Improved Particle Swarm Optimization in Constrained Numerical Search Spaces

Efrén Mezura-Montes and Jorge Isacc Flores-Mendoza

Abstract This chapter presents a study about the behavior of Particle Swarm Optimization (PSO) in constrained search spaces. A comparison of four well-known PSO variants used to solve a set of test problems is presented. Based on the information obtained, the most competitive PSO variant is detected. From this preliminary analysis, the performance of this variant is improved with two simple modifications related with the dynamic control of some parameters and a variation in the constraint-handling technique. These changes keep the simplicity of PSO i.e. no extra parameters, mechanisms controlled by the user or combination of PSO variants are added. This Improved PSO (IPSO) is extensively compared against the original PSO variants, based on the quality and consistency of the final results and also on two performance measures and convergence graphs to analyze their on-line behavior. Finally, IPSO is compared against some state-of-the-art PSO-based approaches for constrained optimization. Statistical tests are used in the experiments in order to add support to the findings and conclusions established.

1 Introduction

Nowadays, it is common to find complex problems to be solved in diverse areas of human life. Optimization problems can be considered among them. Different sources of difficulty can be associated in their resolution e.g. a very high number of possible solutions (very large search spaces), hard-to-satisfy constraints and a high nonlinearity. Mathematical Programming (MP) offers a set of techniques to solve

Efrén Mezura-Montes
Laboratorio Nacional de Informática Avanzada (LANIA A.C.), Rébsamen 80, Centro, Xalapa, Veracruz, 91000, México, e-mail: emezura@lania.mx

Jorge Isacc Flores-Mendoza
Laboratorio Nacional de Informática Avanzada (LANIA A.C.), Rébsamen 80, Centro, Xalapa, Veracruz, 91000, México, e-mail: jflores@lania.edu.mx

different type of problems like numerical, discrete or combinatorial optimization problems. This chapter focuses only on numerical (continuous) optimization problems. MP techniques are always the first option to solve optimization problems. In fact, they provide, under some specific conditions to be accomplished by the problem, convergence to the global optimum solution. However, for some real-world problems, MP techniques are either difficult to apply (i.e. a problem transformation may be required), they cannot be applied or they get trapped in local optimum solutions. Based on the aforementioned, the use of heuristics to solve optimization problems has become very popular in different areas. Tabu Search [11], Simulated Annealing [16] and Scatter Search [12] are examples of successful heuristics commonly used by interested practitioners and researchers to solve difficult search problems. There is also a set of nature-inspired heuristics designed for optimization problem-solving and they comprise the area of Bio-inspired optimization. Two main groups of algorithms can be distinguished: (1) Evolutionary algorithms (EAs) [7] and (2) Swarm Intelligence algorithms (SIAs) [9]. EAs are based on the theory of evolution and the survival of the fittest. A set of complete solutions of a problem are represented and evolved by means of variation operators and selection and replacement processes. There are three main paradigms in this area: (1) Evolutionary Programming [10], Evolution Strategies [37] and Genetic Algorithms [14]. There are other important EAs proposed such as Genetic Programming [17] where solutions are represented by means of nonlinear structures like trees and its aim is oriented to symbolic optimization and Differential Evolution [34], designed to solve numerical optimization problems by using vector differences as search directions coupled with an EA framework.

On the other hand, SIAs emulate different social and cooperative behaviors found in animals or insects. The two original paradigms are the following: (1) Particle Swarm Optimization (PSO) [15] and (2) Ant Colony Optimization (ACO) [6]. PSO is based on the cooperative behavior of bird flocks, whereas ACO models social behaviors of ants e.g. the foraging behavior as to solve mainly combinatorial optimization problems.

These Bio-Inspired Algorithms (BIAs), such as genetic algorithms, evolutionary programming, evolution strategies, differential evolution and particle swarm optimization, share some features. They work with a set of complete solutions for the problem (usually generated at random). These solutions are evaluated in order to obtain a quality measure, i.e. fitness value, for each one of them. A selection mechanism is then implemented as to select those solutions with a better fitness value. These best solutions will be utilized to generate new solutions by using variation operators. Finally, a replacement process occurs, where the size of the population (which was increased) is trimmed as to always maintain a fixed population size.

In their original versions, BIAs are designed to solve unconstrained optimization problems. Then, there is a considerable amount of research dedicated to designing constraint-handling techniques to be added to BIAs. There are some classifications for constraint-handling techniques based on the way they incorporate feasibility information in the quality of a given solution [4, 30]. For the purpose of this chapter, a simple taxonomy is proposed, because the main goal of the current study is not to design a novel constraint-handling mechanism. Instead, the aim is to propose the analysis of the behavior of a BIA (PSO in this case) as a first step in designing a competitive approach to solve Constrained Numerical Optimization Problems (CNOPs). As a result, the simplicity of PSO is maintained i.e. no additional mechanisms and/or parameters controlled by the user are considered.

This chapter is organized as follows: Section 2 contains the statement of the problem of interest and some useful optimization concepts. Section 3 introduces PSO in more detail, considering its main elements and variants. A brief introduction to constraint-handling techniques is summarized in Section 4. In Section 5, the approaches which use PSO to solve CNOPs are detailed and discussed. Section 6 presents the empirical comparison of PSO variants and a discussion of results. After that, Section 7 details the modifications made to the most competitive PSO variant obtained from the previous study, all of them in order to improve its performance when solving CNOPs. An in-depth study of the behavior of this novel PSO and a comparison against state-of-the-art PSO-based approaches to solve CNOPs are presented in Section 8. The chapter ends with a conclusion and a discussion of future work in Section 9.

2 Constrained Optimization Problems

The optimization process consists of finding the best solution for a given problem under certain conditions. As it was mentioned before, this chapter will only consider numerical optimization problems in presence of constraints. Without loss of generality a CNOP can be defined as to:

Find $\mathbf{x}$ which minimizes

$$f(\mathbf{x}) \tag{1}$$

subject to

$$g_i(\mathbf{x}) \leq 0, \;\; i = 1, \ldots, m \tag{2}$$

$$h_j(\mathbf{x}) = 0, \;\; j = 1, \ldots, p \tag{3}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the vector of solutions $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ and each x_i, $i = 1, ..., n$ is bounded by lower and upper limits $L_i \leq x_i \leq U_i$ which define the search space $\mathscr{S}$, $\mathscr{F}$ comprises the set of all solutions which satisfy the constraints of the problems and it is called the feasible region; m is the number of inequality constraints and p is the number of equality constraints (in both cases, constraints could be linear or nonlinear). Equality constraints are transformed into inequalities constraints as follows: $\left|h_j(\mathbf{x})\right| - \varepsilon \leq 0$, where ε is the tolerance allowed (a very small value).

As multiobjective concepts will be used later in the chapter, the multiobjective optimization problem will be also introduced. Without loss of generality, a Multiobjective Optimization Problem (MOP) is defined as:

Find $\mathbf{x}$ which minimizes

$$f(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_k(\mathbf{x})]^T \tag{4}$$

subject to

$$g_i(\mathbf{x}) \ \leq \ 0, \ i = 1,..,m \tag{5}$$

$$h_j(\mathbf{x}) \ = \ 0, \ j = 1,...,p \tag{6}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the vector of solutions $\mathbf{x} = [x_1, x_2, ..., x_n]^T$ and each x_i, $i = 1,...,n$ is bounded by lower and upper limits $L_i \leq x_i \leq U_i$ which define the search space $\mathscr{S}$, $\mathscr{F}$ is the feasible region; m is the number of inequality constraints and p is the number of equality constraints (in both cases, constraints could be linear or nonlinear).

A vector $\mathbf{u}$= $(u_1,...,u_k)$ is said to dominate another vector $\mathbf{v}$ = $(v_1,...,v_k)$ (denoted by $\mathbf{u} \preceq \mathbf{v}$) if and only if $\mathbf{u}$ is partially less than $\mathbf{v}$, i.e. $\forall i \in \{1,...,k\}$, $u_i \leq v_i \wedge \exists i \in \{1,...k\} : u_i < v_i$.

