

A Constraint-Handling Mechanism for Particle Swarm Optimization

Gregorio Toscano Pulido and Carlos A. Coello Coello
Evolutionary Computation Group (EVOCINV)
CINVESTAV-IPN
Computer Science Section
Electrical Engineering Department
México, D.F., 07300, MEXICO
Email: gtoscano@computacion.cs.cinvestav.mx
ccoello@cs.cinvestav.mx

Abstract— This paper presents a simple mechanism to handle constraints with a particle swarm optimization algorithm. Our proposal uses a simple criterion based on closeness of a particle to the feasible region in order to select a leader. Additionally, our algorithm incorporates a turbulence operator that improves the exploratory capabilities of our particle swarm optimization algorithm. Despite its relative simplicity, our comparison of results indicates that the proposed approach is highly competitive with respect to three constraint-handling techniques representative of the state-of-the-art in the area.

I. INTRODUCTION

The Particle Swarm Optimization (PSO) algorithm has become increasingly popular in the last few years, mainly in numerical optimization tasks [8]. However, PSO, like other evolutionary algorithms, lacks an explicit mechanism to incorporate constraints. Remarkably, there has been little work related to the incorporation of constraints into the PSO algorithm, despite the fact that most real-world applications have constraints. In previous work, some researchers have assumed the possibility of being able to generate in a random way feasible solutions to feed the population of a PSO algorithm [6], [7]. The problem with this approach is that it may have a very high computational cost in some cases. For example, in some of the test functions used in this paper, even the generation of one million of random points was insufficient to produce a single feasible solution. Evidently, such a high computational cost turns out to be prohibitive in real-world applications.

When incorporating constraints into the fitness function of an evolutionary algorithm, it is particularly important to maintain diversity in the population and to be able to keep solutions both inside and outside the feasible region [2], [10]. Several studies have shown that, despite their popularity, traditional (external) penalty functions, even when used with dynamic penalty factors, tend to have difficulties to deal with highly constrained search spaces and with problems in which the constraints are active in the optimum [2], [9], [16]. Motivated by this fact, a number of constraint-handling techniques have been proposed for evolutionary algorithms [12], [2]. However, this topic has been only scarcely explored by PSO researchers

[6], [1], [13], [7], [14].

This paper proposes a constraint-handling mechanism that consists of two main components: a turbulence operator (i.e., some form of mutation intended to improve the exploratory capabilities of PSO), and a simple decision-making scheme based on closeness to the feasible region to choose a leader when dealing with constrained search spaces.

The remainder of this paper is organized as follows. In Section II, we introduce the problem of interest to us. Section III introduces our proposed approach, including a description of the mechanism used to handle constraints and the turbulence operator adopted. In Section IV, we describe the 13 test functions adopted to validate our proposed approach. Finally, our conclusions and some possible paths of future research are provided in Section VI.

II. BASIC CONCEPTS

We are interested in the general nonlinear programming problem in which we want to:

$$\text{Find } \vec{x} \text{ which optimizes } f(\vec{x}) \quad (1)$$

subject to:

$$g_i(\vec{x}) \leq 0, \quad i = 1, \dots, n \quad (2)$$

$$h_j(\vec{x}) = 0, \quad j = 1, \dots, p \quad (3)$$

where \vec{x} is the vector of solutions $\vec{x} = [x_1, x_2, \dots, x_r]^T$, n is the number of inequality constraints and p is the number of equality constraints (in both cases, constraints could be linear or nonlinear). It is common practice in the specialized literature on evolutionary optimization to transform equality constraints into inequalities of the form:

$$|h_j(\vec{x})| - \epsilon \leq 0 \quad (4)$$

where ϵ is the tolerance allowed (a very small value). This allows us to deal only with inequality constraints. An analogous transformation is possible to deal only with equality constraints. However, this sort of transformation is uncommon

```

function CPSO Algorithm
Begin
  For Each particle
    1. Initialize
    2. Compute fitness value
    3.  $p_{best}$ =fitness value
  EndFor
  Do
    4. Choose the particle with the
       best fitness value in the population.
       Call it  $g_{best}$ 
    For each particle
      If the fitness value is better than
       $p_{best}$  then
        5.  $p_{best}$ =fitness value
      EndIF
    6. Perform the flight
    7. Add Turbulence
    8. Update fitness value
  EndFor
  While Stopping condition not satisfied
End.

```

Fig. 1. Pseudocode of the PSO algorithm adopted.

since, when using evolutionary algorithms, it tends to be easier to handle inequality constraints (using, for example exterior penalty functions [15]) than equality constraints.

If we denote with \mathcal{F} to the feasible region and with \mathcal{S} to the whole search space, then it should be clear that $\mathcal{F} \subseteq \mathcal{S}$. For an inequality constraint that satisfies $g_i(\vec{x}) = 0$, then we will say that is active at \vec{x} . All equality constraints h_j (regardless of the value of \vec{x} used) are considered active at all points of \mathcal{F} .

