Revista de Matemática: Teoría y Aplicaciones 2016 **23**(1) : 111–142
cimpa – ucr issn: 1409-2433 (Print), 2215-3373 (Online)

SC: A NOVEL FUZZY CRITERION FOR SOLVING ENGINEERING AND CONSTRAINED OPTIMIZATION PROBLEMS

SC: UN NUEVO CRITERIO DIFUSO PARA RESOLVER PROBLEMAS DE INGENIERÍA Y DE OPTIMIZACIÓN CON RESTRICCIONES

Sergio G. de-los-Cobos-Silva*
Miguel A. Gutiérrez-Andrade† Eric A. Rincón-García‡
Pedro Lara-Velázquez§ Roman A. Mora-Gutiérrez¶
Antonin S. Ponsich‖

Received: 2 Oct 2014; Revised: 5 Oct 2015;
Accepted: 6 Oct 2015

Abstract

In this paper a novel fuzzy convergence system (SC) and its fundamentals are presented. The model was implemented on a monoobjetive PSO algorithm with three phases: 1) Stabilization, 2) generation and breadth-first search, and 3) generation and depth-first. The system SC-PSO-3P was tested with several benchmark engineering problems and with several CEC2006 problems. The computing experience and comparison with previously reported results is presented. In some cases the results reported in the literature are improved.

Keywords: particle swarm optimization (PSO); optimization.

Resumen

En este trabajo se presenta un novedoso sistema de convergencia (SC), sus fundamentos y la experiencia computacional. Se implementó en un algoritmo PSO monoobjetivo de tres fases: Estabilización, generación y búsqueda en amplitud, generación y búsqueda a profundidad, el cual se probó con diversos problemas benchmark tanto de ingeniería como de la serie CEC2006. La experiencia computacional y la comparación con resultados previamente reportados se presenta. En algunos casos, se mejoran los resultados de la literatura.

Palabras clave: optimización por enjambres de partículas; optimización.

Mathematics Subject Classification: 90C26, 90C29, 90C59.

1 Introduction

In general, an optimization problem can be defined as:

$$\min_{x \in X} h(x).$$

Many real world problems can be expressed as continuous optimization problems, either with or without restrictions. Due to the complexity and intractability

*Universidad Autónoma Metropolitana-Iztapalapa, Departamento de Ingeniería Eléctrica, Av. San Rafael Atlixco 186, Col. Vicentina, Del. Iztapalapa, México D.F., C.P. 09340, México. E-Mail: cobos@xanum.uam.mx

†Same address as S. de-los-Cobos-Silva. E-Mail: gamma@xanum.uam.mx

‡Universidad Autónoma Metropolitana-Azcapotzalco, Departamento de Sistemas, Av. San Pablo 180, Colonia Reynosa Tamaulipas, México D.F., C.P. 02200, México. E-Mail: rigaeral@correo.azc.uam.mx

§Same address as S. de-los-Cobos-Silva. E-Mail: plara@xanum.uam.mx

¶Same address as E. Rincón-García. E-Mail: mgra@correo.azc.uam.mx

‖Same address as E. Rincón-García. E-Mail: aspo@correo@correo.azc.uam.mx

of many of these problems different heuristic solution methods have been developed, among others: Tabu Search (TS) [9], Simulated Annealing (SA) [21], Genetic Algorithms (GA) [15], Scatter Search (SS) [10], Particle Swarm Optimization (PSO) [19], just for PSO, there are so many different versions that a treaty on its taxonomy can be found in [31]. For a review of its variants, see [16].

With respect to constrained optimization: Toscano-Pulido and Coello (2004) [33]presented a simple mechanism to handle constraints with PSO based on closeness of a particle to feasible region in order to select a leader, this approach also incorporates a mutation operator. The authors perform 340,000 objective function evaluations. Lu & Chen (2008) [26] presented an algorithm called self-adaptive velocity PSO. To handle constrains, they adopted a dynamic-objective constraint-handling method. Liu et al. (2010) [25] proposed an hybrid PSO named PSO-DE, which integrates PSO with differential evolution (DE) in order to force jump out of stagnation. Mazhoud et al. (2013) [27] proposed constraint-handling mechanism consists of closeness evaluation of a particle to the feasible region, at each iteration, the constraint total violation function is introduced as a second objective, in order to solve the bi-objective optimization problem they use a basic lexicographic method, the algorithm is named CVI-PSO and the algorithms have been tested over 24 benchmark functions with 25,000 fitness function evaluations(FFE) for problems with 2 until 24 variables.

In particular, with the SC-PSO-3P model, evidence of convergence for the Grienwank function with 120,000 variables $x_i \in [-600, 600]$ is presented (see Figure 1) using only 3 particles and reaching global optimum in less than 90 iterations on average and 40 seconds on average. The instances were solved using Matlab and running it on a Notebook with an Intel Atom N280 processor at 1.66 Ghz. Subsequently, in the section of numerical examples, an Intel Core i5-3210M processor computer at 2.5 GHz computer was used.

This paper presents a novel approach that guarantees the solution in a single algorithm. The work is divided as follows: background of fuzzy numbers are presented in the second section; the proposed Convergence System SC, as well as its definitions and fundamentals are presented in the third section. The general guidelines of PSO are presented in the fourth section. Numerical examples are presented in the fifth section. Finally, conclusions and future research are presented.

2 Fuzzy numbers

In this section, we introduce the basic concepts of fuzzy numbers based on [6].

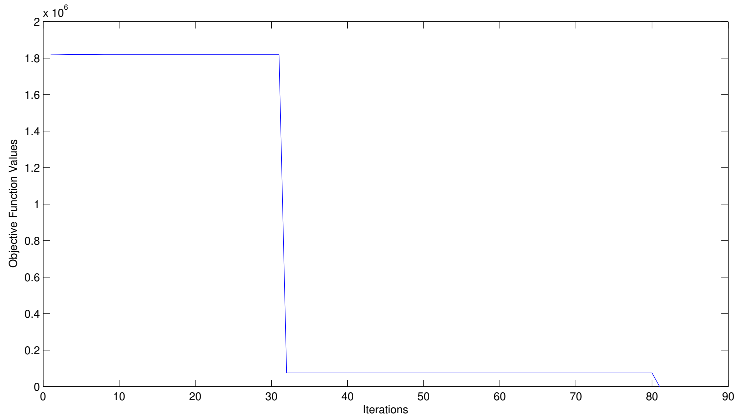

Figure 1: Convergence with SC for Griewank function dimension=120,000.

A fuzzy number $A = (a, b, c, d; w)$ is defined as a fuzzy subset of the real line $\Re$ with membership function h_A such that:

1. h_A is continuous mapping from $\Re$ to the closed interval $[0, w]$.
2. $h_A = 0, \forall x \in (-\infty, a]$.
3. h_A is a strictly increasing function on $[a, b]$.
4. $h_A = w, \forall x \in [b, c]$, where w is a constant and $0 \leq w \leq 1$.
5. h_A is strictly decreasing on $[c, d]$.
6. $h_A = 0, \forall x \in [d, +\infty)$,

where $0 \leq w \leq 1$, and $a, b, c, d \in \mathbb{R}$, and $a \leq b \leq c \leq d$.

If $w = 1$, the generalized fuzzy number A is called a normal trapezoidal fuzzy number (see Fig. 2) denoted as $A = (a, b, c, d)$. If $a = b$ and $c = d$, then A is a crisp interval. If $b = c$, then A is a generalized triangular fuzzy number. If $a = b = c = d$, then A is a real number.

The membership function h_A of A can be expressed as:

$$h_A = \begin{cases} h_A^L(x), & a \leq x \leq b, \\ w & b \leq x \leq c, \\ h_A^R(x), & c \leq x \leq d, \\ 0, & \text{otherwise.} \end{cases}$$

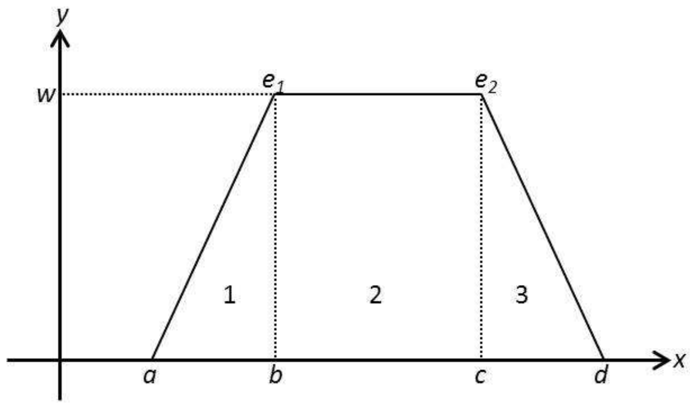

Figure 2: Trapezoidal fuzzy number.

where: $h_A^L(x) : [a,b] \longrightarrow [0,w]$ and $h_A^R(x) : [c,d] \longrightarrow [0,w]$ are continuous, $h_A^L(x)$ is strictly increasing and $h_A^R(x)$ strictly decreasing. The inverse functions of $h_A^L(x)$ and $h_A^R(x)$ are denoted by $g_A^L(x)$ and $g_A^R(x)$, respectively. These functions are continuous on $[0,w]$, this means both $\int_0^w g_A^L(x)$ and $\int_0^w g_A^R(x)$ exist [23].