3 Particle Swarm Optimization

Kennedy and Eberhart [15] proposed PSO, which is based on the social behavior of bird flocks. Each individual "i", called particle, represents a solution to the optimization problem i.e. a vector of decision variables $\mathbf{x}_i$. The particle with the best fitness value is considered the leader of the swarm (population of particles), and guides the other members to promising areas of the search space. Each particle is influenced on its search direction by cognitive (i.e. its own best position found so far, called $\mathbf{x}_{pbest_i}$) and social (i.e. the position of the leader of the swarm named$\mathbf{x}_{gBest}$) information. At each iteration (generation) of the process, the leader of the swarm is updated. These two elements: $\mathbf{x}_{pbest_i}$ and $\mathbf{x}_{gBest}$, besides the current position of particle "i" $\mathbf{x}_i$, are used to calculate its new velocity $\mathbf{v}_i(t+1)$ based on its current velocity $\mathbf{v}_i(t)$ (search direction) as follows:

$$\mathbf{v}_i(t+1) = \mathbf{v}_i(t) + c_1 r_1 (\mathbf{x}_{pbest_i} - \mathbf{x}_i) + c_2 r_2 (\mathbf{x}_{gBest} - \mathbf{x}_i). \tag{7}$$

where c_1 and c_2 are acceleration constants to control the influence of the cognitive and social information respectively and r_1, r_2 are random real numbers between 0 and 1 generated with an uniform distribution.

After each particle updates its corresponding velocity, the flight formula is used to update its position:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1). \quad (8)$$

where $\mathbf{x}_i$(t) is the current position of the particle, $\mathbf{x}_i(t+1)$is the new position of this particle and $\mathbf{v}_i(t+1)$ is its recently updated velocity (search direction).

Based on Equation 7, two main different approaches have been proposed to update the velocity of a particle. The aim is to improve the usefulness of the search direction generated and to avoid premature convergence: (1) PSO with inertia weight and (2) PSO with constriction factor.

3.1 PSO with Inertia Weight

Proposed by Shi and Eberhart [38], the inertia weight was added to the velocity update formula (Equation 7) as a mechanism to control PSO's exploration and exploitation capabilities. Its goal is to control the influence of the previous velocity of a given particle. The inertia weight is represented by w and scales the value of the current velocity $\mathbf{v}_i(t)$ of particle “i”. A small inertia weight value promotes local exploration, whereas a high value promotes global exploration. Shi and Eberhart [38] suggested w=0.8 when using PSO to solve unconstrained optimization problems. The modified formula to update the velocity of a particle by using the inertia weight value is the following:

$$\mathbf{v}_i(t+1) = w\mathbf{v}_i(t) + c_1 r_1(\mathbf{x}_{pbest_i} - \mathbf{x}_i) + c_2 r_2(\mathbf{x}_{gBest} - \mathbf{x}_i) \quad (9)$$

3.2 PSO with Constriction Factor

With the aim of eliminating velocity clamping and encouraging convergence, Clerc and Kennedy [3] proposed, instead of a inertia weight value, a constriction coefficient. This constriction factor is represented by k. Unlike the inertia weight, the constriction factor affects all values involved in the velocity update as follows:

$$\mathbf{v}_i(t+1) = k[\mathbf{v}_i(t) + c_1 r_1(\mathbf{x}_{pbest_i} - \mathbf{x}_i) + c_2 r_2(\mathbf{x}_{gBest} - \mathbf{x}_i)] \quad (10)$$

3.3 Social Network Structures

There are two basic PSO variants depending of the social network structure used [9]: (1) global best and (2) local best PSO. In the global best variant the star social structure allows each particle to communicate with all the remaining particles in the swarm, whereas in the local best PSO, the ring social structure allows each particle to communicate only with those particles in its neighborhood. Therefore, in the global best PSO, there is a unique leader of the swarm. On the other hand, in local best PSO, there is a leader for each neighborhood. There are differences expected in the behavior of these two PSO variants due to the way particles communicate among themselves. In global best PSO a faster convergence is promoted as the probability of being trapped in local optima is increased. However, in local best PSO, a slower convergence usually occurs while a better exploration of the search space is encouraged.

A pseudocode for the global best PSO is presented in Figure 1.

Begin
GEN = 0
Generate a swarm of random solutions ($\mathbf{x}_i$), $i = 1, 2, \ldots, SWARM\_SIZE$.
Initialize **for each particle**, $\mathbf{x}_{pbest_i} = \mathbf{x}_i$, and $\mathbf{v}_i(t) = 0$.
Evaluate the fitness of each particle in the swarm.
Do
Select the leader ($\mathbf{x}_{gBest}$) of the swarm.
For each particle, update its velocity with (7).
For each particle, update its position with (8).
Evaluate the fitness of the new position **for each particle**.
Update the $\mathbf{x}_{pbest_i}$ (memory) value **for each particle**.
GEN=GEN+1
Until GEN = Gmax
End

Fig. 1 Global best PSO pseudocode.

A pseudocode for the local best PSO is presented in Figure 2.

4 Constraint-Handling

As it was mentioned in the introduction to the chapter, EAs and SIAs were originally designed to solve unconstrained optimization problems. Constraint-handling techniques are required to add feasibility information in the fitness calculation of a solution [4, 30]. Roughly, constraint-handling techniques can be divided in two groups:

1. Those based on the fitness penalization of a solution i.e. a combination of the objective function value (Equation 1) and the sum of constraint violation (Equations 2 and 3) .
2. Those based on the separated use of the objective function value (Equation 1) and the sum of constraint violation (Equations 2 and 3) in the fitness value of a solution.

In the first group penalty functions are considered, which is in fact the most popular constraint-handling mechanism. They transform a constrained problem into

```
Begin
    GEN = 0
    Generate a swarm of random solutions (x_i) i = 1, 2, ..., SWARM_SIZE.
    Divide the swarm in n neighborhoods.
    Assign equal number of particles to each neighborhood.
    Initialize for each particle, x_pbest_i = x_i, and v_i(t) = 0.
    Do
        Evaluate the fitness of the particle in each neighborhood.
        Select the leader (x_lBest_i) of each neighborhood.
        For each particle, update its velocity with (7).
            by using the corresponding leader of each neighborhood x_lBest_i
        For each particle, update its position with (8).
        Evaluate the fitness of the new position for each particle.
        Update the x_pbest_i (memory) value for each particle.
        GEN=GEN+1
    Until GEN= Gmax
End
```

Fig. 2 Local best PSO pseudocode.

an unconstrained problem by punishing, i.e. decreasing, the fitness value of infeasible solutions in such a way that feasible solutions are preferred in the selection/replacement processes. However, an important drawback is the definition of penalty factor values, which determine the severity of the penalization. If the penalty is too low, the feasible region may never be reached. On the other hand, if the penalty is too high, the feasible region will be reached so fast, mostly at random and the probability of getting trapped in local optimum might be very high [4].

The second group includes constraint-handling techniques based on Deb's feasibility rules [5], Stochastic Ranking [35], multiobjective concepts [28], lexicographic ordering [36], the α-constrained method [40], Superiority of Feasible points [33] among others.

Different search engines have been used on the above mentioned approaches: Genetic Algorithms [5, 28], Evolution Strategies [35], Differential Evolution [40]. However, to the best of the authors' knowledge, the research usually focuses on adapting a constraint-handling mechanism to a given search engine, but the studies to analyze the performance of a search engine in constrained search spaces are scarce [29].

5 Related Work

This section presents PSO-based approaches proposed to solve CNOPs. Toscano and Coello [42] proposed a global best PSO with inertia weight coupled with a turbulence (mutation) operator, which affects the velocity vector of a particle as follows: $\mathbf{v}_i = \mathbf{v}_j^{\Phi}(t) + r_3$, where $\mathbf{v}_i$ is the current velocity of particle i, $\mathbf{v}_j^{\Phi}(t)$ is the current velocity of its nearest neighbor and r_3 is a random value. The use of this turbulence operator is calculated with a dynamic adaptation approach. The idea is to use more of the turbulence operator in the first part of the search. The constraint-handling technique used was a group-2 approach [5].

Parsopoulos and Vrahatis [32] used their Unified Particle Swarm Optimization (UPSO) to solve CNOPs. The UPSO combines the exploration and exploitation abilities of two basic PSO variants (local best and global best together, both with constriction factor). The scheme of UPSO is the following: A weighted sum of the two velocity values (from the local and global variants) is computed, where a parameter ($0 \leq u \leq 1$) represents the unification factor and controls the influence of each variant in the final search direction. Finally a typical flight formula with this unified velocity is used to update the position of a particle. A group-1 constraint-handling technique i.e. static penalty function, was used in this approach where the number of violated constraints as well as the amount of constraint violation were taken into account.