III. OUR PROPOSED APPROACH

Figure 1 shows the PSO algorithm adopted for our study.

The algorithm is basically a simple PSO implementation, except for three aspects: the way in which the velocity is computed, the turbulence operator and the mechanism adopted to handle constraints. These aspects are discussed in the following subsections.

A. Computing Velocity

For computing the velocity of a particle, we used the expression proposed in [17]:

$$V_{id} = w \times V_{id} + c_1 \times rand_1() \times (p_{best,id} - x_{id}) + c_2 \times rand_2() \times (g_{best,id} - x_{id}) \quad (5)$$

(6)

where V_{id} is the velocity of the id dimension, c_1 and c_2 are two values randomly generated in the range $[1.5, 2.5]$ (this range was empirically derived), $rand_1()$ and $rand_2()$

refer to functions that return a random value within the range $[0.0, 1.0]$, w is the inertia weight, which in our case takes a value randomly generated within the range $[0.1, 0.5]$ (this range was empirically derived), p_{best} is the best position of the current particle found so far and g_{best} is the best position of the best particle found so far.

B. The Turbulence Operator

The turbulence consists of an alteration to the flight velocity of a particle.¹ This modification is performed in all the dimensions (i.e., in all the decision variables), such that the particle can move to a completely isolated region (something much more difficult to achieve by the mere use of the velocity adjustment formula described before). This mechanism aims to perturb the swarm as to avoid that the particles get trapped in local optima. The turbulence operator acts based on a probability that considers the current generation and the total number of iterations to be performed. The idea is to have a much higher probability to perturb the flight of the particles at the beginning of the search. Over time, this probability will be decreased as we progress in the search.

The turbulence can be seen as a mutation operator and it is based on the following expression:

$$temp = current_generation / total_generations$$

$$prob_{turbulence} = temp^{1.7} - 2.0 * (temp) + 1.0 \quad (7)$$

(8)

where $temp$ is used as a temporary variable, $current_generation$ is the current generation number, $total_generations$ is the total number of generations and $prob_{turbulence}$ refers to the probability of affecting the flight of a particle using the turbulence operator. The values used for this expression were empirically derived after a set of experiments.

C. Mechanism to Handle Constraints

The mechanism that we propose in this paper to handle constraints² is applied when selecting a leader. What we did was to perform a small change in the fitness function such that if we compare two feasible particles, the particle that has the highest fitness value wins. If one of the particles is infeasible and the other one is feasible, then the feasible particle wins. If both particles compared are infeasible, then the particle that has the lowest value in its total violation of constraints (normalized with respect to the largest violation of each constraint achieved by any particle in the current population) wins. The idea is to choose as a leader to the particle that, even when infeasible, lies closer to the feasible region. To understand better this idea, let's consider the following example:

Let's consider 3 particles and 2 constraints: **particle 1** violates in 30 units the first constraint and in 40 units the second

¹This mechanism is inspired on [4].

²Note that all constraints will be treated as "hard", since their satisfaction will be considered imperative. No "soft" constraints are considered in the research reported in this paper [11].

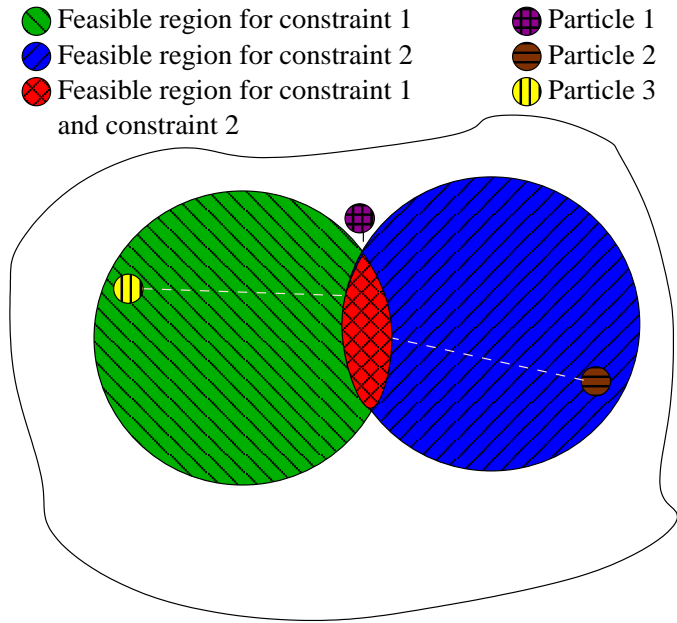


Fig. 2. Graphical representation of our example that explains the constraint-handling mechanism incorporated into our PSO algorithm.