Let be two trapezoidal fuzzy numbers. $\tilde{B}_1 = (a_1, b_1, c_1, d_1; w)$ and $\tilde{B}_2 = (a_2, b_2, c_2, d_2; w)$ and $c \in \mathbb{R}$, then:

1. $c\tilde{B}_1 = (ca_1, cb_1, cc_1, cd_1; w)$.
2. $\tilde{B}_1 + \tilde{B}_2 = (a_1 + a_2, b_1 + b_2, c_1 + c_2, d_1 + d_2; w)$.

3 Convergence system, SC

This section is based on [5]. Let:

1. $0 \leq f_1, f_2, f_3 \leq 1$,
2. $f_1 + f_2 + f_3 = 1$,
3. $0 < w \leq 1$,
4. $a_i \in \mathbb{R}, \alpha_i \in \mathbb{R}$.

Consider the following class of fuzzy numbers:

$$C(\tilde{\mathbf{B}}) = \{\tilde{B}_i = (a_i, f_1, f_2, f_3, \alpha_i; w) | \text{ satisfying the preceding four conditions}\},$$

where:

1. $b_i = a_i + f_1\alpha_i$;

2. $c_i = a_i + (f_1 + f_2)\alpha_i$;

3. $d_i = a_i + (f_1 + f_2 + f_3)\alpha_i = a_i + \alpha_i$.

Property 1 *For any two fuzzy numbers in $\tilde{B}_1, \tilde{B}_2 \in C(\tilde{\mathbf{B}})$, such that $\tilde{B}_1 = (a_1, f_1, f_2, f_3, \alpha_1; w)$ and $\tilde{B}_2 = (a_2, f_1, f_2, f_3, \alpha_2; w)$ and $\forall c \in \mathbb{R}$, satisfied:*

1. $\tilde{B}_1 c = (ca_1, f_1, f_2, f_3, c\alpha_1; w)$.

2. $\tilde{B}_1 + \tilde{B}_2 = (a_1 + a_2, f_1, f_2, f_3, \alpha_1 + \alpha_2; w)$.

Definition 1 *Given a function $G : C(\tilde{\mathbf{B}}) \longrightarrow \mathbb{R}$, and $\tilde{B}_1, \tilde{B}_2 \in C(\tilde{\mathbf{B}})$ it is said that they are SC-equivalent if and only if:*

$$G(\tilde{B}_1) = G(\tilde{B}_2).$$

Definition 2 *Given a function $G : C(\tilde{\mathbf{B}}) \longrightarrow \mathbb{R}$, and a $g_G \in \mathbb{R}$, in the codomain of G, the following SC-equivalence class is defined $\tilde{\mathbf{B}}_{g_G} \subset C(\tilde{\mathbf{B}})$ as:*

$$\tilde{\mathbf{B}}_{g_G} = \left\{ \tilde{B} \in C(\tilde{\mathbf{B}}) \mid G(\tilde{B}) = g_G \right\}.$$

Comments:

1. $\tilde{\mathbf{B}}_{g_G}$ is an equivalence class.

2. $\tilde{\mathbf{B}}_{g_1} \bigcap \tilde{\mathbf{B}}_{g_2} = \emptyset, \forall g_1 \neq g_2$

3. $\bigcup_{g \in \mathbb{R}} \tilde{\mathbf{B}}_g = C(\tilde{\mathbf{B}})$.

Definition 3 *Let f_1, f_2, f_3, w such that they satisfy:*

1. $0 \leq f_1, f_2, f_3 \leq 1$,

2. $f_1 + f_2 + f_3 = 1$,

3. $0 < w \leq 1$.

We define:

1. $F_1 = (1 + f_2)$.

2. $F_2 = (2f_1^2 + 6f_1 f_2 + 3f_2^2 + 3f_3 - 2f_3^2)$.

3. $F_3 = (3f_1^3 + 4f_2^3 + 3f_3^3 + 12f_1f_2^2 + 12f_2f_1^2 + 6f_3 - 8f_3^2)$, *It can be proven that* $F_3 > 0$.

4. $A = \frac{12}{w^3(1+3f_2)}$.

Theorem 1 *Let:*

$$G : C(\tilde{\mathbf{B}}) \longrightarrow \mathbb{R}$$

such that:

$$G(\tilde{B}) = A\left[\alpha^2 F_3 - 4\alpha F_2 + 6aF_1\right].$$

Then G is bijective on each SC-equivalence class.

Proof. See [5], Appendix A. ■

Proposition 1 *Let* $\tilde{B} \in C(\tilde{\mathbf{B}})$, $s^*, s, \epsilon \in \mathbb{R}$ *such that* $s = s^* + \epsilon$, *then:*

$$\lim_{\epsilon \to 0} G(\tilde{Bs}) = G(\tilde{Bs^*}).$$

Proof. See [5], Appendix B. ■

Proposition 2 *Let* $\tilde{B}_i \in C(\tilde{\mathbf{B}})$, *and* $s_i, \epsilon_i \in \mathbb{R}$ *such that* $s_i = s_i^* + \epsilon_i$, $i = 1,2,\ldots,n$

$$\lim_{\{\epsilon_i\}_{i=1}^n \to 0} G\left(\sum_{i=1}^n \tilde{B}_i s_i\right) = G\left(\sum_{i=1}^n \tilde{B}_i s_i^*\right).$$

Proof. See [5], Appendix C. ■

Definition 4 *Consider the following multiobjective optimization problem* $h(x)$, $x \in X \subseteq \mathbb{R}^n$, *where:*

$$h(x) = (h_1(x), h_2(x), \ldots, h_k(x)); \;\; h_i : \Re^n \to \mathbb{R}, i = 1, 2, \ldots, k.$$

Then:

1. *Given* $y = (y_1, y_2, \ldots, y_k)$ *is said that it dominates* $z = (z_1, z_2, \ldots, z_k)$ *if and only if* $\forall i \in \{1, 2, \ldots, k\}$ $y_i \leq z_i$ *and* $\exists i_0 \in \{1, 2, \ldots, k\}$ *such that* $y_{i_0} < z_{i_0}$.

2. *Pareto Frontier (PF) = {* $h(x)|$ *are non dominated}.*

3. *A solution vector* x^* *is said to be Pareto Optimal if and only if there is no other vector* x *such that* $h(x)$ *dominates* $h(x^*)$.

Definition 5 *The SC-Frontier (SC-F) is defined as:*

$SC\text{-}F = \{h(x)|\forall\epsilon > 0 \exists y \in Codomine(h) - Image(h) \text{ such that } ||y - h(x)|| < \epsilon\}.$

Proposition 3 *PF $\subseteq$ SC-F.*

Proof. See [5], Appendix D. ■

Definition 6 *Given a multiobjective optimization problem* $h(x), \ \ x \in X \subseteq \mathbb{R}^n$, *where:*

$$h(x) = (h_1(x), h_2(x), \ldots, h_k(x)); \ \ h_i : \Re^n \to \mathbb{R}, i = 1, 2, \ldots, k,$$

and

$$\tilde{\mathbf{B}} = \left(\tilde{B}_1, \tilde{B}_2, \ldots, \tilde{B}_k\right)$$

where $\tilde{B}_i \in C(\tilde{\mathbf{B}}), \ \ i = 1, \ldots, k$. *Define:*

$$G\left(\tilde{\mathbf{B}}h(x)\right) = G\left(\sum_{i=1}^{k} \tilde{B}_i h_i(x)\right).$$

Definition 7 *Given a multiobjective optimization problem and a* $\tilde{\mathbf{B}}$*, define:*

$$x^*_{\tilde{B}} = \textit{argmin}\ \{G(\tilde{\mathbf{B}}h(x)) \ | \ h(x) \in \textit{SC-F}\}.$$

Definition 8 *The optimal set of SC-(SC-O) solutions is defined as:*

$$\textit{SC-O} \ = \{x^*_{\tilde{B}}, \ \ \tilde{B} \in C(\tilde{\mathbf{B}})\}.$$

3.1 Empirical boundary conditions

Given an optimization problem with k objective functions, the following values have been used empirically:

1. $f_1 = f_3 = 0.25, f_2 = 0.5, w = 1,$
2. $\alpha_i \in [-12K_1, 12K_1], i = 1, 2, \ldots, k,$
3. $a_i = K_2 + K_3 g_i,$

where: $K_1 = \frac{2F_2}{F_3}, K_2 = \frac{2F_2}{3kF_1}, K_3 = \frac{A^{-1}F_3}{6kF_1}, g_i = \begin{cases} 1 & \text{if} \quad h_i(x^*) > 0 \\ 0 & \text{if} \quad h_i(x^*) = 0 \\ -1 & \text{if} \quad h_i(x^*) < 0. \end{cases}$

3.2 SC algorithm

Step 0:

1. Provide stopping criteria
2. Set f_1, f_2, f_3, w such that they satisfy the conditions of Theorem 1.
3. Given the function to optimize $h = (h_1, h_2, \ldots h_k)$, find the boundary conditions (or use the suggested empirical conditions), take $\tilde{B}_i \in C(\tilde{\mathbf{B}})$, $i = 1,2,\ldots, k$.
4. Set values for a_i and α_i.
5. Take any value x in the Domain of h and evaluate:

$$G\left(\tilde{B}h(x)\right) = G\left(\sum_{i=1}^{k} \tilde{B}_i h_i(x)\right).$$

Set:

$$g_G(x) \leftarrow G\left(\tilde{B}h(x)\right).$$

Step 1: Through m neighborhoods, find $x_j \in Phase^j(x)$ and evaluate $G\left(\tilde{B}h(x_j)\right), j = 1, 2, \ldots, m$

$$g_G(x_1) = \min_j \{G(\tilde{B}h(x_j)), G(\tilde{B}h(x))\},$$

$$x \leftarrow x_1.$$

Step 2: Stop in the following conditions:

1. If for a certain number of iterations $g_G(x_1) = g_G(x)$, then an optimum was reached.
2. Stop criteria is met.

In any other case go to **Step 1**.

Remark: when g_G is very close to zero, it can be used, for example: $(1 - g_G)^2$.

4 Particle swarm optimization (PSO)

The particle swarm optimization is a metaheuristic based on swarm intelligence and has its roots in artificial life, social psychology, engineering and computer science. PSO differs from evolutionary computation (c.f. [20]) because the population members or agents, also called particles, are "flying" through the problem hyperspace.

PSO is an adaptive method that uses agents or particles moving through the search space using the principles of evaluation, comparison and imitation [20].

PSO is based on the use of a set of particles or agents that correspond to states of an optimization problem, where each particle moves across the solution space in search of an optimal position or at least a good solution. In PSO, agents communicate with each other, and the agent with the best position (measured according to an objective function) influences the others by attracting them towards itself.

The population is initialized by assigning a random position and speed to each agent. At each iteration, the velocity of each particle is randomly accelerated towards its best position (where the value of the fitness function or objective function improves) and also considering its neighbors' best position.

To solve a problem, PSO uses a dynamic management of particles; this approach allows breaking cycles and diversifying the search. In this work, a r-particle swarm is represented at time t under the form:

$$\theta_{1t}, \theta_{2t}, \ldots, \theta_{rt}$$

with $\theta_{jt} \in D, j = 1, 2, \ldots, r$, then a movement of the swarm is defined according to equation 1:

$$\theta_{jt+1} = \theta_{jt} + V_{jt+1} \tag{1}$$

where the velocity V_{jt+1} is given in equation 2:

$$V_{j,t+1} = \alpha V_{j,t} + rand(0, \varphi_1)[\theta'_{j,t} - \theta_{j,t}] + rand(0, \varphi_2)[\theta'_{g,t} - \theta_{j,t}], \tag{2}$$

where:

D: space of feasible solutions,

$V_{j,t}$: speed at time t of the j-th particle,

$V_{j,t+1}$: speed at time $t+1$ of the j-th particle,

$\theta_{j,t}$: j-th particle at time t,

$\theta'_{g,t}$: the particle with the best value found so far (i.e., before time t),

$\theta'_{j,t}$: best position found so far by the j-th particle (before time t),

$rand(0,\varphi)$: random number uniformly distributed over the interval [0,φ],

α: inertia weight factor.

The PSO algorithm is described in Table 1.

Table 1: PSO algorithm.

1. Create a population of particles distributed in the feasible space.
2. Evaluate each position of the particles according to the objective function (fitness function).
3. If the current position of a particle is better than the previous one, update it.
4. Determine the best particle (according to the best previous positions).
5. Update the particle velocities $j = 1, 2, \ldots, r$ according to equation 2.
6. Move the particles to new positions according to equation 1.
7. Go to Step 2 until the termination criterion is satisfied.

4.1 SC-PSO-3P

The three-phase PSO algorithm along with the SC criterion is named SC-PSO-3P. The SC-PSO-3P algorithm is described in Table 1.

As can be seen in the algorithm in table 1, the key modification is to consider the SC criterion, and the 3 phases, the rest of the algorithm remains unchanged. It should be mentioned, that any other search algorithm can be used instead of PSO, although the respective tests must be performed.

We describe the main characteristics of the proposed algorithm, called PSO-3P. This algorithm is based in a traditional PSO heuristic. However, the position of the particles can be modified using different strategies, that are applied in different times, or phases, of the searching process.

In phase 1, called stabilization, according to the description presented in section 4, the PSO-3P algorithm generates randomly a set of particles in the solution space. Then, during itF_1 iterations the position of the particles is modified using equations (1) and (2). Thus, at the end of this phase the particles are concentrated, or stabilized, in a promising region.

When phase 1 is completed, a breadth-first search strategy, called phase 2, is incorporated. In this phase, if the global best solution is not improved after three consecutive iterations, the position of M_2 particles is modified randomly. However, the particle with the best known position is preserved. Thus, the population is dispersed in the solution space, but it can be attracted to the best region visited so far. This diversification strategy is considered during itF_2 iterations.

Finally, phase 3 is initialized. During itF_3 iterations the following depth-first search strategy is applied. If the global best solution is not improved after three consecutive iterations, M_3 particles are randomly positioned in a neighborhood of the best known solution. Thus, phase 3 includes an intensification process in a promising region.

The algorithm was implemented in Matlab R2008a and was run in an Intel Core i5-3210M processor computer at 2.5 GHz, running on Windows 8.

5 Computational results

The parameters were set as follows, $\phi_1 = \phi_2 = 2.05$, and $\phi = \phi_1 + \phi_2$, then:

$$\alpha = \frac{2}{\phi-2+\sqrt{\phi^2-4*\phi}}; \quad \varphi_1 = \alpha * \phi_1; \quad \varphi_2 = \alpha * \phi_2.$$

For phase 2 and phase 3 the parameters were set as $c = 3$, $itF_1 = 75$, $itF_2 = 150$, $itF_3 = MaxIt$, $prop = 0.25$.

The select problems have been well studied before as benchmarks by various approaches. The 24 constrained problems(G1-G24) have been solved before using the following algorithms : self-adaptive multi-strategy differential evolution with local search(Memetic-SAMSDE) [7], adaptive penalty formulation with GA(APF-GA) [32], modified differential evolution(MDE) [29], adapted constrained particle swarm optimization(CVI-PSO) [27], a differential evolution combined variants (DECV) [29] and a heuristic inspired by T-Cell model of the immune system (T-CELL) [1].

The three engineering problems considered are a pressure vessel design problem(eng01) [18], a welded beam design problem(eng02) [30] and a tension-compression string design problem(eng03) [2]. Eng01 and eng03, have been solved by the following approaches: constraint violation with interval arithmetic PSO(CVI-PSO) [27], a co-evolutionary PSO(CPSO) [11], a hybrid PSO(HPSO) [12], an accelerating technique(AATM) [35], a T-Cell heuristic [1]. Problem

```
Begin
while Termination criterion is not satisfied do
    Set variables c, itF1, itF2, itF3, MaxIt and prop.
    Create a population of nPop random particles.
    Set cont = 0 and it = 1. Evaluate each position of the particles
    according to the SC function.
    If the current position of a particle is better (respect to the fitness
    function) than the previous update it.
    Determine the best particle (according to the best previous positions
    against the optimization criterion). If a better particle cannot be
    founded, let cont = cont + 1.
    Update the particle velocities j = 1, 2, ..., nPop according to
    equation (2).
    (Phase 1: Stabilization) if it ≤ itF1 then
        go to Step 34.
    end
    (Phase 2: Breadth-first search) if itF1 < it ≤ itF2 then
        if cont = c then
            Set n=1. while n ≤ nPop * prop do
                Create a random particle and, with a probability bigger
                than 0.5 substitute randomly a particle in the swarm.
                Set n = n + 1.
            end
            Set cont = 0.
        end
        go to Step 34.
    end
end
(Phase 3: Depth-first search). if itF2 < it ≤ itF3 then
    if cont = c then
        Set n = 1. while n ≤ nPop * prop. do
            Create a random particle in a variable neighborhood of θ'_{g,t}
            and substitute randomly a particle in the swarm.
            Set n = n + 1.
        end
    end
    Set cont = 0.
    go to Step 34.
end
Select the best nPop particles according to SC optimization criterion.
Set it = it + 1. Go to Step 3 until the termination criterion is satisfied.
```

Algorithm 1: SC-PSO-3P algorithm.

eng02 has been solved by these approach: CVI-PSO, a GA with binary representation (GAPF) [4], a GA based co-evolution model (CGA) [3], CPSO and HPSO.