Liang and Suganthan [23] proposed a PSO-based approach to solve CNOPs by using dynamic multi-swarms (DMS-PSO). The DMS-PSO was implemented using a local best PSO with inertia weight, where the size and particles of the sub-swarms change periodically. Two concepts are modified from the original PSO in DMS-PSO: (1) Instead of just keeping the best position found so far, all the best positions reached for a particle are recorded to improve the global search and (2) a local search mechanism, i.e. sequential quadratic programming method, was added. The constraints of the problems are dynamically assigned, based on the difficulty to be satisfied for each sub-swarm. Moreover, one sub-swarm will optimize the objective function. As the function and constraints are handled separately, they use a group-2 constraint-handling. The sequential quadratic programming method is applied to the pbest values (not to the current positions of the particles) as to improve them.

Li, Tian and Kong [21] solved CNOPs by coupling an inertia weight local best PSO with a mutation strategy. Their constraint-handling mechanism belongs to group 2 and was based on the superiority of feasible points [33]. The mutation strategy used a diversity metric for population diversity control and for convergence improvement. When the population diversity was low (based on a defined value), the swarm is expanded through the mutation strategy. This mutation strategy consisted of a random perturbation applied to the particles in the swarm. Li, Tian and Min [22] used a similar constraint-handling mechanism, but without mutation strategy and using instead a global best PSO variant to solve Bilevel Programming Problem (BLPP).

Lu and Chen [25] implemented a group-2 constraint-handling technique by using a global best PSO with inertia weight and velocity restriction. The original problem (CNOP) is transformed into a bi-objective problem using a Dynamic-Objective Strategy (DOM). DOM consists of the following: if a particle is infeasible, its unique objective is to enter the feasible region. On the other hand, if the particle is feasible its unique objective is now to optimize the original objective function. This process is dynamically adjusted according to the feasibility of the particle. The bi-objective problem is defined as: minimize $F(\mathbf{x}) = (\phi(\mathbf{x}), f(\mathbf{x}))$. $\phi(\mathbf{x})$ is the sum of constraint violations and $f(\mathbf{x})$ is the original function objective. Based on the feasibility of $\mathbf{x}_{gBest}$ and $\mathbf{x}_{pbest_i}$, the values of important parameters like c_1 and c_2 are defined to promote feasible particles to remain feasible. The formula to update the velocity is modified in such a way that the positions of the pbest and gbest are mixed in the search direction defined.

Cagnina, Esquivel and Coello [2] used a group-2 constraint-handling technique, Deb's rules [5], in a combination of global-local best PSO particle swarm optimizer to solve CNOPs. The velocity update formula and also the flight formula are changed as to include information of the global and local best leaders and to use a Gaussian distribution to get the new position for the particle, respectively. Furthermore, a dynamic mutation operator is added for diversity promotion in the swarm.

Wei and Wang [43] presented a global best PSO with inertia weight which transformed the problem, as in [25], into a bi-objective problem (group-2 constraint-handling). The original objective function was the second objective and the first one was the degree of constraint violation: $\min(\delta(\mathbf{x}), f(\mathbf{x}))$. Deb's feasibility rules were used as selection criteria. A new three-parent crossover operator (TPCO) is also added to the PSO. Finally, a dynamic adaptation for the inertia weight value was included to encourage a correct balance between global and local search.

Krohling and dos Santos Coelho [18] proposed a global best PSO with constriction factor and a co-evolutionary approach to solve CNOPs. This problem is transformed into a min-max problem. The Lagrange-based method (group-1 constraint-handling) is used to formulate the problem in terms of a min-max problem. Two swarms are used: The first one moves in the space defined by the variables of the problem, whereas the second swarm optimizes the Lagrange multipliers.

He, Prempain and Wu [13] proposed a 'fly-back' mechanism added to a global best PSO with inertia weight to solve CNOPs. In their approach, the authors also solved mixed (i.e. continuous-discrete) optimization problems. Discrete variables were handled by a truncation mechanism. The initial swarm must be always located in the feasible region of the search space, which may be a disadvantage when dealing with problems with a very small feasible region. The 'fly-back' mechanism keeps particles from flying out of the feasible region by discarding those flights which generate infeasible solutions. Then, the velocity value is reduced and a new flight is computed.

Based on the related work, some interesting modifications were found regarding PSO for solving CNOPs: (1) Mutation, crossover operators or even local search are added to PSO to promote diversity in the swarm [2, 21, 23, 42, 43], (2) there is a tendency to mix global and local best PSO variants into a single one [2, 32], (3) the original CNOP is transformed into a multiobjective problem [23, 25, 43], and finally, (4) the original velocity update and flight formulas are modified [2, 25].

6 Motivation and Empirical Comparison

Unlike the previous research, the motivation of this work is two-fold: (1) to acquire more knowledge about the behavior of PSO in its original variants when solving CNOPs and (2) after considering this knowledge as a first step of design, to propose simple modifications to PSO in order to get a competitive approach to solve CNOPs by maintaining PSO's simplicity.

In this section, two original PSO variants (inertia weight and constriction factor) combined with two social network structures (star and ring) are compared. In the remaining of this chapter, each combination of variant-social network will be called as variant. They are selected based on the following criteria:

- They are the most used in the approaches reported in the specialized literature on numerical constrained optimization (Section 5).
- As mentioned in the beginning of this Section, the motivation of this work is to acquire knowledge about the behavior of PSO in its original variants i.e. variants without additional mechanisms.

The four variants are: (1) global best PSO with inertia weight, (2) global best PSO with constriction factor, (3) local best PSO with inertia weight and (4) local best PSO with constriction factor.

In order to promote a fair analysis of the four PSO variants and not add extra parameters to be fine-tuned, a group-2 (objective function and constraints handled separately) parameter-free constraint-handling technique is chosen for all the variants. This technique consists of a set of three feasibility rules proposed by Deb [5]. They are the following: (1) If two solutions are feasible, the one with the best value of the objective function is preferred, (2) if one solution is feasible and the other one is infeasible, the feasible one is preferred and (3) if two solutions are infeasible, the one with the lowest normalized sum of constraint violation is preferred.

24 test problems (all minimization problems) were taken from the specialized literature [24] and used to test the performance of the four PSO variants. These problems are an extension of the well-known benchmark used to test BIAs in constrained search spaces. In fact, these problems were used to evaluate state-of-the-art approaches in the IEEE Congress on Evolutionary Computation (CEC 2006). Details of the problems can be found in [24]. A summary of their features can be found in Table 1.

As can be noted, the problems have different characteristics such as dimensionality, type of objective function, type and number of constraints and active constraints at the optimum (i.e. the solution lies in the boundaries between the feasible and infeasible regions). Therefore, they present different challenges to the algorithms tested.

This first experiment is designed as follows: 30 independent runs were computed per PSO variant per test problem. Statistical results (best, mean and standard deviation) were calculated from the final results. They are presented, for the first twelve problems in Table 2 and for the last twelve in Table 3. The parameters used in this experiment are the following: 80 particles and 2000 generations (160,000 total evaluations), $c_1 = 2.7$ and $c_2 = 2.5$ for all PSO variants. For the two local

Table 1 Details of the 24 test problems. "*n*" is the number of decision variables, $\rho = |F|/|S|$ is the estimated ratio between the feasible region and the search space, LI is the number of linear inequality constraints, NI the number of nonlinear inequality constraints, LE is the number of linear equality constraints and NE is the number of nonlinear equality constraints. *a* is the number of active constraints at the optimum.

Prob.	***n***	**Type of function**	ρ	**LI**	**NI**	**LE**	**NE**	***a***
g01	13	quadratic	0.0111%	9	0	0	0	6
g02	20	nonlinear	99.9971%	0	2	0	0	1
g03	10	polynomial	0.0000%	0	0	0	1	1
g04	5	quadratic	52.1230%	0	6	0	0	2
g05	4	cubic	0.0000%	2	0	0	3	3
g06	2	cubic	0.0066%	0	2	0	0	2
g07	10	quadratic	0.0003%	3	5	0	0	6
g08	2	nonlinear	0.8560%	0	2	0	0	0
g09	7	polynomial	0.5121%	0	4	0	0	2
g10	8	linear	0.0010%	3	3	0	0	6
g11	2	quadratic	0.0000%	0	0	0	1	1
g12	3	quadratic	4.7713%	0	1	0	0	0
g13	5	nonlinear	0.0000%	0	0	0	3	3
g14	10	nonlinear	0.0000%	0	0	3	0	3
g15	3	quadratic	0.0000%	0	0	1	1	2
g16	5	nonlinear	0.0204%	4	34	0	0	4
g17	6	nonlinear	0.0000%	0	0	0	4	4
g18	9	quadratic	0.0000%	0	12	0	0	6
g19	15	nonlinear	33.4761%	0	5	0	0	0
g20	24	linear	0.0000%	0	6	2	12	16
g21	7	linear	0.0000%	0	1	0	5	6
g22	22	linear	0.0000%	0	1	8	11	19
g23	9	linear	0.0000%	0	2	3	1	6
g24	2	linear	79.6556%	0	2	0	0	2

best variants 8 neighborhoods were used, $w = 0.7$ for both inertia weight variants and $k = 0.729$ [3] for both constriction factor variants. The tolerance for equality constraints was set to ε =0.0001 for all variants.