constraint. **Particle 2** does not violate the first constraint, but it violates in 100 units the second constraint. Finally, **particle 3** violates in 130 units the first constraint, but it does not violate the second constraint. Furthermore, with respect to the total population, the largest violation of the first constraint is 200 and the largest violation of the second constraint is 120. Thus, the fitness of **particle 1** is $30/200 + 40/120 = 0.48333$. The fitness of **particle 2** is $0 + 100/120 = 0.83333$. The fitness of **particle 3** is $130/200 + 0 = 0.65000$. So, **particle 1** has a better fitness than **particle 2** and **particle 3** (let's keep in mind that in this case, a smaller value indicates that the particle is closer to the feasible region), despite the fact that this particle violated the 2 constraints of the problem and the two other particles only violate one of them. This behavior is graphically depicted in Figure III-C. Note that there may be ill-defined constraints in which the scheme adopted in this paper may not properly work. However, we will not deal with those cases in this paper.

IV. TEST FUNCTIONS

To evaluate the performance of the proposed approach we used the 13 test functions described in [16]. The test functions chosen contain characteristics that are representative of what can be considered “difficult” global optimization problems for an evolutionary algorithm. Their expressions are provided next.

1) **g01**:

Minimize: $f(\vec{x}) = 5 \sum_{i=1}^4 x_i - 5 \sum_{i=1}^4 x_i^2 - \sum_{i=5}^{13} x_i$
subject to:

$$g_1(\vec{x}) = 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \leq 0$$

$$g_2(\vec{x}) = 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \leq 0$$

$$g_3(\vec{x}) = 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \leq 0$$

$$g_4(\vec{x}) = -8x_1 + x_{10} \leq 0$$

$$g_5(\vec{x}) = -8x_2 + x_{11} \leq 0$$

$$g_6(\vec{x}) = -8x_3 + x_{12} \leq 0$$

$$g_7(\vec{x}) = -2x_4 - x_5 + x_{10} \leq 0$$

$$g_8(\vec{x}) = -2x_6 - x_7 + x_{11} \leq 0$$

$$g_9(\vec{x}) = -2x_8 - x_9 + x_{12} \leq 0$$

where the bounds are $0 \leq x_i \leq 1$ ($i = 1, \dots, 9$), $0 \leq x_i \leq 100$ ($i = 10, 11, 12$) and $0 \leq x_{13} \leq 1$. The global optimum is at $x^* = (1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 1)$ where $f(x^*) = -15$. Constraints g_1, g_2, g_3, g_4, g_5 and g_6 are active.

2) **g02**:

Maximize: $f(\vec{x}) = \left| \frac{\sum_{i=1}^n \cos^4(x_i) - 2 \prod_{i=1}^n \cos^2(x_i)}{\sqrt{\sum_{i=1}^n ix_i^2}} \right|$ sub-
ject to:

$$g_1(\vec{x}) = 0.75 - \prod_{i=1}^n x_i \leq 0$$

$$g_2(\vec{x}) = \sum_{i=1}^n x_i - 7.5n \leq 0$$

where $n = 20$ and $0 \leq x_i \leq 10$ ($i = 1, \dots, n$). The global maximum is unknown; the best reported solution is [16] $f(x^*) = 0.803619$. Constraint g_1 is close to being active ($g_1 = -10^{-8}$).

3) **g03**:

Maximize: $f(\vec{x}) = (\sqrt{n})^n \prod_{i=1}^n x_i$

subject to:

$$h(\vec{x}) = \sum_{i=1}^n x_i^2 - 1 = 0$$

where $n = 10$ and $0 \leq x_i \leq 1$ ($i = 1, \dots, n$).
The global maximum is at $x_i^* = 1/\sqrt{n}$ ($i = 1, \dots, n$)
where $f(x^*) = 1$.

4) **g04:**

$$\text{Minimize: } f(\vec{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$$

subject to:

$$\begin{aligned} g_1(\vec{x}) &= 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 - 92 \leq 0 \\ g_2(\vec{x}) &= -85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 + 0.0022053x_3x_5 \leq 0 \\ g_3(\vec{x}) &= 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 - 110 \leq 0 \\ g_4(\vec{x}) &= -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 - 0.0021813x_3^2 + 90 \leq 0 \\ g_5(\vec{x}) &= 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 - 25 \leq 0 \\ g_6(\vec{x}) &= -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \leq 0 \end{aligned}$$

where: $78 \leq x_1 \leq 102$, $33 \leq x_2 \leq 45$,
 $27 \leq x_i \leq 45$ ($i = 3, 4, 5$). The optimum solution is
 $x^* = (78, 33, 29.995256025682, 45, 36.775812905788)$
where $f(x^*) = -30665.539$. Constraints g_1 y g_6 are active.