In this section the results obtained for constrained optimization will be presented, in order to show the efficiency of SC. A study in depth for multiobjective optimization, and unconstrained optimization, will be presented in future work.

In this paper, the functions for restricted optimization were taken from [22].

5.1 Constrained optimization

In this section we consider the global optimization problem with restrictions:

$$\begin{array}{lrl} \text{Minimize:} & h(x) & \\ \text{subject to:} & f_i(x) \leq 0 & i = 1, 2, \ldots, m \\ & t_j(x) = 0 & j = 1, 2, \ldots, k \\ & x \in D \subseteq S = [L, U] & \end{array} \tag{3}$$

where:

- $[L, U] = \{x = (x_1, x_2, \ldots, x_n) | l_i \leq x_i \leq u_i\} \subset \mathbb{R}^n$.
- the feasible region (D) is defined as:

$$D = \{x \in S, f_i(x) \leq 0, \ \ i = 1, 2, \ldots, m, \ \ t_j(x) = 0, \ \ j = 1, 2, \ldots, k\}.$$

The problem given in (3) can be transformed into the problem (4) [24] given by:

$$\begin{array}{crl} \text{Minimize}_{x \in D}: & F(x) = & (h_1(x), h_2(x), h_3(x)) \\ \text{with:} & h_1(x) = & h(x) \\ & h_2(x) = & P_1 \sum_{i=1}^{m} \frac{c_i(x)}{c(x)+\epsilon} \\ & h_3(x) = & P_2 \sum_{j=1}^{k} d_j \end{array} \tag{4}$$

where:

- $c_i(x) = \max\{0, f_i(x)\}. i = 1, 2, \ldots, m.$
- $c(x) = \max_{i=1,2,\ldots,m}\{c_i(x)\}.$
- $d_j(x) = \max\{0, |t_j(x)| - \delta\}. j = 1, 2, \ldots, k.$
- $\epsilon, \delta > 0, P_i >> 0, i = 1, 2.$

In this section, the efficiency is defined as:

$$\text{efficiency} = \begin{cases} 1 - abs(\frac{h(x)-h(x^*)}{h(x^*)}) & \text{if} \quad h(x^*) \neq 0 \\ \\ 1 - abs(h(x)) & \text{if} \quad h(x^*) = 0. \end{cases}$$

Function(n)	Type of h	Ratio* ρ %
G1(13)	Quadratic	0.0001
G2(20)	Non-linear	99.996
G3(12)	Polynomial	0.0000
G4(5)	Quadratic	26.925
G5(4)	Cubic	0.0000
G6(2)	Quadratic	0.0054
G7(10)	Quadratic	0.0002
G8(2)	Non-linear	0.8611
G9(7)	Polynomial	0.5314
G10(8)	Linear	0.0002
G11(2)	Quadratic	0.0000
G12(3)	Quadratic	4.5420
G13(5)	Non-linear	0.0000
G14(10)	Non-linear	0.0000
G15(3)	Quadratic	0.0000
G16(5)	Non-linear	0.0186
G17(6)	Non-linear	0.0000
G18(9)	Quadratic	0.0000
G19(15)	Non-linear	35.548
G20(24)	Linear	0.0000
G21(7)	Linear	0.0000
G22(22)	Linear	0.0000
G23(9)	Linear	0.0000
G24(2)	Linear	74.268
eng01(4)	Non-linear	39.737
eng02(4)	Non-linear	2.6638
eng03(3)	Non-linear	0.7560

Table 2: Summary ([27]), $*\rho = \frac{\text{number of solution}_{\in D}}{\text{number of solution}_{\in S}}$.

Table 2 shows the ratio(ρ) for problems G1 to G24, as well as the engineering problems eng01-eng03 and a summary of some characteristics of these problems [27].

		SC-PSO-3P	Memetic	APF-GA	MDE	CVI-PSO	DECV	T-CELL
	FFE	150000	240000	500000	500000	200000	500000	350000
G1	Best	-14.999999	-15.000000	-15.000000	-15.000000	-15.000000	-15.000000	-15.000000
	Average	-14.216335	-15.000000	-15.000000	-15.000000	-15.000000	-14.855000	-15.000000
	Std. Dev.	1.205257E+00	0.000000E+00	0.000000E+00	0.000000E+00	4.000000E-16	4.590000E-01	0.000000E+00
G2	Best	-0.760896	-0.803619	-0.803601	-0.800362	-0.800098	-0.704009	-0.801367
	Average	-0.560853	-0.803619	-0.803518	-0.786160	-0.790876	-0.569458	-0.752975
	Std. Dev.	9.614486E-02	0.000000E+00	1.000000E-04	1.260000E-02	1.091200E-02	9.510000E-02	3.209500E-02
G3	Best	-1.000588	-1.000500	-1.001000	-1.000500	-1.000000	-0.461000	-1.000000
	Average	-1.000399	-1.000500	-1.001000	-1.000500	-1.000000	-0.134000	-1.000000
	Std. Dev.	2.452163E-04	0.000000E+00	0.000000E+00	0.000000E+00	3.700000E-16	1.170000E-02	0.000000E+00
G4	Best	-30665.465015	-30665.539000	-30665.539000	-30665.538600	-30665.821702	-30665.539000	-30665.538500
	Average	-30665.287211	-30665.539000	-30665.539000	-30665.538600	-30665.820996	-30665.539000	-30665.538400
	Std. Dev.	1.155335E-01	0.000000E+00	1.000000E-04	0.000000E+00	3.391000E-03	1.500000E-06	1.000000E-04
G5	Best	5126.490316	5126.497000	5126.497000	5126.497000	5127.277667	5126.497000	5126.625500
	Average	5161.796768	5126.497000	5127.542300	5126.497000	5127.277667	5126.497000	5378.267800
	Std. Dev.	3.437387E+01	0.000000E+00	1.432400E+00	0.000000E+00	0.000000E+00	0.000000E+00	2.980173E+02
G6	Best	-6961.773245	-6961.813875	-6961.814000	-6961.814000	-6961.813876	-6961.814000	-6961.813870
	Average	-6960.340642	-6961.813875	-6961.814000	-6961.814000	-6961.813876	-6961.814000	-6961.813860
	Std. Dev.	1.163449E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	3.900000E-05
G7	Best	25.343143	24.306200	24.306200	24.306200	24.473827	24.306000	24.320900
	Average	28.080169	24.306200	24.306200	24.306200	26.561295	24.794000	24.653400
	Std. Dev.	2.217613E+00	0.000000E+00	0.000000E+00	0.000000E+00	1.642519E+00	1.370000E+00	2.198150E-01
G8	Best	-0.095825041	-0.095825000	-0.095825000	-0.095825000	-0.105495	-0.095825	-0.095825
	Average	-0.095824540	-0.095825000	-0.095825000	-0.095825000	-0.105495	-0.095825	-0.095825
	Std. Dev.	3.298078E-07	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	4.230000E-17	0.000000E+00
	Best	680.698889	680.630000	680.630000	680.630000	680.635400	680.630000	680.630000

Table 3: Comparative results for several algorithms.