These parameter values were defined by a trial and error process. The population size was varied from low values (40) to higher values (120), however no improvement was reported. c_1 and c_2 values required unusually higher values to provide competitive results. w and k values were taken as recommended in previous research [39, 38, 3] where the performance was the most consistent. In fact, PSO presented a high sensitivity to w and k values. Higher or lower values for these parameters decreased the performance of the variants, which, at times, were unable to reach the feasible region of the search space in some problems, despite slightly improving the results in other test functions.

Lower L_i and upper U_i limits for each decision variable i are handled in the flight formula (Equation 8) as follows: After the flight, if the new value $\mathbf{x}_i(t+1)$ is outside the limits, the velocity value $\mathbf{v}_i(t+1)$ is halved until the new position is within the valid limits. In this way, the search direction is maintained.

The results will be discussed based on quality and consistency. Quality is measured by the best solution found from the set of 30 independent runs. Consistency is measured by the mean and standard deviation values, i.e. a mean value closer to the best known solution and a standard deviation value close to zero indicate a more consistent performance of the approach.

In order to have more statistical support, nonparametric statistical tests were applied to the samples presented in Tables 2 and 3. Kruskal-Wallis test was applied to pair of samples with the same size (30 runs) and Mann-Whitney test was applied to samples with different sizes ($<$30 runs) as to verify if the differences shown in the samples are indeed significant. Test problems where no feasible solutions were found for all the algorithms e.g. g20 and g22, or when just the one variant found feasible results e.g. g13 and g17 are not considered in these tests. The results obtained confirmed the differences shown in Tables 2 and 3, except in the following cases, where the performance of the compared approaches is considered similar in problems g03 and g11 for the global best PSO with inertia weight and the local best PSO with constriction factor, in problem g11 for the local best PSO with inertia weight and the local best PSO with constriction factor and in problems g02, g03, g08 and g24 for both (global and local) constriction factor variants.

The results in Tables 2 and 3 suggest that the local best PSO with constriction factor (last column in Tables 2 and 3) provides the best performance overall. With respect to the global best PSO with inertia weight, the local best PSO with constriction factor obtains results with better quality and consistency in twenty test problems (g01, g02, g04, g05, g06, g07, g08, g09, g10, g12, g13, g14, g15, g16, g17, g18, g19, g21, g23 and g24). With respect to the local best PSO with inertia weight, the local best PSO with constriction factor provides better quality and consistency results in sixteen problems (g01, g02, g05, g06, g07, g09, g10, g12, g13, g14, g15, g16, g17, g18, g19 and g21). Finally, with respect to the global best PSO with constriction factor, the local best PSO with constriction factor presents better quality and consistency results in seventeen problems (g01, g04, g05, g06, g07, g09, g10, g11, g12, g13, g14, g15, g16, g17, g18, g19 and g21).

The local best PSO with inertia weight provides the “best” quality result in one problem (g23).

The global best PSO with constriction factor obtains more consistent results in one problem (g23). Problems g20 and g22 could not be solved by any PSO variant; these problems have several equality constraints and are the most difficult to solve [23].

Comparing the global best variants (third and fourth columns in Tables 2 and 3) with respect to those local best PSOs (fifth and sixth columns in Tables 2 and 3) the results suggest that the last ones perform better in this sample of constrained search spaces i.e. the global best variants have problems finding the feasible region in some problems where the local best variants indeed find it (g05, g14, g15, g18 and g21). Finally, when comparing inertia weight variants (third and fifth columns in Tables 2 and 3) with respect to constriction factor variants (fourth and sixth columns in

Table 2 Statistical results of 30 independent runs on the first 12 test problems for the four PSO variants compared."(n)" means that in only "n" runs feasible solutions were found. Boldface remarks the best result per function. "-" means that no feasible solutions were found in any single run.

STATISTICS FROM 30 INDEPENDENT RUNS FOR THE PSO VARIANTS

Problem & best-known solution		global best (w=0.7)	global best (k=0.729)	local best (w=0.7)	local best (k=0.729)
g01	Best	-14.961	-14.951	-14.999	**-15.000**
-15.000	Mean	-11.217	-11.947	-12.100	**-13.363**
	St. Dev.	2.48E+00	1.81E+00	3.05E+00	**1.39E+00**
g02	Best	-0.655973	-0.634737	-0.614785	**-0.790982**
-0.803619	Mean	-0.606774	-0.559591	-0.543933	**-0.707470**
	St. Dev.	2.64E-02	3.03E-02	**2.00E-02**	5.92E-02
g03	Best	-0.080	-0.019	-0.045	**-0.126**
-1.000	Mean	-9.72E-03	-1.72E-03	-1.00E-02	**-1.70 E-02**
	St. Dev.	1.60E-02	**4.64E-03**	1.20E-02	2.70E-02
g04	Best	-30655.331	-30665.439	**-30665.539**	**-30665.539**
-30665.539	Mean	-30664.613	-30664.606	**-30665.539**	**-30665.539**
	St. Dev.	5.70E-01	5.40E-01	**7.40E-012**	**7.40E-012**
g05	Best	-	-	5126.646 (18)	**5126.496**
5126.498	Mean	-	-	6057.259	**5140.060**
	St. Dev.	-	-	232.25E+00	**15.52E+00**
g06	Best	-6959.517	-6959.926	-6958.704	**-6961.814**
-6961.814	Mean	-6948.937	-6948.121	-6941.207	**-6961.814**
	St. Dev.	6.31E+00	6.41E+00	9.05E+00	**2.67E-04**
g07	Best	43.731	38.916	41.747	**24.444**
24.306	Mean	68.394	64.186	59.077	**25.188**
	St. Dev.	40.69E+00	17.15E+00	7.65E+00	**5.9E-01**
g08	Best	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**
-0.095825	Mean	-0.095824	**-0.095825**	**-0.095825**	**-0.095825**
	St. Dev.	1.75E-07	7.25E-08	**4.23E-17**	**4.23E-17**
g09	Best	692.852	693.878	696.947	**680.637**
680.630	Mean	713.650	708.274	728.730	**680.671**
	St. Dev.	12.96E+00	10.15E+00	15.80E+00	**2.10E-02**
g10	Best	8024.273	8769.477	8947.646	**7097.001**
7049.248	Mean	8931.263	9243.752	9247.134	**7641.849**
	St. Dev.	39.0.6E+01	22.94E+01	**18.4.7E+01**	36.14E+01
g11	Best	**0.749**	**0.749**	0.750	**0.749**
0.749	Mean	0.752	0.755	0.799	**0.749**
	St. Dev.	9.27E-03	1.40E-02	5.70E-02	**1.99E-03**
g12	Best	-0.999	-0.999	-0.999	**-1.000**
-1.000	Mean	-0.999	-0.999	-0.999	**-1.000**
	St. Dev.	6.96E-07	5.13E-07	2.59E-05	**0.00E+00**

Table 3 Statistical results of 30 independent runs on the last 12 test problems for the four PSO variants compared."(n)" means that in only "n" runs feasible solutions were found. Boldface remarks the best result per function. "-" means that no feasible solutions were found in any single run.