5) **g05**

$$\text{Minimize: } f(\vec{x}) = 3x_1 + 0.000001x_1^3 + 2x_2 + (0.000002/3)x_2^3$$

subject to:

$$\begin{aligned} g_1(\vec{x}) &= -x_4 + x_3 - 0.55 \leq 0 \\ g_2(\vec{x}) &= -x_3 + x_4 - 0.55 \leq 0 \\ h_3(\vec{x}) &= 1000 \sin(-x_3 - 0.25) + 1000 \sin(-x_4 - 0.25) + 894.8 - x_1 = 0 \\ h_4(\vec{x}) &= 1000 \sin(x_3 - 0.25) + 1000 \sin(x_3 - x_4 - 0.25) + 894.8 - x_2 = 0 \\ h_5(\vec{x}) &= 1000 \sin(x_4 - 0.25) + 1000 \sin(x_4 - x_3 - 0.25) + 1294.8 = 0 \end{aligned}$$

where $0 \leq x_1 \leq 1200$, $0 \leq x_2 \leq 1200$, $-0.55 \leq x_3 \leq 0.55$, and $-0.55 \leq x_4 \leq 0.55$. The best known solution is $x^* = (679.9453, 1026.067, 0.1188764, -0.3962336)$ where $f(x^*) = 5126.4981$.

6) **g06**

$$\text{Minimize: } f(\vec{x}) = (x_1 - 10)^3 + (x_2 - 20)^3$$

subject to:

$$\begin{aligned} g_1(\vec{x}) &= -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \leq 0 \\ g_2(\vec{x}) &= (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \leq 0 \end{aligned}$$

where $13 \leq x_1 \leq 100$ and $0 \leq x_2 \leq 100$. The optimum solution is $x^* = (14.095, 0.84296)$ where $f(x^*) = -6961.81388$. Both constraints are active.

7) **g07**

$$\text{Minimize: } f(\vec{x}) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45$$

subject to:

$$\begin{aligned} g_1(\vec{x}) &= -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \leq 0 \\ g_2(\vec{x}) &= 10x_1 - 8x_2 - 17x_7 + 2x_8 \leq 0 \\ g_3(\vec{x}) &= -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \leq 0 \\ g_4(\vec{x}) &= 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \leq 0 \\ g_5(\vec{x}) &= 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \leq 0 \\ g_6(\vec{x}) &= x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \leq 0 \\ g_7(\vec{x}) &= 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 - 30 \leq 0 \\ g_8(\vec{x}) &= -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \leq 0 \end{aligned}$$

where $-10 \leq x_i \leq 10$ ($i = 1, \dots, 10$). The global optimum is $x^* = (2.171996, 2.363683, 8.773926, 5.095984, 0.9906548, 1.430574, 1.321644, 9.828726, 8.280092, 8.375927)$ where $f(x^*) = 24.3062091$. Constraints g_1, g_2, g_3, g_4, g_5 and g_6 are active.

8) **g08**

$$\text{Maximize: } f(\vec{x}) = \frac{\sin^3(2\pi x_1) \sin(2\pi x_2)}{x_1^2(x_1 + x_2)}$$

subject to:

$$\begin{aligned} g_1(\vec{x}) &= x_1^2 - x_2 + 1 \leq 0 \\ g_2(\vec{x}) &= 1 - x_1 + (x_2 - 4)^2 \leq 0 \end{aligned}$$

where $0 \leq x_1 \leq 10$ and $0 \leq x_2 \leq 10$. The optimum solution is located at $x^* = (1.2279713, 4.2453733)$ where $f(x^*) = 0.095825$.

9) **g09**

$$\text{Minimize: } f(\vec{x}) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7$$

subject to:

$$\begin{aligned} g_1(\vec{x}) &= -127 + 2x_1^2 + 3x_4^2 + x_3 + 4x_4^2 + 5x_5 \leq 0 \\ g_2(\vec{x}) &= -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \leq 0 \\ g_3(\vec{x}) &= -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \leq 0 \\ g_4(\vec{x}) &= 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \leq 0 \end{aligned}$$

where $-10 \leq x_i \leq 10$ ($i = 1, \dots, 7$). The global optimum is $x^* = (2.330499, 1.951372, -0.4775414, 4.365726, -0.6244870, 1.038131, 1.594227)$ where $f(x^*) = 680.6300573$. Two constraints are active (g_1 and g_4).

10) **g10**

$$\text{Minimize: } f(\vec{x}) = x_1 + x_2 + x_3$$

$$\text{subject to: } g_1(\vec{x}) = -1 + 0.0025(x_4 + x_6) \leq 0$$

$g_2(\vec{x}) = -1 + 0.0025(x_5 + x_7 - x_4) \leq 0$
 $g_3(\vec{x}) = -1 + 0.01(x_8 - x_5) \leq 0$
 $g_4(\vec{x}) = -x_1x_6 + 833.33252x_4 + 100x_1 - 83333.333 \leq 0$
 $g_5(\vec{x}) = -x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \leq 0$
 $g_6(\vec{x}) = -x_3x_8 + 1250000 + x_3x_5 - 2500x_5 \leq 0$
 where $100 \leq x_1 \leq 10000$, $1000 \leq x_i \leq 10000$,
 $(i = 2, 3)$, $10 \leq x_i \leq 1000$, $(i = 4, \dots, 8)$. The global
 optimum is: $x^* = (579.19, 1360.13, 5109.92,$
 $182.0174, 295.5985, 217.9799, 286.40, 395.5979)$,
 where $f(x^*) = 7049.25$. g_1 , g_2 and g_3 are active.

