29e-003**	3.50e-003	5.35e-003
	Std.	2.35e-003	4.27e+001	3.08e-003	3.00e-003	2.87e-003
CF6:	Mean	6.37e+002	6.77e+002	**5.21e+002**	5.56e+002	**5.21e+002**
	Std.	1.94e+002	1.77e+002	1.97e+002	2.01e+002	1.97e+002

C. *Wilcoxon Rank Sum Test*

Wilcoxon Rank Sum Test [19] is a non-parametric statistical test that can used to compare the means of the two independent continuous populations X1 and X2. Actually, t-test is standard for testing the difference of two groups. However, when the groups are distributed non-normally and number of the samples is limited, it is not possible to execute the t-test. Hence, the rank sum test is applied in these situations. As a non-parametric rank sum test, Wilcoxon Rank Sum test is preferred in this study. Some conditions shall be satisfied in order to apply the Wilcoxon test; such as both populations having the same shape, and having values spread with similar shape. This test can be used to check the null hypothesis that means of these two groups are the same. If the result obtained from test, (p) is close to one; this means that the mean values of the groups are converging to each other, and vice versa. In addition, this test is also used in order to show whether the means of two groups differs each other. In order to show the efficiency of “SA” to SPSO11 statistically, this test is applied.

TABLE V. RESULTS FOR WILCOXON RANK SUM TEST, WHICH IS EXECUTED VIA SPSO11 AND SA RESULTS.

	P Wilcoxon	Absolute difference of the means given in Table 2 and 3
CF 1:	0.617	≈4.01
CF2:	0.552	≈20.99
CF3:	0.425	≈15.3
CF4:	0.507	≈4
CF5:	0.310	≈10.29
CF6:	0.187	≈80

Table 5 presents the results of Wilcoxon Rank Sum test and differences of the mean values extracted from Tables 2 and 3. For all composition functions, the means differ from each other. From Wilcoxon test, composition benchmark functions one, two and three has the almost similar mean difference, however for function two the results are presented larger than expected. For larger number of independent Monte Carlo runs, the difference becomes closer to 4-5.

D. *Comparison Results - 2*

In the literature, many researchers propose their own methods and generally evaluate them via benchmark functions. But for some specific studies, the effect of the proposed component might not be necessarily constant throughout the whole iterations of the algorithm. As an example, the “inertia weight” in PSO can be considered. The influence of this add-on parameter decays along with iterations; generally from 0.9 down to 0.4. Now, in order to achieve better performance, we apply the proposed “SA” method for only a limited number of iterations, after the limited iterations finished the algorithm continues with the original SPSO11 position and velocity update formulation. For example the algorithm begins with proposed case SA formulation given in Eq. (18) after 100 iterations, instead of Eq. (18), Eq.s(8) and(9) are evaluated in the code. The implementation results are presented in Table 4.

From these results, it is not quite possible to come up with a general rule-of-thumb conclusion since for CF1, 2, and 6 proposed case SA should applied for all iterations; for CF3 SA formulation only applied for first 100 generation is sufficient and CF4-6 SA should apply for first thousand iterations for better performance (Note that for CF6 all iteration and first thousand iterations present the same performance). It is clear that implementation of the proposed method for a limited number of iterations will increase the performance. However, the question of “What is the exact number of iteration for proposed method” is still unanswered; and it remains as a future work.

VI. CONCLUSION

This study aimed to increase the performance of the SPSO11 and provide sufficient results to show the improvement. Three different cases are defined as; CA: SPSO11 with Cognitive Avoidance, SA: SPSO11 with Social Avoidance, CSA: SPSO11 Cognitive Social Avoidance. From the first results of simulations, it is observed that only the “SA: SPSO11 with Social Avoidance”, outperforms to SPSO11. Among all proposed cases, only SA (Social Avoidance) shows better statistical results. This is also shown with the aid of a statistical test. Unlike conventional PSO variants, SPSO11 uses average of three vectors for determining the center of the circle which uses as a selection area of the new position and velocity (Eq. (22)).