STATISTICS FROM 30 INDEPENDENT RUNS FOR THE PSO VARIANTS					
Problem & best-known solution		global best (w=0.7)	global best (k=0.729)	local best (w=0.7)	local best (k=0.729)
g13	Best	-	-	-	**8.10E-02**
0.053949	Mean	-	-	-	**0.45**
	St. Dev.	-	-	-	**2.50E-01**
g14	Best	-	-	-41.400 (9)	**-41.496** (3)
-47.764	Mean	-	-	-38.181	**-40.074**
	St. Dev.	-	-	2.18E+00	**1.45E+00**
g15	Best	-	-	967.519 (5)	**961.715**
961.715	Mean	-	-	970.395	**961.989**
	St. Dev.	-	-	2.62E+00	**3.9E-01**
g16	Best	-1.904	-1.903	-1.904	**-1.905**
-1.905	Mean	-1.901	-1.901	-1.904	**-1.905**
	St. Dev.	1.46E-03	1.37E-03	1.51E-04	**5.28E-11**
g17	Best	-	-	-	**8877.634**
8876.981	Mean	-	-	-	**8932.536**
	St. Dev.	-	-	-	**29.28E+00**
g18	Best	-	-	-0.450967 (3)	**-0.866023**
-0.865735	Mean	-	-	-0.287266	**-0.865383**
	St. Dev.	-	-	1.40E-01	**8.65E-04**
g19	Best	36.610	36.631	36.158	**33.264**
32.656	Mean	42.583	43.033	39.725	**39.074**
	St. Dev.	7.05E+00	4.30E+00	**2.30E+00**	6.01E+00
g20	Best	-	-	-	-
0.188446	Mean	-	-	-	-
	St. Dev.	-	-	-	-
g21	Best	-	-	800.275 (3)	**193.778**
193.778	Mean	-	-	878.722	**237.353**
	St. Dev.	-	-	10.64E+01	**35.29E+00**
g22	Best	-	-	-	-
382.902	Mean	-	-	-	-
	St. Dev.	-	-	-	-
g23	Best	-3.00E-02 (5)	-228.338 (20)	**-335.387** (20)	-98.033 (16)
-400.003	Mean	107.882	**-20.159**	159.312	134.154
	St. Dev.	14.03E+01	**13.23E+01**	25.47E+01	17.99E+01
g24	Best	-5.507	-5.507	**-5.508**	**-5.508**
-5.508	Mean	-5.507	-5.507	**-5.508**	**-5.508**
	St. Dev.	2.87E-04	1.87E-04	**9.03E-16**	**9.03E-16**

Tables 2 and 3), there is no clear superiority. However, both variants (inertia weight and constriction factor) perform better coupled with local best PSO (ring social network).

The overall results from this first experiment suggest that the local best PSO with constriction factor is the most competitive approach (based on quality and consistency) in this set of test CNOPs. Besides, some important information regarding the behavior of PSO in constrained search spaces was obtained and discussed.

As an interesting comparison, in Table 4 the two most competitive PSO variants from this experiment (local best PSO with constriction factor and inertia weight) are compared with three state-of-the-art PSO-based approaches. The results show that these two variants are competitive in some test problems (g04, g08, g11 and g12). However, they are far from providing a performance like those presented by the state-of-the-art algorithms. Therefore, the most competitive PSO variant (local best PSO with constriction factor) will be improved in the next Section of this chapter.

7 Simple Modifications to the Original PSO

Besides the results presented, the experiment in the previous Section provided valuable information regarding two issues related to PSO for constrained search spaces. (1) The local best PSO with constriction factor presents a lower tendency to converge prematurely when solving CNOPs and (2) all PSO variants compared have problems dealing with test functions with equality constraints. Therefore, two simple modifications are proposed to this most competitive variant to improve its performance. This new version will be called Improved PSO (IPSO).

7.1 Dynamic Adaptation

Based on the velocity update formula (Eq. 10) two parameters were detected as the most influential in this calculation: (1) k, which affects the entire value of the velocity and (2) c_2, which has more influence in the calculation because, most of the time, the pbest value is the same as the current position of the particle i.e. this term in Eq. 10 may be eliminated, whereas the gbest value is different from the position of the particle in all the search process (except for the leader). Moreover, PSO has a tendency to prematurely converge [9]. Then, a dynamic (deterministic) adaptation mechanism [8] for these two parameters k and c_2 is proposed to start with low velocity values for some particles and to increase these values during the search process as follows: A dynamic value for k and c_2, based on the generation number will be used for a (also variable) percentage of particles in the swarm. The remaining particles in the swarm will use the fixed values for these two parameters. It is important to note that, at each generation, the particles which will use the dynamic values will be different e.g. a given particle may use the fixed values at generation "t" and the

Table 4 Comparison of results provided by two state-of-the-art PSO-based approaches and the two local best PSO variants. “(n)” means that in only “n” runs feasible solutions were found. Boldface remarks the best result per function. “-” means that no feasible solutions were found in any single run.

PSO VARIANTS AND STATE-OF-THE-ART ALGORITHMS						
Problem & best-known solution		local best ($w = 07$)	local best ($k = 0.729$)	Toscano & Coello [42]	Lu & Chen [25]	Cagnina et al. [2]
g01	Best	-14.999	**-15.000**	**-15.000**	**-15.000**	**-15.000**
-15.000	Mean	-12.100	-13.363	**-15.000**	-14.418	**-15.000**
g02	Best	-0.614785	-0.790982	**-0.803432**	-0.664	-0.801
-0.803619	Mean	-0.543933	-0.707470	**-0.790406**	-0.413	0.765
g03	Best	-0.045	-0.126	-1.004	**-1.005**	-1.000
-1.000	Mean	-1.00E-02	-1.70 E-02	**-1.003**	-1.002	-1.000
g04	Best	**-30665.539**	**-30665.539**	-30665.500	**-30665.659**	**-30665.659**
-30665.539	Mean	**-30665.539**	**-30665.539**	-30665.500	**-30665.539**	-30665.656
g05	Best	5126.646 (18)	5126.496	5126.640	**5126.484**	5126.497
5126.498	Mean	6057.259	**5140.060**	5461.081	5241.054	5327.956
g06	Best	-6958.704	**-6961.814**	-6961.810	-6961.813	-6961.825
-6961.814	Mean	-6941.207	**-6961.814**	-6961.810	-6961.813	-6859.075
g07	Best	41.747	24.444	24.351	**24.306**	24.400
24.306	Mean	59.077	25.188	25.355	**24.317**	31.485
g08	Best	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**
-0.095825	Mean	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**	-0.095800
g09	Best	696.947	680.637	680.638	**680.630**	680.636
680.630	Mean	728.730	680.671	680.852	**680.630**	682.397
g10	Best	8947.646	7097.001	7057.900	**7049.248**	7052.852
7049.248	Mean	9247.134	7641.849	7560.047	**7049.271**	8533.699
g11	Best	0.750	**0.749**	**0.749**	**0.749**	**0.749**
0.749	Mean	0.799	**0.749**	0.750	**0.749**	0.750
g12	Best	-0.999	**-1.000**	**-1.000**	**-1.000**	**-1.000**
-1.000	Mean	-0.999	**-1.000**	**-1.000**	**-1.000**	**-1.000**
g13	Best	-	8.10E-02	0.068	**0.053**	0.054
0.053949	Mean	-	**0.45**	1.716	0.681	0.967

dynamic values at generation “$t+1$”. The aim is to let, at each generation, some particles (those which use the dynamic values) to move at a slower velocity with respect to the remaining ones. The expected behavior is to slow down convergence and, as a result, better performance i.e. better quality and consistent results.

Based on the strong tendency of PSO to converge fast, a dynamic variation was chosen in such a way that in the first part of the process (half of total generations) k and c_2 values would remain low, and in the second half of the process they would increase faster. Then, the following function was chosen: $f(y) = y^4$, where $y = GEN/Gmax$. This function is presented in Figure 3, where it is noted that very low values are generated before 0.5 in the x-axis i.e. in the first half of the search

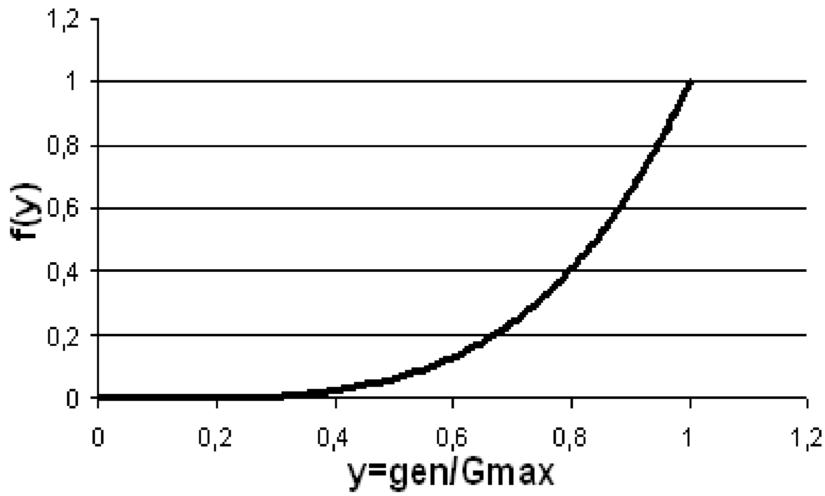

Fig. 3 Function used to dynamically adapt k and c_2 parameters. In the first half of the search low values are generated, while in the second half the values increase very fast.

process. This means that the values for the adapted parameters will be also low. However, in the second part of the search (0.5 to 1.0) the parameter values increase faster to reach their original values.