11) **g11**

Minimize: $f(\vec{x}) = x_1^2 + (x_2 - 1)^2$
 subject to:
 $h(\vec{x}) = x_2 - x_1^2 = 0$

where: $-1 \leq x_1 \leq 1$, $-1 \leq x_2 \leq 1$. The optimum
 solution is $x^* = (\pm 1/\sqrt{2}, 1/2)$ where $f(x^*) = 0.75$.

12) **g12**

Maximize: $f(\vec{x}) = \frac{100 - (x_1 - 5)^2 - (x_2 - 5)^2 - (x_3 - 5)^2}{100}$
 subject to:
 $g_1(\vec{x}) = (x_1 - p)^2 + (x_2 - q)^2 + (x_3 - r)^2 - 0.0625 \leq 0$

where $0 \leq x_i \leq 10$ ($i = 1, 2, 3$) and
 $p, q, r = 1, 2, \dots, 9$. The feasible region of the
 search space consists of 9^3 disjointed spheres. A
 point (x_1, x_2, x_3) is feasible if and only if there exist
 p, q, r such the above inequality (12) holds. The global
 optimum is located at $x^* = (5, 5, 5)$ where $f(x^*) = 1$.

13) **g13**

Minimize: $f(\vec{x}) = e^{x_1x_2x_3x_4x_5}$
 subject to:
 $h_1(\vec{x}) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0$
 $h_2(\vec{x}) = x_2x_3 - 5x_4x_5 = 0$
 $h_3(\vec{x}) = x_1^3 + x_2^3 + 1 = 0$ where $-2.3 \leq$
 $x_i \leq 2.3$ ($i = 1, 2$) and $-3.2 \leq x_i \leq 3.2$
 $(i = 3, 4, 5)$. The optimum solution is $x^* =$
 $(-1.717143, 1.595709, 1.827247, -0.7636413,$
 $-0.763645)$ where $f(x^*) = 0.0539498$.

To get an estimate of the ratio between the feasible and the
 infeasible region of each of these problems, a ρ metric (as
 suggested by Michalewicz & Schoenauer [12]) was computed
 using the following expression: $\rho = |F|/|S|$ where $|F|$ is the
 number of feasible solutions and $|S|$ is the total number of
 solutions randomly generated. In this work, $S = 1,000,000$
 random solutions.

The different values of ρ for each of the functions chosen are
 shown in Table I, where n is the number of decision variables,
 LI is the number of linear inequalities, NI the number of
 nonlinear inequalities, LE is the number of linear equalities
 and NE is the number of nonlinear equalities.

We evaluated the performance of our PSO algorithm using
 the turbulence operator and the constraint-handling mechanism
 described before. We performed 30 independent runs of our
 approach for each test function, and we compared our results
 with respect to three constraint-handling techniques that are
 representative of the state-of-the-art in the area: Stochastic
 Ranking (SR) [16], the Homomorphous Maps (HM) [9], and
 the Adaptive Segregational Constraint Handling Evolutionary
 Algorithm (ASCHEA) [5].

The Homomorphous Maps [9] perform a homomorphous
 mapping between an n -dimensional cube and the feasible
 search region (either convex or non-convex). The main idea
 of this approach is to transform the original problem into
 another (topologically equivalent) function that is easier to
 optimize by an evolutionary algorithm (a genetic algorithm
 in this case). Both, the Stochastic Ranking and ASCHEA are
 based on a penalty function approach. Stochastic Ranking [16]
 sorts the individuals in the population in order to assign them
 a rank value. However, based on the value of a user-defined
 parameter, the comparison between two adjacent solutions will
 be performed using only the objective function. The remaining
 comparisons will be performed using only the penalty value
 (the sum of constraint violation). ASCHEA [5] uses three
 combined mechanisms: (1) an adaptive penalty function, (2) a
 constraint-driven recombination that forces to select a feasible
 individual to recombine it with an infeasible one and (3) a
 segregational selection based on feasibility which maintains
 a balance between feasible and infeasible solutions in the
 population. ASCHEA also requires a niching mechanism to
 improve the diversity in the population. Each mechanism
 requires the definition of several user-defined parameters.

Our comparison of results with respect to the three ap-
 proaches previously described is presented in Tables II, III
 and IV.

Comparing our PSO approach with respect to the Homo-
 morphous Maps (see Table II), our technique was able to
 improve the ‘‘best’’ results in several problems (remarkably
 including g05, which could not be solved by the homomor-
 phous maps). In the remaining test functions, there is practi-
 cally a match between our approach and the homomorphous
 maps. Regarding average and worst results, our approach
 is better than the homomorphous maps in several problems
 (remarkably, in g05 and g06). No comparisons were made
 with function g13 because such results were not available for
 HM.