		SC-PSO-3P	Memetic	APF-GA	MDE	CVI-PSO	DECV	T-CELL
	FFE	150000	240000	500000	500000	200000	500000	350000
G9	Average	681.152138	680.630000	680.630000	680.630000	680.755705	680.630000	680.630000
	Std. Dev.	3.630035E-01	0.000000E+00	0.000000E+00	0.000000E+00	7.923200E-02	3.450000E-07	1.670000E-02
	Best	7097.106239	7049.248020	7049.248020	7049.248020	7049.276586	7049.248000	7050.834200
G10	Average	7446.397229	7049.248020	7077.682100	7049.248020	7053.214311	7103.548000	8020.755100
	Std. Dev.	3.768273E+02	0.000000E+00	5.124000E+01	0.000000E+00	1.061564E+01	1.480000E+02	6.217231E+02
	Best	0.749994	0.749900	0.749900	0.749900	0.750000	0.750000	0.749900
G11	Average	0.750000	0.749900	0.749900	0.749900	0.750000	0.750000	0.749900
	Std. Dev.	3.588812E-06	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	1.120000E-16	0.000000E+00
	Best	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
G12	Average	-0.999995	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
	Std. Dev.	3.154629E-06	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
	Best	0.053449	0.053942	0.053942	0.053942	0.055556	0.059798	0.054638
G13	Average	0.055757	0.053942	0.053942	0.053942	0.065591	0.382401	0.458857
	Std. Dev.	1.93E-03	0.00E+00	0.00E+00	0.00E+00	1.02E-02	2.68E-01	3.45E-01
	Best	-48.29818726	-47.764888	-47.764790	-47.764887	-47.453011	-47.764888	-47.517100
G14	Average	-47.9751512	-47.764888	-47.764790	-47.764874	-44.424690	-47.722542	-45.310800
	Std. Dev.	3.16E-01	0.00E+00	1.00E-04	1.40E-05	1.41E+00	1.62E-01	1.12E+00
	Best	961.7218517	961.715020	961.715020	961.715020	961.715707	961.715022	961.715020
G15	Average	962.2834191	961.715020	961.715020	961.715020	961.718595	961.715022	963.374820
	Std. Dev.	9.29E-01	0.00E+00	0.00E+00	0.00E+00	6.87E-04	2.31E-13	2.28E+00
	Best	-2.156472	-1.905155	-1.905155	-1.905155	-1.905150	-1.905155	-1.905155
G16	Average	-1.833968	-1.905155	-1.905155	-1.905155	-1.905150	-1.905155	-1.905155
	Std. Dev.	1.837774E-01	0.000000E+00	0.000000E+00	0.000000E+00	8.520000E-15	1.100000E-06	0.000000E+00
	Best	8684.768462	8853.539700	8853.539800	8853.539700	8853.538913	8853.541289	8861.821000

Table 4: Comparative results for several algorithms.

		SC-PSO-3P	Memetic	APF-GA	MDE	CVI-PSO	DECV	T-CELL
	FFE	150000	240000	500000	500000	200000	500000	350000
G17	Average	8897.083482	8853.539700	8888.487600	8853.539700	8853.538913	8919.936362	8990.997000
	Std. Dev.	1.372256E+02	0.000000E+00	2.903470E+01	0.000000E+00	3.700000E-12	2.590000E+01	1.061932E+02
	Best	-0.857012	-0.866025	-0.866025	-0.866025	-0.864631	-0.866025	-0.865960
G18	Average	-0.765061	-0.866025	-0.865925	-0.866025	-0.809109	-0.859657	-0.784550
	Std. Dev.	1.125068E-01	0.000000E+00	1.000000E-04	0.000000E+00	6.271000E-02	3.480000E-02	9.746000E-02
	Best	32.180687	32.655593	32.655593	32.648270	32.827027	32.655593	33.390780
G19	Average	34.116005	32.655593	32.655593	33.341250	35.067337	32.660587	38.927610
	Std. Dev.	2.411570E+00	0.000000E+00	0.000000E+00	8.475000E-01	2.286728E+00	2.370000E-02	3.191020E+00
	Best	**5.024074E-02**	2.146897E-01	NA	NA	NA	NA	NA
G20	Average	2.181222E-01	NA	NA	NA	NA	NA	NA
	Std. Dev.	1.125954E-01	NA	NA	NA	NA	NA	NA
	Best	199.6299601	193.724500	196.633010	193.724500	193.786925	193.724510	NA
G21	Average	729.0817718	193.724500	199.515810	193.724500	193.786935	198.090578	NA
	Std. Dev.	1.726319E+02	0.000000E+00	2.356500E+00	0.000000E+00	3.300000E-05	2.390000E+01	NA
	Best	**0**	236.370331	NA	NA	NA	NA	NA
G22	Average	**0**	245.738829	NA	NA	NA	NA	NA
	Std. Dev.	**0**	9.059390E+00	NA	NA	NA	NA	NA
	Best	-400.008150	-400.055100	-399.762400	-400.055100	-400.000000	-400.055093	NA
G23	Average	-399.990059	-400.055100	-394.762700	-400.055100	-400.000000	-392.029610	NA
	Std. Dev.	2.052941E-01	0.000000E+00	3.865600E+00	0.000000E+00	0.000000E+00	1.240000E+01	NA
	Best	**-7.000000**	-5.508013	-5.508013	-5.508013	-5.508013	-5.508013	-5.508013
G24	Average	**-7.000000**	-5.508013	-5.508013	-5.508013	-5.508013	-5.508013	-5.508013
	Std. Dev.	**0.000000E+00**	0.000000E+00	0.000000E+00	0.000000E+00	9.460000E-15	2.710000E-15	0.000000E+00

Table 5: Comparative results for several algorithms.

Tables 3, 4 and 5 show that the best solutions obtained by SC-PSO-3P are competitive with those achieved by other methods. In particular, for the problem G24 a better feasible solution was consistently reached as can be seen in Table 6. It is noteworthy that for this particular problem, in all runs the solution was reached in the first iteration.

	best known	SC-PSO-3P
$h(x^*)$	-5.5080132771	**-7.000000000**
x_1^*	2.39520197477613	3
x_2^*	3.178493074117714	4
$f_1 \leq 0$	-1.7763568394e-15	-160.00
$f_2 \leq 0$	0.00	-152.00

Table 6: Table comparing the best known solution vs. SC-PSO-3P in G24.

From literature it is known that regarding the function G20 there is not a single feasible solution while for the function G22, a feasible solution is rarely found [22], for this reason is very common that these instances are not considered when they propose some method. However, it may be noticed that SC-PSO-3P is competitive results even consider that improves the results reported in the literature as can be seen from tables 7 and 8.

For instance G20, in [22] is said "This solution is a little unfeasible and no feasible solution is found so far":

[22] G20:
x^*=(1.28582343498528086E-18, 4.83460302526130066E-34, 0,
0, 6.30459929660781851E-18, 7.57192526201145068E-34,
5.03350698372404037E-34, 9.282680796166E-34, 0,
1.76723384525547359E-17, 3.55686101822965701E-34,
2.99413850083471346E-34, 0.158143376337580827,
2.29601774161699833E-19, 1.06106938611042947E-18,
1.31968344319506391E-18, 0.530902525044209539, 0,
0.289148310257773535E-18, 3.34892126180666159E-18, 0,
0.310999974151577319, 5.41244666317833561E-5,
4.84993165246959553E-16);
h^*=204.979400285636e-003.

With SC several improved solutions. Below are two of them:

SC-PSO-3P (Example 1) G20:
x^*=(1.00E-300, 1.00E-300, 1.00E-300, 1.00E-300,

1.00E-300, 1.00E-300, 1.00E-300, 1.00E-300, 1.00E-300,
1.00E-300, 1.00E-300, 0.296470402, 1.00E-300, 1.00E-300,
1.00E-300, 1.00E-300, 1.00E-300, 1.00E-300, 1.00E-300,
1.00E-300, 1.00E-300, 1.00E-300, 1.00E-300, 0.703529139);
h^*=8.999996E-02.

SC-PSO-3P (Example 2) G20:
x^*=(1.00E-300, 1.00E-300, 1.00E-300, 1.00E-300,
1.00E-300, 1.00E-300, 1.00E-300, 1.00E-300, 1.00E-300,
1.00E-300, 1.37E-06, 1.00E-300, 1.00E-300, 1.00E-300,
1.00E-300, 1.00E-300, 1.00E-300, 1.00E-300, 1.00E-300,
1.00E-300, 1.00E-300, 1.00E-300, 0.502406032, 1.00E-300);
h^*=50.24074020e-003.

[22]	SC-PSO-3P (Example 1)	SC-PSO-3P (Example 2)
2.049794E-01	**5.024074E-02**	**8.999996E-02**
$f(x) \leq 0$		
1.317752E-01	2.847350E-300	1.666667E-300
1.766032E-19	2.492499E-300	1.538462E-300
7.578526E-19	2.216294E-300	1.428572E-300
2.224047E-19	2.492499E-300	1.538462E-300
2.092945E-18	1.814211E-300	1.250000E-300
0.000000E+00	2.492499E-300	1.538462E-300
$t(x) = 0$		
1.000000E-04	-2.358454E-294	-1.227952E-299
-2.197257E-17	-4.585322E-295	-1.294290E-300
1.553971E-18	-6.610392E-295	-2.515003E-300
8.175374E-19	-8.994252E-296	7.952623E-302
1.000000E-04	-5.889714E-295	-2.843776E-300
-1.038308E-17	-1.362617E-295	-3.215523E-301
-3.346898E-17	-6.685073E-295	-2.663043E-300
-3.208538E-18	-7.027115E-296	5.820753E-301
0.000000E+00	-1.323114E-296	5.604647E-301
1.000000E-04	-1.803505E-295	-5.182852E-302
1.000000E-04	9.787500E-01	1.760682E-300
6.866522E-16	-8.951397E-297	9.840000E-01
1.000000E-04	4.975926E-01	4.586120E-07
1.000000E-04	1.200004E-01	1.433791E-07

Table 7: Table comparing G20 [22] vs. SC-PSO-3P.