The expressions to update both parameter values at each generation "$t+1$" are defined as follows: $k^{t+1} = k \cdot f(y)$ and $c_2^{t+1} = c_2 \cdot f(y)$, where k and c_2 are the static values for these parameters. The initial values are small values close to zero e.g. 4E-13, and the values at the last generation will be exactly the fixed values ($k = 0.729$ and $c_2 = 2.5$).

As it was mentioned before, the number of particles which will use these dynamic values is also dynamic. In this case, based on observations considering the best performance, an oscillatory percentage of particles was the most suited. Therefore, a probability value is computed as to decide if a given particle will use either the static or the dynamic values: $p = k + \frac{(\sin(4\pi y))}{10.3}$, where k is the fixed value for this parameter ($k = 0.729$) and $y = GEN/Gmax$. The constant value 10.3 defines the maximum and minimum values p can take ($p \in [0.62, 0.82]$). A higher constant value decreases this range and a lower value increases it. The value suggested (10.3) worked well in all the experiments performed. The percentage of particles which will use the fixed parameters is modified as shown in Figure 4.

The main advantage of the dynamic mechanism proposed in this chapter over the addition of extra parameters (e.g. mutation operators), the combination of PSO variants or the modification of the original problem, all of them to keep PSO from converging prematurely (as shown on previous approaches in Section 5), is that the

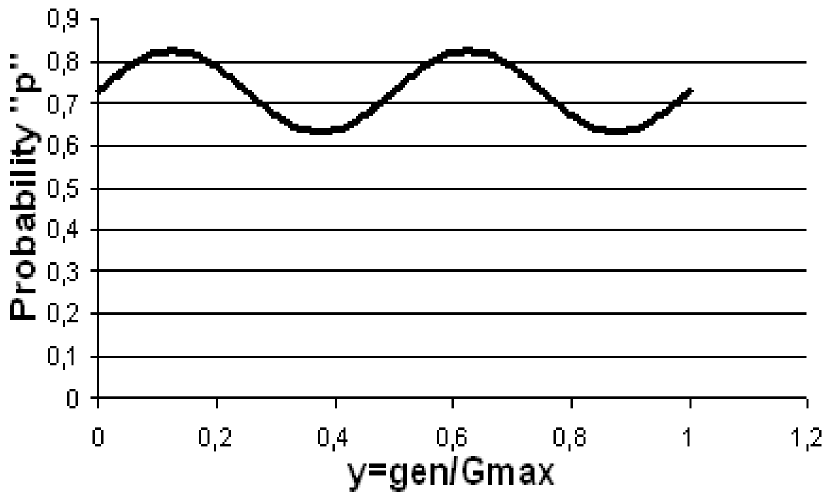

Fig. 4 Oscillatory percentage of particles that will use the fixed values for k and c_2. The remaining particles will use the dynamic values.

user does not need to fine-tune additional parameter values i.e. this work is done in IPSO by the own PSO. Even though the dynamic approach seems to be more complicated with respect to the addition of a parameter, this additional mechanism maintains the simplicity of PSO from the user's point of view.

7.2 Modified Constraint-Handling

The third feasibility rule proposed by Deb [5] selects, from two infeasible solutions, the one with the lowest sum of normalized constraint violation:

$$s = \sum_{i=1}^{m} \max\left(0, g\left(\mathbf{x}\right)\right) + \sum_{j=1}^{p} \max\left(0, \left(\left|h\left(\mathbf{x}\right)\right| - \varepsilon\right)\right) \tag{11}$$

As can be noted from Equation 11, the information of the violation of inequality and equality constraints are merged into one single value s. Besides, in the specialized literature there is empirical evidence that equality constraints are more difficult to satisfy than inequality constraints [27, 40, 31].

Based on the way s is computed, some undesired situations may occur when two infeasible solutions a and b are compared e.g. the s value from one of them (called s_a) can be lower than the the other one (s_b), but the violation sum for equality

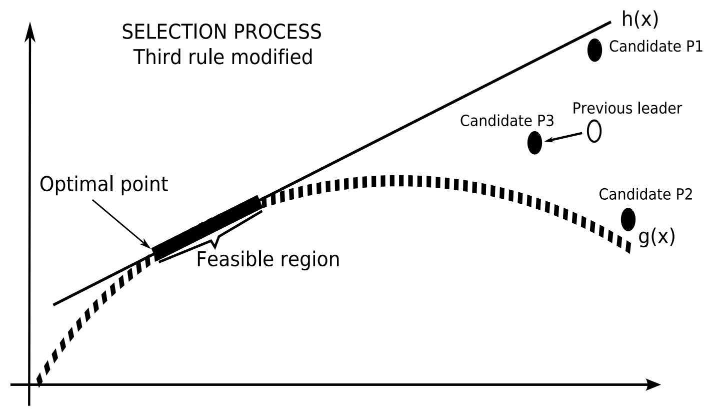

Fig. 5 Expected behavior on the modified constraint-handling mechanism.

constraints can be higher in s_a. Therefore, it may be more convenient to handle these sums separately as to provide the search with more detailed information in the selection process:

$$s_1 = \sum_{i=1}^{m} \max(0, g(\mathbf{x})) \tag{12}$$

$$s_2 = \sum_{j=1}^{p} \max(0, (|h(\mathbf{x})| - \varepsilon)) \tag{13}$$

After that, a dominance criterion (as defined in Section 2) is used to select the best solution by using the vector [*s1*,*s2*] for both solutions to be compared. The solution which dominates the other is chosen. If both solutions are nondominated between them, the age of the solution is considered i.e for the leader selection and for the pbest update there will be always a current solution and a new solution to be compared, if both solutions do not dominate each other, the older solution is kept. In this way, the solutions will be chosen/updated only if one amount of violation is decreased without increasing the other or if both amounts are decreased. The expected effect is detailed in Figure 5, where the current solution (white circle) must be replaced by a new one. Three candidate solutions are available: $P1$, $P2$ and $P3$ (all black circles). $P1$ is discarded because it decreases the violation of the equality constraint but also increases the violation of the inequality constraint. $P2$ is also discarded because it decreases the violation amount of the inequality constraint but also increases the violation of the equality constraint. $P3$ is chosen because both violation amounts are decreased i.e. $P3$ dominates $P1$ and $P2$.

IPSO is then based on the local best PSO with constriction factor as a search engine, coupled with the dynamic adaptation mechanism for k and c_2 parameters and the modification to the third rule of the constraint-handling technique. No additional operators, parameters, local search, problem re-definition, PSO variants mixtures nor original velocity or flight formulas modifications were considered on IPSO, whose details are found in Figure 6 (modifications are remarked) and its behavior and performance is analyzed in the next Section. The same mechanism explained in Section 6 to generate values within the allowed boundaries for the variables of the problem is also used in IPSO.

Begin
  GEN = 0
  Generate a swarm of random solutions ($\mathbf{x}_i$) $i = 1, 2, \ldots, SWARM\_SIZE$.
  Divide the swarm in n neighborhoods.
  Assign equal number of particles to each neighborhood.
  Initialize **for each particle**, $\mathbf{x}_{pbest_i} = \mathbf{x}_i$, and $\mathbf{v}_i(t) = 0$.
  Evaluate the fitness of the particle in each neighborhood.
  Do
    Select the leader ($\mathbf{x}_{lBest_i}$) of each neighborhood.
      <u>**by using the modified feasibility rules**</u>
    For each particle, update its velocity with (10).
      by using the corresponding leader of each neighborhood $\mathbf{x}_{lBest_i}$
      <u>**Depending of the p value use the fixed values for k and c_2**</u>
      <u>**Otherwise use the dynamic values for these parameters**</u>
    For each particle, update its position with (8).
    Evaluate the fitness of the new position **for each particle**.
    Update the $\mathbf{x}_{pbest_i}$ (memory) value **for each particle**.
      <u>**by using the modified feasibility rules**</u>
    GEN=GEN+1
  Until GEN = Gmax
End

Fig. 6 Improved PSO pseudocode. Modifications are underlined.