With respect to Stochastic Ranking (see Table III), our
 approach was able to match most of their ‘‘best’’ results.
 However, stochastic ranking found slightly better results in
 some problems (remarkably in g10 and g13, which are some
 of the most difficult test functions from this benchmark). The
 average and worst results of stochastic ranking are also better
 than those of our approach in some problems.

Compared against the Adaptive Segregational Constraint
 Handling Evolutionary Algorithm (see Table IV), our approach

Problem	n	Function	ρ	LI	NI	LE	NE
g01	13	quadratic	0.0003%	9	0	0	0
g02	20	nonlinear	99.9973%	1	1	0	0
g03	10	nonlinear	0.0026%	0	0	0	1
g04	5	quadratic	27.0079%	0	6	0	0
g05	4	nonlinear	0.0000%	2	0	0	3
g06	2	nonlinear	0.0057%	0	2	0	0
g07	10	quadratic	0.0000%	3	5	0	0
g08	2	nonlinear	0.8581%	0	2	0	0
g09	7	nonlinear	0.5199%	0	4	0	0
g10	8	linear	0.0020%	3	3	0	0
g11	2	quadratic	0.0973%	0	0	0	1
g12	3	quadratic	4.7697%	0	9 ³	0	0
g13	5	nonlinear	0.0000%	0	0	1	2

TABLE I
VALUES OF ρ FOR THE 13 TEST PROBLEMS CHOSEN.

Problem	Optimal	Best Result		Mean Result		Worst Result	
		PSO	HM	PSO	HM	PSO	HM
g01	-15.000000	-15.000000	-14.788600	-15.000000	-14.708200	-15.000000	-14.615400
g02	0.803619	0.803432	0.799530	0.790406	0.796710	0.750393	0.791190
g03	1.000000	1.004720	0.999700	1.003814	0.998900	1.002490	0.997800
g04	-30665.539000	-30665.500000	-30664.500000	-30665.500000	-30655.300000	-30665.500000	-30645.900000
g05	5126.498000	5126.640000	NA	5461.081333	NA	6104.750000	NA
g06	-6961.814000	-6961.810000	-6952.100000	-6961.810000	-6342.600000	-6961.810000	-5473.900000
g07	24.306000	24.351100	24.620000	25.355771	24.826000	27.316800	25.069000
g08	0.095825	0.095825	0.095825	0.095825	0.089157	0.095825	0.029144
g09	680.630000	680.638000	680.910000	680.852393	681.160000	681.553000	683.180000
g10	7049.330700	7057.590000	7147.990000	7560.047857	8163.600000	8104.310000	9659.300000
g11	0.750000	0.749999	0.750000	0.750107	0.750000	0.752885	0.750000
g12	1.000000	1.000000	1.000000	1.000000	0.999135	1.000000	0.991950
g13	0.053950	0.068665	NA	1.716426	NA	13.669500	NA

TABLE II
COMPARISON OF OUR PSO ALGORITHM WITH RESPECT TO THE HOMOMORPHOUS MAPS (HM) [9].

Problem	Optimal	Best Result		Mean Result		Worst Result	
		PSO	SR	PSO	SR	PSO	SR
g01	-15.000000	-15.000000	-15.000000	-15.000000	-15.000000	-15.000000	-15.000000
g02	0.803619	0.803432	0.803515	0.790406	0.781975	0.750393	0.726288
g03	1.000000	1.004720	1.000000	1.003814	1.000000	1.002490	1.000000
g04	-30665.539000	-30665.500000	-30665.539000	-30665.500000	-30665.539000	-30665.500000	-30665.539000
g05	5126.498000	5126.640000	5126.497000	5461.081333	5128.881000	6104.750000	5142.472000
g06	-6961.814000	-6961.810000	-6961.814000	-6961.810000	-6875.940000	-6961.810000	-6350.262000
g07	24.306000	24.351100	24.307000	25.355771	24.374000	27.316800	24.642000
g08	0.095825	0.095825	0.095825	0.095825	0.095825	0.095825	0.095825
g09	680.630000	680.638000	680.630000	680.852393	680.656000	681.553000	680.763000
g10	7049.330700	7057.590000	7054.316000	7560.047857	7559.192000	8104.310000	8835.655000
g11	0.750000	0.749999	0.750000	0.750107	0.750000	0.752885	0.750000
g12	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
g13	0.053950	0.068665	0.053957	1.716426	0.057006	13.669500	0.216915

TABLE III
COMPARISON OF RESULTS OF OUR PSO WITH RESPECT TO STOCHASTIC RANKING (SR) [16].