In Table 7 a comparison of the values of the function and constraints is presented for the function G20. Two solutions found by SC-PSO-3P are provided, can be observed that both improve over the best reported value, however, example 2 is "less unfeasible" than example 1.

Regarding the problem G22, the best reported solution in the literature and the solution obtained by SC-PSO-3P are provided below.

[22] G22:
x^*=(236.430975504001054, 135.82847151732463, 204.818152544824585, 6446.54654059436416, 3007540.83940215595, 4074188.65771341929, 32918270.5028952882, 130.075408394314167, 170.817294970528621, 299.924591605478554, 399.258113423595205, 330.817294971142758, 184.51831230897065, 248.64670239647424, 127.658546694545862, 269.182627528746707, 160.000016724090955, 5.29788288102680571, 5.13529735903945728, 5.59531526444068827, 5.43444479314453499, 5.07517453535834395);
h^* = 236.430975504001e+000.

With SC-PSO-3P an improved solution:

SC G22:
x^*=(0, 0, 0, 0, 0, 0, 0, 100, 100,,100.010, 100, 100, 101.9292514, 0, 13.653013870, 100.00034450, 0.010, -4.700, -2.7379525440, 4.6039351080, -0.7642880070, -2.7215211340);
h^*=0.0000E00.

In Table 8 a comparison of the values of the function and constraints is presented for the function G22.

Regarding the engineering problems eng01–eng03, the results obtained by various methods are shown in Tables 9, 11 and 12. We can see that for the pressure vessel design problem, in Table 10 a better feasible solution was found with respect to those reported previously. For the other two problems, the results obtained by SC-PSO-3P are very competitive over other proposed algorithms.

In Table 13, results of the bootstrap test are shown.

[22]	SC-PSO-3P
h^*=2.36E+02	h^*=**0.00E+00**
$f_1(x) \leq 0$	
-2.21E-07	0.00E+00
$t_i(x) = 0$	
-2.93E-05	0.00E+00
9.20E-05	0.00E+00
-3.42E+07	-6.00E+07
-6.00E+07	-4.30E+07
-7.99E+07	-5.40E+07
-6.62E+07	-7.60E+07
5.61E-05	0.00E+00
1.67E-05	0.00E+00
5.41E-05	0.00E+00
4.07E+06	**3.44E-04**
1.67E-05	1.00E-02
5.74E-05	9.48E-02
5.74E-05	8.04E+00
7.48E-05	1.24E-03
7.47E-05	6.47E+00
-6.16E-07	-1.88E+00
6.12E-06	-2.84E-05
-8.78E-05	3.00E+02
9.59E-05	-3.14E-02
$\sum_i abs(t_i)$	
2.44E+08	2.33E+08

Table 8: Table comparing G22 [22] vs. SC.

	FFE	Best	Average	Worst	Std. Dev.
SC-PSO-3P	50000	**5898.549386**	6278.578324	7335.299154	2.551714E+02
CPSO	200000	6061.077700	6147.133200	6363.804100	8.645450E+01
HPSO	81000	6059.725500	6099.932300	6288.677000	8.620220E+01
AATM	30000	6059.725500	6061.987800	6090.802200	4.700000E+00
T-Cell	81000	6390.554000	6737.065100	7694.066800	3.570000E+02
CVI-PSO	25000	6059.714300	6292.123100	6820.410100	2.884550E+02

Table 9: Results of different algorithms for pressure vessel design problem.

	SC-PSO-3P	Best Known
$h(x^*)$	**5898.549386**	6059.7143
x_1^*	0.778643603	0.8125
x_2^*	0.38712201	0.4575
x_3^*	40.33557909	42.0984456
x_4^*	200	176.6365958
$f_1 \leq 0$	-1.66926028E-04	-1.05992992E-12
$f_2 \leq 0$	-2.32058574E-03	-3.58808290E-02
$f_3 \leq 0$	-1.13500617E+03	6.23567029E-06
$f_4 \leq 0$	-4.00000000E+01	-6.33634042E+01

Table 10: Results comparative pressure vessel problem best known vs. SC-PSO-3P.

	FFE	Best	Average	Worst	Std. Dev.
SC-PSO-3P	50000	1.724958	1.738032	1.843013	2.555789E-02
GAPF	NA	2.433116	NA	NA	NA
CGA	900000	1.748309	1.771973	1.785835	1.122000E-02
CPSO	200000	1.728024	1.748831	1.782145	1.292600E-02
HPSO	81000	1.724852	1.749040	1.814295	4.004900E-02
CVI-PSO	25000	1.724852	1.725124	1.727665	6.120000E-04

Table 11: Results of different algorithms for the welded beam design problem.

	FFE	Best	Average	Worst	Std. Dev.
SC-PSO-3P	50000	0.0126687	0.0128715	0.0137703	2.538650E-04
CPSO	200000	0.0126747	0.0127300	0.0129240	5.190000E-05
HPSO	81000	0.0126652	0.0127072	0.0127191	1.580000E-05
AATM	25000	0.0126682	0.0127080	0.0128613	4.500000E-05
T-Cell	36000	0.0126650	0.0127320	0.0133090	9.400000E-05
CVI-PSO	25000	0.0126655	0.0127310	0.0128426	5.500000E-05

Table 12: Results of different algorithms for the tension-compression string problem.

Problem	Low limit	Upper limit
1	-13.69776169	-14.62498168
2	-0.524584754	-0.599460572
3	-1.000298967	-1.00048471
4	-30665.24483	-30665.34067
5	5174.327258	5148.021136
6	-6959.868837	-6960.770545
7	28.99774975	27.28710176
8	-0.095824409	-0.095824666
9	681.3461332	681.0367763
10	7618.087189	7321.516701
11	0.750001544	0.7499987
12	-0.999994195	-0.999996666
13	0.056508561	0.055016703
14	-47.83817353	-48.08651249
15	962.6392106	961.9545713
16	-1.76784791	-1.903749563
17	8955.82206	8848.421709
18	-0.684804785	-0.777015372
19	38.29876149	34.66589102
21	794.7558171	661.3324645
23	-399.9122071	-400.0644544
24	-7	-7

Table 13: Results of bootstrap test with an $\alpha = 0.05$.

6 Conclusions and further research

In this work, a novel fuzzy criterion called System of Convergence SC is proposed, which was implemented using a PSO based algorithm with three phases: 1) Stabilization, 2) Generation and breadth-first search, and 3) Generation and depth-first. This algorithm SC-PSO-3P was tested in a set of benchmark instances for constrained optimization and engineering problems.

As an important remark, a deeper study concerning the boundary conditions of the parameters a and α of SC, as well as their implementation with an algorithm different from PSO-3P must be performed.

The SC-PSO-3P generates best solutions by problems G20 and G22 and G24 that results reported in the literature. Based on information of Table 14, we can say that our method has a behaviour similar that DECV and T-CELL.

	SC-PSO-3P	Memetic	APF-GA	MDE	CVI-PSO	DECV	T-CELL
SC-PSO-3P	false	true	true	true	true	false	false
Memetic	true	false	false	false	false	true	true
APF-GA	true	false	false	false	false	true	true
MDE	true	false	false	false	true	true	true
CVI-PSO	true	false	false	true	false	false	false
DECV	false	true	true	true	false	false	false
T-CELL	false	true	true	true	false	false	false

Table 14: Results of Wilcoxon Rank test for average normalized values.

It is also necessary to conduct a further study about the topology generated by SC. It was observed that in all the cases studied, SC can reach the global optimum or being very close to it with shorter iteration times and less iterations.

A Proof of theorem 1

Proof. G **is injective** by construction.

G **is surjective**:
Let $g \in \mathbb{R}$, to prove that there is $\tilde{\mathbf{B}}_g \subset C(\tilde{\mathbf{B}})$ such that:

$$G(\tilde{B}) = g, \forall \tilde{B} \in \tilde{\mathbf{B}}_g$$

it will be proven by construction.