$$G_i^{(d)}[k+1]=\frac{x_i^{(d)}[k]+z_i^{(d)}[k]+q_i^{(d)}[k]}{3} \qquad (22)$$

From the above equation, the center of the selection circle is equally depended on social (*q*) and cognitive (*z*) parameters with the previous position (*x*). From this equation CA changes cognitive vector (*q*), SA changes social vector (*z*), and CSA changes both of them (*z* and *q*). The results show that as the particles move away from their and neighbors worst position causes to miss the optima, since the problem may have sharp geometric changes on the surface, which means that the optima may present in a different area of the problem, in other word the CSA definition reduces the exploration property of all particles. Hence, both social and cognitive avoidance has negative influence on the performance of the algorithm. As mathematical description (Eq. (23)) the new particle is selected in a range between best and worst results. That means relatively smaller selection area is defined since the position has domination as given in Eq. (23).

$$G_i^{(d)}[k+1] = \frac{3x_i^{(d)}[k] + c\left[\left(p_{best,i}^{(d)}[k] - p_{worst,i}^{(d)}[k]\right) + \left(l_{best,i}^{(d)}[k] - l_{worst,i}^{(d)}[k]\right)\right]}{3} \quad (23)$$

The case CA also presents no better than CSA. As explained before the gravity equation of CA is given in Eq. (24), and SA which is the case that present best performance in this study given in Eq. (25).

$$G_i^{(d)}[k+1] = \frac{3x_i^{(d)}[k] + c\left(p_{best,i}^{(d)}[k] + l_{best,i}^{(d)}[k] - p_{worst,i}^{(d)}[k]\right)}{3} \quad (24)$$

$$G_i^{(d)}[k+1] = \frac{3x_i^{(d)}[k] + c\left(l_{best,i}^{(d)}[k] + p_{best,i}^{(d)}[k] - l_{worst,i}^{(d)}[k]\right)}{3} \quad (25)$$

The domination of previous position reduced in both formulations for CA and SA. The all cases defined in this paper have an effect of decreasing the selection area of new particle by adding a negative term to the formulation. However, the effects are smaller for cases SA and CA than case CSA. The Eq.s (24) and (25) looks similar to each other. The difference is tendency of particle from which worst property that they move away from it. When compared to personal worst with neighborhood's worst with respect to vector difference, it is highly probable that the difference of l_{worst} has much further away from the particle's current position than personal worst position. Hence, with this definition, new positions are now selected in an area that much closer to neighborhood best position and much wider area. These reasons are cause improvement on the performance of SPSO11 with formulation of case SA.

The second part of the study is about achievement of further increase in the performance of "SA" by applying it throughout only a limited number of iterations. It is shown that the performance can be improved by this way, even though the exact number of halting this behavior still remains as an open question. As a future study, not only the exact number of iterations, but also the ideal values of the parameters for the proposed cases will be investigated.

ACKNOWLEDGMENT

This study is made possible by a joint grant from TUBITAK (with Grant Nr. 112E168) and ANCS-UEFISCDI (with Grant Nr. 605/01.01.2013). The authors would like to express their gratitude to these institutions for their support.

REFERENCES

[1] G. Beni, J. Wang, "Swarm intelligence in cellular robotic systems," NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy, 1989.

[2] M. Dorigo, "Optimization, learning and natural algorithms (in Italian)," Ph.D. Thesis, Politecnico di Milano, Italy, 1992.

[3] M. Dorigo, G. DiCaro and L.M. Gambardella, "Ant algorithms for discrete optimization," Artificial Life, vol. 5, no. 2, pp. 137–172, 1999.[Online].Available: http://dx.doi.org/10.1162/106454699568728

[4] J. Kennedy, R. Eberhart, "Particle swarm optimization," IEEE International Conference on Neural Networks, pp. 1942–1948, 1995. [Online]. Available: http://dx.doi.org/10.1109/ICNN.1995.488968

[5] G. Bilchev, I. C. Parmee, "The ant colony metaphor for searching continuous design spaces," AISB Workshop on Evolutionary Computation, vol. 993, pp. 25–39, 1995. [Online]. Available: http://dx.doi.org/10.1007/3-540-60469-3_22

[6] N. Monmarché, G. Venturini, M. Slimane, "On how pachycondyla apicalis ants suggest a new search algorithm," Future Generation Computer Systems, vol. 16, no. 8, pp. 937–946, 2000. [Online]. Available: http://dx.doi.org/10.1016/S0167-739X(00)00047-9.