Problem	Best Result			Mean Result		Worst Result	
	Optimal	PSO	ASCHEA	PSO	ASCHEA	PSO	ASCHEA
g01	-15.000000	-15.000000	-15.000000	-15.000000	-14.840000	-15.000000	NA
g02	0.803619	0.803432	0.785000	0.790406	0.590000	0.750393	NA
g03	1.000000	1.004720	1.000000	1.003814	0.999890	1.002490	NA
g04	-30665.539000	-30665.500000	30665.500000	-30665.500000	30665.500000	-30665.500000	NA
g05	5126.498000	5126.640000	5126.500000	5461.081333	5141.650000	6104.750000	NA
g06	-6961.814000	-6961.810000	-6961.810000	-6961.810000	-6961.810000	-6961.810000	NA
g07	24.306000	24.351100	24.332300	25.355771	24.660000	27.316800	NA
g08	0.095825	0.095825	0.095825	0.095825	0.095825	0.095825	NA
g09	680.630000	680.638000	680.630000	680.852393	680.641000	681.553000	NA
g10	7049.330700	7057.590000	7061.130000	7560.047857	7193.110000	8104.310000	NA
g11	0.750000	0.749999	0.750000	0.750107	0.750000	0.752885	NA
g12	1.000000	1.000000	NA	1.000000	NA	1.000000	NA
g13	0.053950	0.068665	NA	1.716426	NA	13.669500	NA

TABLE IV

COMPARISON OF RESULTS OF OUR PSO WITH RESPECT TO THE ADAPTIVE SEGREGATIONAL CONSTRAINT HANDLING EVOLUTIONARY ALGORITHM (ASCHEA) [5].

was able to improve the “best” results in several problems (remarkably, in g02 and g10). ASCHEA produced slightly better results only in g03, g05, g07 and g11. We did not compare the worst results because they were not available for ASCHEA. We did not perform comparisons with respect to ASCHEA using functions g12 and g13 for the same reason.

As we can see, our approach showed a very competitive performance with respect to these three state-of-the-art approaches.

Our approach can deal with moderately constrained problems (g04), highly constrained problems, problems with low (g06, g08), moderated (g09) and high (g01, g02, g03, g07) dimensionality, with different types of combined constraints (linear, nonlinear, equality and inequality) and with very large (g02), very small (g05 and g13) or even disjoint (g12) feasible regions. Also, the algorithm is able to deal with large search spaces (based on the intervals of the decision variables) with a very small feasible region (g10). Furthermore, the approach can find the global optimum in problems where such optimum lies on the boundaries of the feasible region (g01, g02, g04, g06, g07, g09).

Note that our approach does not require any parameters. In contrast, the homomorphous maps require an additional parameter (called v) which has to be found empirically [9]. Stochastic ranking requires the definition of a parameter called P_f , whose value has an important impact on the performance of the approach [16]. ASCHEA also requires the definition of several extra parameters, and in its latest version, it uses niching [3], which is a process that also has at least one additional parameter [5].

The computational cost measured in the number of evaluations of the objective function (FFE) performed by our approach is lower than the other techniques with respect to which it was compared. Our approach performed (in all the test problems shown) 340,000 FFE (we used 40 particles running for 8500 generations), the Stochastic Ranking per-

formed 350,000 FFE, the Homomorphous Maps performed 1,400,000 FFE, and ASCHEA required 1,500,000 FFE.

VI. CONCLUSIONS AND FUTURE WORK

We have presented a relatively simple constraint-handling mechanism for choosing leaders in the particle swarm optimization algorithm. This mechanism is combined with a turbulence operator responsible for improving the exploratory capabilities of the PSO algorithm.

The proposed approach does not use any special mechanism to deal with constrained search spaces in which the global optimum lies on the boundaries between the feasible and the infeasible regions, despite the fact that such type of problems are the main target of the most competitive constraint-handling techniques proposed in the specialized literature [16], [5], [9]. Additionally, our PSO approach does not require any user-defined parameters and it performs less objective evaluations than any of the other approaches with respect to which it was compared.

Despite all of the above reasons, the results obtained by our approach are highly competitive, and in some cases, even improve on the results obtained by much more elaborate approaches such as the Homomorphous Maps [9] and ASCHEA [5].

As part of our future work, we plan to experiment with the use of neighborhoods and other mechanisms to improve the exploratory capabilities of PSO, mainly when dealing with highly constrained search spaces. We also intend to study alternative mechanisms to accelerate convergence while keeping the same quality of the results achieved in this paper. Furthermore, we are interested in studying online or self-adaptation mechanisms that allow our PSO approach to stop without having to pre-define a certain number of iterations (as done, for example, with our microGA² [18]). Such type of approach may be particularly useful for real-world applications.

ACKNOWLEDGMENTS

The first author acknowledges support from the Mexican Consejo Nacional de Ciencia y Tecnología (CONACyT) through a scholarship to pursue graduate studies at CINVESTAV-IPN's. The second author acknowledges support from CONACyT through project number 34201-A.