Let $\tilde{\mathbf{B}}_g = \{\tilde{B} = (a, f_1, f_2, f_3, \alpha; w)\}$ which satisfy the conditions of theorem 1, and let:

$$\begin{aligned}
\alpha &= \frac{2F_2}{F_3}, \\
a &= \frac{4F_2^2 + A^{-1}F_3 g}{6F_1F_3}.
\end{aligned}$$

By definition it is known that:

$$\begin{aligned}
G(\tilde{B}) &= A\left[\alpha^2 F_3 - 4\alpha F_2 + 6aF_1\right] \\
&= A\left[\left(\frac{2F_2}{F_3}\right)^2 F_3 - 4\frac{2F_2}{F_3}F_2 + 6\frac{4F_2^2 + A^{-1}F_3 g}{6F_1F_3}F_1\right] \\
&= A\left[\frac{4F_2^2}{F_3} - \frac{8F_2^2}{F_3} + \frac{4F_2^2}{F_3} + A^{-1}g\right] = g.
\end{aligned}$$

Therefore, function G is bijective by classes. ■

B Proof of proposition 1

Proof. Let $\tilde{B} \in C(\tilde{\mathbf{B}})$ such that $\tilde{B} = (a, f_1, f_2, f_3, \alpha; w)$, and $s \in \mathbb{R}$ then:

$$\tilde{B}s = (as, f_1, f_2, f_3, \alpha s; w).$$

Using the definition of G we have that:

$$\begin{aligned}
G(\tilde{B}s) &= A\left[(\alpha s)^2 F_3 - 4(\alpha s)F_2 + 6(as)F_1\right] \\
& \quad A\left[((\alpha(s^* + \epsilon))^2 F_3 - 4\alpha(s^* + \epsilon)F_2 + 6a(s^* + \epsilon)F_1\right] \\
& \quad A\left[(\alpha s^*)^2 F_3 + 2\alpha^2 s^* \epsilon F_3 + (\alpha\epsilon)^2 F_3 - 4\alpha s^* F_2 - 4\alpha\epsilon F_2\right. \\
& \quad \left. + 6as^* F_1 + 6a\epsilon F_1\right] \\
&= A\left[(\alpha s^*)^2 F_3 - 4\alpha s^* F_2 + 6as^* F_1\right] \\
& \quad + A\left[2\alpha^2 s^* \epsilon F_3 + (\alpha\epsilon)^2 F_3 - 4\alpha\epsilon F_2 + 6a\epsilon F_1\right].
\end{aligned}$$

Therefore:

$$\begin{aligned}
\lim_{\epsilon \to 0} G(\tilde{B}s) &= \lim_{\epsilon \to 0} A\left[(\alpha s^*)^2 F_3 - 4\alpha s^* F_2 + 6as^* F_1\right] \\
& \quad + \lim_{\epsilon \to 0} A\left[2\alpha^2 s^* \epsilon F_3 + (\alpha\epsilon)^2 F_3 - 4\alpha\epsilon F_2 + 6a\epsilon F_1\right] \\
&= G(\tilde{B}s^*). \ \blacksquare
\end{aligned}$$

C Proof of proposition 2

Proof. It will be demonstrated by mathematical induction.
First it will be proven for $k = 2$:

$$\begin{aligned}
&G(\tilde{B}_1 s_1 + \tilde{B}_2 s_2)\\
&= A\left[(s_1\alpha_1 + s_2\alpha_2)^2 F_3 - 4(s_1\alpha_1 + s_2\alpha_2)F_2 + 6(s_1 a_1 + s_2 a_2)F_1\right]\\
&= A\left[((s_1^*\alpha_1)^2 + 2s_1^*\alpha_1 s_2^*\alpha_2 + (s_2^*\alpha_2)^2)F_3\right.\\
&\quad \left. - 4(s_1^*\alpha_1 + s_2^*\alpha_2)F_2 + 6(s_1^* a_1 + s_2^* a_2)\right]\\
&= A\left[(s_1^*\alpha_1 + s_2^*\alpha_2)^2 F_3 - 4(s_1^*\alpha_1 + s_2^*\alpha_2)F_2\right.\\
&\quad \left. +6(s_1^* a_1 + s_2^* a_2)F_1\right] + A[H(\epsilon_1, \epsilon_2)]
\end{aligned}$$

where:

$$\begin{aligned}
&H(\epsilon_1, \epsilon_2)\\
&= (2\epsilon_1 s_1^*\alpha_1 + (\epsilon_1\alpha_1)^2 + 2\alpha_1\alpha_2(s_1^*\epsilon_2 + s_2^*\epsilon_1 + \epsilon_1\epsilon_2) + 2\epsilon_2 s_2^*\alpha_2\\
&\quad +(\epsilon_2\alpha_2)^2)F_3 - 4(\epsilon_1\alpha_1 + \epsilon_2\alpha_2)F_2 + 6(\epsilon_1 a_1 + \epsilon_2 a_2).
\end{aligned}$$

Therefore:

$$\begin{aligned}
&\lim_{\{\epsilon_1,\epsilon_2\}\to 0} G(\tilde{B}_1 s_1 + \tilde{B}_2 s_2)\\
&= \lim_{\{\epsilon_1,\epsilon_2\}\to 0} A\left[(s_1^*\alpha_1 + s_2^*\alpha_2)^2 F_3 - 4(s_1^*\alpha_1 + s_2^*\alpha_2)F_2\right.\\
&\quad \left. +6(s_1^* a_1 + s_2^* a_2)F_1\right] + \lim_{\{\epsilon_1,\epsilon_2\}\to 0} A[H(\epsilon_1, \epsilon_2)]\\
&= G(\tilde{B}_1 s_1^* + \tilde{B}_2 s_2^*).
\end{aligned}$$

Using mathematical induction, it is assumed true for $k = n - 1$ and will be proven for $k = n$.

Note that $\forall i = 1, 2, \ldots, k \leq n - 1$ it is true that:

$$G\left(\sum_{i=1}^{k}(\tilde{B}_i s_i)^2\right) = G\left(\sum_{i=1}^{k}(\tilde{B}_i s_i^*)^2\right) + A\left[H(\{\epsilon_i\}_{i=1}^{k})\right]$$

such that

$$\lim_{\{\epsilon_i\}\to 0} H\left(\{\epsilon_i\}_{i=1}^{k}\right) = 0.$$

For $k = n$, we have:

$$\text{let } \tilde{B} = \sum_{i=1}^{n} \tilde{B}_i s_i, \text{ such that } \tilde{B}_i \in C(\tilde{\mathbf{B}}), i = 1, 2, \ldots, n$$

It can be seen that:

1. $\tilde{B} = \sum_{i=1}^{n} \tilde{B}_i s_i = \sum_{i=1}^{n-1} \tilde{B}_i s_i + \tilde{B}_n s_n = \tilde{B}^{'}_{n-1} + \tilde{B}_n s_n,$
2. $\alpha = \sum_{i=1}^{n-1} (\alpha_i s_i) + \alpha_n s_n = (\alpha^{'}_{n-1}) + \alpha_n s_n,$
3. $a = \sum_{i=1}^{n-1} (a_i s_i) + a_n s_n = (a^{'}_{n-1}) + a_n s_n.$

Therefore:

$$
\begin{aligned}
G\left(\sum_{i=1}^{n} \tilde{B}_i s_i\right) = & \\
& A\Big[(\alpha^{'}_{n-1} + s_n\alpha_n)^2 F_3 - 4(\alpha^{'}_{n-1} + s_n\alpha_n)F_2 + \\
& + 6(a^{'}_{n-1} + s_n a_n)F_1 + H(\{\epsilon_i\}_{i=1}^{n-1})\Big] \\
= \quad & A\Big[(\alpha^{'}_{n-1} + (s_n^* + \epsilon_n)\alpha_n)^2 F_3 - 4(\alpha^{'}_{n-1} + (s_n^* + \epsilon_n)\alpha_n)F_2 + \\
& + 6(a^{'}_{n-1} + s_n a_n)F_1 + H(\{\epsilon_i\}_{i=1}^{n-1})\Big] \\
= \quad & A\left[\left((\alpha^{'}_{n-1})^2 + (s_n^*\alpha_n)^2 + 2\alpha^{'}_{n-1} s_n^* \alpha_n\right) F_3 - 4\left(\alpha^{'}_{n-1} + s_n^*\alpha_n\right) F_2 + \right. \\
& \left. + 6\left((a^{'}_{n-1} + s_n^* a_n)F_1 + H(\{\epsilon_i\}_{i=1}^{n-1}\right) + H(\epsilon_n)\right] \\
= \quad & A\left[\left(\sum_{i=1}^{n} (\alpha_i s_i^*)^2\right) F3 - 4\left(\sum_{i=1}^{n} (\alpha_i s_i^*)\right) F_2 + \right. \\
& \left. + \left(\sum_{i=1}^{n} a_i s_i^*\right) F_1 + H(\{\epsilon_i\}_{i=1}^{n-1}) + H(\epsilon_n)\right]
\end{aligned}
$$

where:

$$H(\epsilon_n) = \epsilon_n(2s_n^*\alpha_n^2 + 2\alpha^{'}_{n-1}\alpha_n + \alpha_n^2\epsilon_n + \alpha_n F_2 + a_n F_1).$$