[7] J. Dréo, P. Siarry, "A new ant colony algorithm using the hierarchical concept aimed at optimization of multi minima continuous functions," 3rd International Workshop on Ant Algorithms (ANTS'2002), vol. 2463, pp. 216–221, 2002.[Online].Available: http://dx.doi.org/10.1007/3-540-45724-0_18

[8] J. Kennedy, R. C. Eberhart, "Discrete binary version of the particle swarm algorithm," IEEE International Conference on Systems, Man, and Cybernetics, pp. 4104-4108, 1997. [Online]. Available: http://dx.doi.org/10.1109/ICSMC.1997.637339

[9] M. Clerc, "The swarm and the queen: towards a deterministic and adaptive particle swarm optimization," Congress on Evolutionary Computation, pp. 1951-1957, 1999. [Online]. Available: http://dx.doi.org/ 10.1109/CEC.1999.785513.

[10] C. K. Mohan, B. Al-kazemi, "Discrete particle swarm optimization," Workshop on Particle Swarm Optimization, Purdue School of Engineering and Technology, Indianapolis, IN, 2001.

[11] J. Robinson, Y. Rahmat-Samii, "Particle swarm optimization in electromagnetics," IEEE Transactions on Antennas and Propagation vol. 52, no. 2, pp. 397-407, 2004. [Online]. Available: http://dx.doi.org/10.1109/TAP.2004.823969.

[12] P. Poli, "Analysis of the publications on the applications of particle swarm optimization," Journal of Artificial Evolution and Applications, Article ID: 685175, 2008. [Online]. Available: http://dx.doi.org/10.1155/2008/685175.

[13] M. Zambrano-Bigiarini, M. Clerc, R. Rojas, "Standard Particle Swarm Optimisation 2011 at CEC-2013: A baseline for future PSO improvements," IEEE Congress on Evolutionary Computation (CEC), pp. 2337 – 2344, 2013. [Online]. Available: http://dx.doi.org/ 10.1109/CEC.2013.6557848.

[14] C. Yang, D. Simon, "A new particle swarm optimization technique," 18th International Conference on Systems Engineering, pp. 164-169, 2005. [Online]. Available: http://dx.doi.org/10.1109/ICSENG.2005.9.

[15] G. Ciuprina, D. Ioan, I. Munteanu, "Use of intelligent-particle swarm optimization in electromagnetics," IEEE Transactions on Magnetics, vol. 38, no. 2, pp. 1037-1040, 2002. [Online]. Available: http://dx.doi.org/10.1109/20.996266.

[16] O.T.Altinoz, A.E.Yilmaz and G.Ciuprina, "Use of Karczmarz's method in Intelligent-Particle Swarm Optimization", International Conference on Electrical and Electronics Engineering, pp. 526-530, Bursa, 2013. [Online]. Available: http://dx.doi.org/10.1109/ ELECO.2013.6713898

[17] A. Biswas, A. Kumar, K.K. Mishra, "Particle Swarm Optimization with cognitive avoidance component," International Conf. on Advances in Computer, Communication and Informatics, pp. 149 - 154, 2013. [Online]. Available: http://dx.doi.org/10.1109/ ICACCI.2013.6637162.

[18] J.J. Liang, P.N. Suganthan and K. Deb, "Novel Composition Test Functions for Numerical Global Optimization," IEEE Congress on Evolutionary Computation, pp. 68-75, 2005. Matlab codes of benchmark problems: http://www.ntu.edu.sg/home/epnsugan/ index_files/SIS2005-function-codes.zip

[19] F.C. Lam, M.T. Longnecker, "A modified Wilcoxon rank sum test for paired data," Biometrika, vol. 70, no. 2, pp. 510-513, 1983. [Online]. Available: http://dx.doi.org/10.1093/biomet/70.2.510

[20] C. K. Monson and K. D. Seppi, "Exposing Origin-Seeking Bias in PSO," Genetic and Evolutionary Computation Conference GECCO, pp. 241-248, 2005. [Online]. Available: http://dx.doi.org/ 10.1145/1068009.1068045.