REFERENCES

- [1] G. Coath and S. K. Halgamuge. A Comparison of Constraint-Handling Methods for the Application of Particle Swarm Optimization to Constrained Nonlinear Optimization Problems. In *Proceedings of the Congress on Evolutionary Computation 2003 (CEC'2003)*, volume 4, pages 2419–2425, Piscataway, New Jersey, December 2003. Canberra, Australia, IEEE Service Center.
- [2] C. A. Coello Coello. Theoretical and Numerical Constraint Handling Techniques used with Evolutionary Algorithms: A Survey of the State of the Art. *Computer Methods in Applied Mechanics and Engineering*, 191(11-12):1245–1287, January 2002.
- [3] K. Deb and D. E. Goldberg. An Investigation of Niche and Species Formation in Genetic Function Optimization. In J. D. Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 42–50, San Mateo, California, June 1989. George Mason University, Morgan Kaufmann Publishers.
- [4] J. E. Fieldsend and S. Singh. A Multi-Objective Algorithm based upon Particle Swarm Optimisation, an Efficient Data Structure and Turbulence. In *Proceedings of the 2002 U.K. Workshop on Computational Intelligence*, pages 37–44, Birmingham, UK, September 2002.
- [5] S. B. Hamida and M. Schoenauer. ASCHEA: New Results Using Adaptive Segregational Constraint Handling. In *Proceedings of the Congress on Evolutionary Computation 2002 (CEC'2002)*, volume 1, pages 884–889, Piscataway, New Jersey, May 2002. IEEE Service Center.
- [6] X. Hu and R. Eberhart. Solving Constrained Nonlinear Optimization Problems with Particle Swarm Optimization. In *Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2002)*, volume 5, Orlando, USA, IIS, July 2002.
- [7] X. Hu, R. C. Eberhart, and Y. Shi. Engineering Optimization with Particle Swarm. In *Proceedings of the 2003 IEEE Swarm Intelligence Symposium*, pages 53–57. Indianapolis, Indiana, USA, IEEE Service Center, April 2003.
- [8] J. Kennedy and R. C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann, UK, 2001.
- [9] S. Koziel and Z. Michalewicz. Evolutionary Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization. *Evolutionary Computation*, 7(1):19–44, 1999.
- [10] E. Mezura-Montes and C. A. Coello Coello. Adding a Diversity Mechanism to a Simple Evolution Strategy to Solve Constrained Optimization Problems. In *Proceedings of the Congress on Evolutionary Computation 2003 (CEC'2003)*, volume 1, pages 6–13, Piscataway, New Jersey, December 2003. Canberra, Australia, IEEE Service Center.
- [11] Z. Michalewicz and D. B. Fogel. *How to Solve It: Modern Heuristics*. Springer, Berlin, 2000.
- [12] Z. Michalewicz and M. Schoenauer. Evolutionary Algorithms for Constrained Parameter Optimization Problems. *Evolutionary Computation*, 4(1):1–32, 1996.
- [13] U. Paquet and A. P. Engelbrecht. A New Particle Swarm Optimiser for Linearly Constrained Optimization. In *Proceedings of the Congress on Evolutionary Computation 2003 (CEC'2003)*, volume 1, pages 227–233, Piscataway, New Jersey, December 2003. Canberra, Australia, IEEE Service Center.
- [14] K. Parsopoulos and M. Vrahatis. Particle Swarm Optimization Method for Constrained Optimization Problems. In P. Sincak, J. Vascak, V. Kvasnicka, and J. Pospicha, editors, *Intelligent Technologies - Theory and Applications: New Trends in Intelligent Technologies*, pages 214–220. IOS Press, 2002. Frontiers in Artificial Intelligence and Applications series, Vol. 76 ISBN: 1-58603-256-9.
- [15] J. T. Richardson, M. R. Palmer, G. Liepins, and M. Hilliard. Some Guidelines for Genetic Algorithms with Penalty Functions. In J. D. Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms (ICGA-89)*, pages 191–197, San Mateo, California, June 1989. George Mason University, Morgan Kaufmann Publishers.
- [16] T. P. Runarsson and X. Yao. Stochastic Ranking for Constrained Evolutionary Optimization. *IEEE Transactions on Evolutionary Computation*, 4(3):284–294, September 2000.
- [17] Y. Shi and R. Eberhart. A Modified Particle Swarm Optimizer. In *Proceedings of the IEEE International Conference on Evolutionary Computation*, pages 69–73, Piscataway, NJ, 1998. IEEE Press.
- [18] G. Toscano Pulido and C. A. Coello Coello. The Micro Genetic Algorithm 2: Towards Online Adaptation in Evolutionary Multiobjective Optimization. In C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele, editors, *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*, pages 252–266, Faro, Portugal, April 2003. Springer. Lecture Notes in Computer Science. Volume 2632.