8 Experiments and Results

In this Section, four aspects of IPSO are analyzed: (1) The quality and consistency of its final results, (2) its online behavior by using two performance measures found in the specialized literature [27], (3) its convergence behavior by analyzing convergence graphs and (4) its performance compared to those provided by state-of-the-art PSO-based approaches to solve CNOPs. The same 24 test problems used in the preliminary experiment are considered in this Section.

8.1 Quality and Consistency Analysis

A similar experimental design to that used in the comparison of PSO variants is considered here. IPSO is compared against two PSO original variants: global best and local best PSO, both with constriction factor, as to analyze the convenience of the two modifications proposed. The parameter values are the same used in the previous experiment. 30 independent runs were performed and the statistical results are summarized in Tables 5 and 6 for the 24 test problems.

Like in the previous experiment, the nonparametric statistical tests were applied to the samples summarized in Tables 5 and 6. For the following problems, no significant differences were found among the results provided by the three algorithms: g04, g08 and g24. Besides, no significant difference in performance is found in problems g05, g13 and g17 when the local best PSO with constriction factor and IPSO are compared, and in problem g23 when the global and local best PSOs, both with constriction factor, are also compared. In all the remaining comparisons, the differences are significant. IPSO provides better quality and more consistent results in five problems (g03, g07, g10, g14 and g21), better quality results in two problems (g02 and g18) and it also obtains more consistent results in six problems (g01, g06, g09, g11, g19 and g23), all with respect to the local best PSO with constriction factor, which is the variant in which IPSO is based. The original PSO with constriction factor presents the best performance in problems g15. Also, it is more consistent in problems g02 and g18 and it finds the "best" quality result in problems g09 and g23. The global best PSO with constriction factor is not better in any single problem.

The overall analysis of this experiment indicates that the two simple modifications made to a competitive PSO variant lead to an improvement in the quality and mostly in the consistency of the final results e.g. in problems with a combination of equality and inequality constraints such as g21 and g23, IPSO provided a very consistent and good performance. The exception was g05, where, despite the better results in the samples for IPSO, the statistical test considered the differences as not significant.

8.2 On-Line Behavior Analysis

Two performance measures will be used to compare the two PSO variants and IPSO to know: (1) how fast the feasible region is reached and (2) the ability of each PSO to move inside the feasible region (difficult for most BIAs as analyzed in [27]).

1. Evals: Proposed by Lampinen [19]. It counts the number of evaluations (objective function and constraints) required to generate the first feasible solution. Then, a lower value is preferred because it indicates a faster approach to the feasible region of the search space.
2. Progress Ratio: Proposed by Mezura-Montes & Coello [27], it is a modification of Bäck's original proposal for unconstrained optimization [1]. It measures the

Table 5 Statistical results of 30 independent runs on the first 12 test problems for IPSO and the two PSO variants with constriction factor compared. “(n)” means that in only “n” runs feasible solutions were found. Boldface remarks the best result per function.

STATISTICS FROM 30 INDEPENDENT RUNS				
Problem & best-known solution		global best (k)	local best (k)	IPSO
g01	Best	**-15.000**	**-15.000**	**-15.000**
-15.000	Mean	-10.715	-13.815	**-15.000**
	St. Dev.	2.54E+00	1.58E+00	**0.00E+00**
g02	Best	-0.612932	-0.777758	**-0.802629**
-0.803619	Mean	-0.549707	**-0.717471**	-0.713879
	St. Dev.	**2.39E-02**	4.32 E-02	4.62 E-02
g03	Best	-0.157	-0.426	**-0.641**
-1.000	Mean	-0.020	-0.037	**-0.154**
	St. Dev.	**3.00E-02**	9.20E-02	1.70 E-01
g04	Best	**-30665.539**	**-30665.539**	**-30665.539**
-30665.539	Mean	**-30665.539**	**-30665.539**	**-30665.539**
	St. Dev.	**7.40E-12**	**7.40E-12**	**7.40E-12**
g05	Best	6083.449 (12)	5126.502	**5126.498**
5126.498	Mean	6108.013	5135.700	**5135.521**
	St. Dev.	9.78E+00	**9.63E+00**	1.23E+01
g06	Best	-6957.915	**-6961.814**	**-6961.814**
-6961.814	Mean	-6943.444	-6961.813	**-6961.814**
	St. Dev.	9.45E+00	4.66E-04	**2.81E-05**
g07	Best	45.633	24.463	**24.366**
24.306	Mean	60.682	25.045	**24.691**
	St. Dev.	7.50E+00	5.10E-01	**2.20E-01**
g08	Best	**-0.095825**	**-0.095825**	**-0.095825**
-0.095825	Mean	**-0.095825**	**-0.095825**	**-0.095825**
	St. Dev.	**4.23E-17**	**4.23E-17**	**4.23E-17**
g09	Best	705.362	**680.635**	680.638
680.630	Mean	736.532	680.675	**680.674**
	St. Dev.	1.58E+01	**2.90E-02**	3.00E-02
g10	Best	8673.098	7124.709	**7053.963**
7049.248	Mean	9140.877	7611.759	**7306.466**
	St. Dev.	2.36E+02	3.22E+02	**2.22E+02**
g11	Best	**0.749**	**0.749**	**0.749**
0.749	Mean	0.794	**0.753**	**0.753**
	St. Dev.	5.90E-02	1.00E-02	**6.53E-03**
g12	Best	-0.999	**-1.000**	**-1.000**
-1.000	Mean	-0.999	**-1.000**	**-1.000**
	St. Dev.	2.77E-05	**0.00E+00**	**0.00E+00**

Table 6 Statistical results of 30 independent runs on the last 12 test problems for IPSO and the two PSO variants with constriction factor compared. "(n)" means that in only "n" runs feasible solutions were found. Boldface remarks the best result per function. "-" means that no feasible solutions were found in any single run.

STATISTICS FROM 30 INDEPENDENT RUNS				
Problem & best-known solution		global best (k)	local best (k)	IPSO
g13 0.053949	Best	-	0.127872	**0.066845**
	Mean	-	0.520039	**0.430408**
	St. Dev.	-	**2.30E+00**	**2.30E+00**
g14 -47.764	Best	-47.394 (7)	-45.062 (4)	**-47.449**
	Mean	-38.619	-43.427	**-44.572**
	St. Dev.	4.95E+00	1.59E+00	**1.58E+00**
g15 961.715	Best	967.519 (5)	**961.715**	**961.715**
	Mean	969.437	**961.963**	962.242
	St. Dev.	2.62E+00	**3.20E-01**	6.20E-01
g16 -1.905	Best	-1.904	**-1.905**	**-1.905**
	Mean	-1.904	**-1.905**	**-1.905**
	St. Dev.	1.46E-04	5.28E-11	**2.42E-12**
g17 8876.981	Best	-	**8853.721**	8863.293
	Mean	-	8917.155	**8911.738**
	St. Dev.	-	3.17E+01	**2.73E+01**
g18 -0.865735	Best	-0.295425 (5)	-0.865989	**-0.865994**
	Mean	-0.191064	**-0.864966**	-0.862842
	St. Dev.	1.20E-01	**1.38E-03**	4.41E-03
g19 32.656	Best	37.568	**33.939**	33.967
	Mean	40.250	38.789	**37.927**
	St. Dev.	3.97E+00	3.97E+00	**3.20E+00**
g20 0.188446	Best	-	-	-
	Mean	-	-	-
	St. Dev.	-	-	-
g21 193.778	Best	666.081 (7)	193.768	**193.758**
	Mean	896.690	237.604	**217.356**
	St. Dev.	1.21E+02	3.60E+01	**2.65E+01**
g22 382.902	Best	-	-	-
	Mean	-	-	-
	St. Dev.	-	-	-
g23 -400.003	Best	-98.033 (16)	**-264.445**	-250.707
	Mean	134.154	70.930	**-99.598**
	St. Dev.	1.79E+02	2.58E+02	**1.20E+02**
g24 -5.508	Best	**-5.508**	**-5.508**	**-5.508**
	Mean	**-5.508**	**-5.508**	**-5.508**
	St. Dev.	**9.03E-16**	**9.03E-16**	**9.03E-16**

improvement inside the feasible region by using the objective function values of the first feasible solution and the best feasible solution reached at the end of the process. The formula is the following: $\mathrm{Pr} = \left| \ln \sqrt{\frac{f_{\min}(G_{ff})}{f_{\min}(T)}} \right|$, where $f_{\min}(G_{ff})$ is the objective function value of the first feasible solution found and $f_{\min}(T)$ is the objective function value of the best feasible solution found in all the search so far. A higher value means a better improvement inside the feasible region.