Hence:

$$
\begin{aligned}
& \lim_{\{\epsilon_i\}_{i=1}^{n} \to 0} G\left(\sum_{i=1}^{n} \tilde{B}_i s_i\right) \\
= \quad & \lim_{\{\epsilon_i\}_{i=1}^{n} \to 0} A\left[\left(\sum_{i=1}^{n} (\alpha_i s_i^*)^2\right) F3 - 4\left(\sum_{i=1}^{n} (\alpha_i s_i^*)\right) F_2 + \right. \\
& \left. + \left(\sum_{i=1}^{n} a_i s_i^*\right) F_1 + H(\{\epsilon_i\}_{i=1}^{n-1}) + H(\epsilon_n)\right] \\
= \quad & G\left(\sum_{i=1}^{n} \tilde{B}_i s_i^*\right). \ \blacksquare
\end{aligned}
$$

D Proof of proposition 3

Proof. The proof will be done by contradiction.
Suppose:

1. $y^* = (y_1^*, y_2^*, \ldots, y_k^*) \in$ PF
2. $y^* \notin$ SC-F

Then we have:

1. $y^* \in Image(h)$, and y^* is non dominated.
2. $\exists\ \epsilon > 0,$ such that if $||y^* - y|| \leq \epsilon \Rightarrow y \in Image(h).$

consider a point: $y_0 = (y_1^* - \frac{\epsilon}{\sqrt{k}}, y_2^* - \frac{\epsilon}{\sqrt{k}}, \ldots, y_k^* - \frac{\epsilon}{\sqrt{k}})$. Note that:

$$||y^* - y_0|| = \sqrt{(\frac{\epsilon}{\sqrt{k}})^2 + \ldots + (\frac{\epsilon}{\sqrt{k}})^2} = \epsilon,$$

for that reason $y_0 \in Image(h)$ and y_0 dominates to y^*, which is a contradiction, therefore: PF $\subseteq$ SC-F. ■

Acknowledgements

The authors wish to thank the three anonymous referees since their comments helped to considerably improve the present work.
S.G. de C. would also like to thank D.Sto. and to P.V.Gpe. for their inspiration, and to Ma, Ser, Mon, Chema, and to his Flaquita for all their support

References

[1] Aragon, V.S.; Coello, C.A.C. (2010) "A modified version of a t-cell algorithm for constrained optimization problems", *Int. J. Numer. Methods Eng.* **84**(3): 351–378.

[2] Belegundu, A.D. (1982) *A Study of Mathematical Programming Methods for Structural Optimization*. Ph.D. Thesis, Department of Civil Environmental Engineering, University of Iowa, Iowa City.

[3] Coello, C.A.C. (2000) "Use of a self-adaptive penalty approach for engineering optimization problems", *Comput. Ind.* **41**(2): 113–127.

[4] Deb, K. (1991) "Optimal design of a welded beam via genetic algorithms", *AIAA J.* **29**(11): 2013–2015.

[5] de-los-Cobos-Silva, S.G. (2015) "SC: system of convergence. Theory and fundaments", *Revista de Matemática: Teoría y Aplicaciones* **22**(2): 341–367.

[6] Dubois, D.; Prade, H. (1978) "Operations on fuzzy numbers", *International Journal of Systems Science* **9**(6): 613–626.

[7] Elsayed, S.M.; Sarker, R.A.; Essam, D.L. (2012) "On an evolutionary approach for constrained optimization problem solving", *Applied Soft Computing* **12**(10): 3208–3227.

[8] Gavana, A. (2007?) "Global optimization benchmarks and AMPGO. Test functions", in: `http://infinity77.net/global_optimization/`

[9] Glover, F. (1989) "Tabu search, Part I", *ORSA Journal on Computing* **1**(3): 190–206.

[10] Glover, F. (1998) "A template for scatter search and path relink", in: J.K. Hao, E. Lutton, E. Ronald, M. Schoenauer & D. Snyers (Eds.) *Artificial Evolution*, Lecture Notes on Computer Science 1363, Springer, Berlin: 1–51.

[11] He, Q.; Wang, L. (2007a) "An effective co-evolutionary particle swarm optimization for constrained engineering design problems", *Eng. Appl. Artif. Intell.* **20**(1): 89–99.

[12] He, Q.; Wang, L. (2007b) "A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization", *Appl. Math. Comput.* **186**(2): 1407–1422.

[13] Hedar, A.R. (2007?) "Global optimization test problems", in: `http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/go.htm`

[14] Hibbeler, R.C. (2000) *Mechanics of Materials*, 8th ed. Prentice Hall, New Jersey.

[15] Holland, J.H. (1975) *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor MI.

[16] Imran, M.; Hashima, R.; Khalid, N.E.A. (2013) "An overview of particle swarm optimization variants", *Procedia Engineering* **53**: 491–496.

[17] Kalami Heris, S.M. (s.f.) "S. Mostapha Kalami Heris' homepage", `http://www.kalami.ir`

[18] Kannan, B.K.; Kramer, S.N. (1994) "An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design", *J. Mech. Des.* **116**(2): 405–411.

[19] Kennedy, J.; Eberhart, R. (1995) "Particle swarm optimization", in: *Proc. of the IEEE International Conference on Neural Networks*, vol. 4: 1942–1948.

[20] Kennedy, J.; Eberhart, R.C.; Shi, Y. (2001) *Swarm Intelligence*. Morgan Kaufmann, San Francisco.

[21] Kirkpatrick, S.; Gelatt, C.; Vecchi, M. (1983) "Optimization by simulated annealing", *Science* **220**: 671–680.

[22] Liang, J.J.; Runarsson, T.P.; Mezura-Montes, E.; Clerc, M.; Suganthan, P.N.; Coello Coello, C.A.; Deb, K. (2006) "Problem definitions and evaluation criteria for the CEC 2006, special session on constrained real-parameter optimization", Technical Report, *IEEE Congress on Evolutionary Computation*, 24 pp.

[23] Liou, T.S.; Wang, M.J.J. (1992) "Ranking fuzzy numbers with integral value", *Fuzzy Sets and Systems* **50**(3): 247–255.

[24] Liu, C.A. (2007) "New multiobjective PSO algorithm for nonlinear constrained programming problems", in: R. Wang, E. Shen E. & F. Gu (Eds.) *Advances in Cognitive Neurodynamics ICCN*: 955-962.

[25] Liu, H.; Cai, Z.; Wang, Y. (2010) "Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization", *Appl. Soft Comput.* **10**(2): 629–640.

[26] Lu, H.; Chen, W. (2008) "Self-adaptive velocity particle swarm optimization for solving constrained optimization problems", *J. Glob. Opt.* **41**(3): 427–445.

[27] Mazhoud, I.; Hadj-Hamou, K.; Bigeon J.; Joyeux P. (2013) "Particle swarm optimization for solving engineering problems: a new constraint-handling mechanism", *Eng. Appl. of Art. Intel.* **26**(4): 1263–1273.

[28] Mezura-Montes, E.; Velázquez-Reyes, J.V.; Coello-Coello, C.A. (2006) "Modified differential evolution for constrained optimization", in: *IEEE Congress on Evolutionary Computation*: 25–32.

[29] Mezura-Montes, E.; Miranda-Varela, M.E.; Gómez-Ramón, R.C. (2010) "Differential evolution in constrained numerical optimization: An empirical study", *Inf. Sci.* **180**(22): 4223–4262.

[30] Rao, S.S. (1996) *Engineering Optimization: Theory and Practice*. Wiley, New York.

[31] Sedighizadeh, D.; Masehian, E. (2009) "Particle swarm optimization methods, taxonomy and applications", *Int. Journal of Computer Theory and Engineering* **1**(5): 486–502.

[32] Tessema, B.; Yen, G.G. (2009) "An adaptive penalty formulation for constrained evolutionary optimization", *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* **39**(3): 565–578.

[33] Toscano-Pulido, G.; Coello, C.A.C. (2004) "A constraint-handling mechanism for particle swarm optimization", in: *IEEE Congress on Evolutionary Computation*, vol. 2, IEEE Press: 1396–1403.

[34] Zitzler, E.; Deb, K.; Thiele, L. (2000) "Comparison of multiobjective evolutionary algorithm: empirical results", *Evolutionary Computation* **8**(2): 173–195

[35] Wang, Y.; Cai, Z.; Zhou, Y. (2009) "Accelerating adaptive trade-off model using shrinking space technique for constrained evolutionary optimization", *Int. J. Numer. Methods Eng.* **77**(11): 1501–1534.