30 independent runs for each PSO variant, for each test problem, for each performance measure were computed. Statistical results are calculated and summarized in Tables 7 and 8 for the Evals performance measure and in Tables 9 and 10 for the Progress Ratio. The parameter values for the three PSOs are the same utilized in the previous experiment.

Regarding the Evals measure, some test problems are not considered in the discussion because feasible solutions were found in the initial swarm generated randomly. This was due to the size of the feasible region with respect to the whole search space (Table 1, fourth column). These problems are g02, g04, g08, g09, g12, g19 and g24. Problems g20 and g22 are also excluded because none of the algorithms could find a single feasible solution.The nonparametric tests applied to the samples of the remaining problems confirmed the significance of differences for all of them, with the exception of problem g11 for the three algorithms and in the comparison between the global best PSO and IPSO in problem g23.

The global best PSO with constriction factor is the fastest and also the most consistent variant to reach the feasible region in four problems (g01, g06, g07, g10). It is also the fastest (but not the most consistent) in problem g03 and it is the most consistent in problem g16. However, it failed to find a single feasible solution in problems g13 and g17 and it is able to find feasible solutions in just some runs (out of 30) in problems g05 (12/30), g14 (17/30), g15 (5/30), g18 (5/30), g21 (7/30) and g23 (16/30). The local best PSO with constriction factor provides the fastest and more consistent approach to the feasible region in problem g23 and it is the fastest (but not the most consistent) in problems g16, g18 and g21. Finally, IPSO presents the fastest and more consistent approach to the feasible region in four problems (g05, g13, g15 and g17) and it is the most consistent in four problems (g03, g14, g18 and g21).

The overall results for the Evals performance measure show that the global best PSO with constriction factor, based on its fast convergence, presents a very irregular approach to the feasible region, being the fastest in some problems, but failing to find feasible solutions in others. The local best PSO with constriction factor is not very competitive at all, whereas IPSO provides a very consistent performance, while not the fastest. However, IPSO has a good performance in problems g05 and g21, both with a combination of equality and inequality constraints. The exception in this regard is problem g23.

The behavior, regarding Evals, presented by IPSO is somehow expected, because the approach to the feasible region might be slower because some particles will use lower parameter values in the velocity update, mostly in the first half of the search.

Table 7 Statistical results for the EVALS performance measure based on 30 independent runs in the first 12 test problems for IPSO and the two PSO variants with constriction factor. "(n)" means that in only "n" runs feasible solutions were found. Boldface remarks the best result per function.

EVALS				
Problem		global best (k)	local best (k)	IPSO
g01	Best	**162**	246	252
	Mean	**306**	368	419
	St. Dev.	6.40E+01	**5.41E+01**	7.77E+01
g02	Best	**0**	**0**	**0**
	Mean	**0**	**0**	**0**
	St. Dev.	**0.00E+00**	**0.00E+00**	**0.00E+00**
g03	Best	**189**	366	457
	Mean	3568	2118	**1891**
	St. Dev.	3.50E+03	1.35E+03	**9.82E+02**
g04	Best	**0**	**0**	**0**
	Mean	4	**2**	**2**
	St. Dev.	5.14E+00	3.11E+00	**2.92E+00**
g05	Best	33459 (12)	16845	**13087**
	Mean	86809	23776	**17037**
	St. Dev.	4.53E+04	3.68E+03	**2.21E+03**
g06	Best	**180**	256	254
	Mean	**440**	513	562
	St. Dev.	2.73E+02	2.58E+02	**1.86E+02**
g07	Best	**178**	484	812
	Mean	**873**	1164	1316
	St. Dev.	7.11E+02	4.01E+02	**2.52E+02**
g08	Best	4	**0**	3
	Mean	**56**	78	76
	St. Dev.	**3.97E+01**	5.21E+01	5.79E+01
g09	Best	**5**	8	17
	Mean	**88**	94	107
	St. Dev.	5.01E+01	**4.81E+01**	6.91E+01
g10	Best	**242**	579	522
	Mean	**861**	972	1202
	St. Dev.	3.29E+02	**2.69E+02**	4.56E+02
g11	Best	**85**	249	364
	Mean	1662	1152	**1009**
	St. Dev.	2.10E+03	8.32E+02	**5.72E+02**
g12	Best	2	**0**	1
	Mean	19	**15**	25
	St. Dev.	**1.60E+01**	1.77E+01	2.17E+01

Table 8 Statistical results for the EVALS performance measure based on 30 independent runs in the last 12 test problems for IPSO and the two PSO variants with constriction factor. "(n)" means that in only "n" runs feasible solutions were found. Boldface remarks the best result per function. "-" means that no feasible solutions were found in any single run.

EVALS				
Problem		global best (k)	local best (k)	IPSO
g13	Best	-	11497	**8402**
	Mean	-	1.75E+04	**1.28E+04**
	St. Dev.	-	2.82E+03	**1.82E+03**
g14	Best	**7820** (17)	9481 (4)	8353
	Mean	33686	13485	**12564**
	St. Dev.	2.47E+03	3.13E+03	**3.07E+03**
g15	Best	24712 (5)	7299	**5228**
	Mean	68444	11805	**8911**
	St. Dev.	4.16E+04	2.43E+03	**1.72E+03**
g16	Best	164	**32**	106
	Mean	**309**	442	493
	St. Dev.	**1.28E+02**	2.26E+02	2.23E+02
g17	Best	-	21971	**16489**
	Mean	-	29458	**22166**
	St. Dev.	-	5.07E+03	**2.73E+03**
g18	Best	110395 (5)	**2593**	2614
	Mean	125303	5211	**4479**
	St. Dev.	2.31E+04	1.04E+03	**8.87E+02**
g19	Best	**0**	**0**	**0**
	Mean	**2**	**2**	**2**
	St. Dev.	2.15E+00	**2.13E+00**	2.90E+00
g20	Best	-	-	-
	Mean	-	-	-
	St. Dev.	-	-	-
g21	Best	43574 (7)	**11617**	13403
	Mean	82594	27978	**19652**
	St. Dev.	3.45E+04	7.11E+03	**3.51E+03**
g22	Best	-	-	-
	Mean	-	-	-
	St. Dev.	-	-	-
g23	Best	8499 (16)	**2081**	18304
	Mean	32661	**17797**	28764
	St. Dev.	2.58E+04	1.35E+04	**5.34E+03**
g24	Best	**0**	**0**	**0**
	Mean	**1**	**1**	2
	St. Dev.	1.95E+00	**1.88E+00**	2.36 E+00

Table 9 Statistical results for the PROGRESS RATIO performance measure based on 30 independent runs in the first 12 test problems for IPSO and the two PSO variants with constriction factor. "(n)" means that in only "n" runs feasible solutions were found. Boldface remarks the best result per function.

PROGRESS RATIO				
Problem		global best (k)	local best (k)	IPSO
g01	Best	0.302	0.346	**0.368**
	Mean	0.196	0.266	**0.295**
	St. Dev.	5.90E-02	5.00E-02	**3.80E-02**
g02	Best	**1.388**	1.373	1.218
	Mean	0.884	**1.015**	1.013
	St. Dev.	1.20E-01	1.00E-01	**9.00E-02**
g03	Best	**0.346**	**0.346**	0.334
	Mean	0.037	0.026	**0.067**
	St. Dev.	**6.50E-02**	7.25E-01	8.10E-02
g04	Best	0.110	0.120	**0.124**
	Mean	0.070	**0.080**	0.071
	St. Dev.	**2.30E-02**	**2.30E-02**	2.50-02
g05	Best	4.273E-07 (12)	**0.087**	**0.087**
	Mean	1.250E-07	**0.056**	0.036
	St. Dev.	**1.64E-07**	3.70E-02	3.20E-02
g06	Best	0.799	**0.807**	0.772
	Mean	0.306	**0.348**	0.296
	St. Dev.	2.00E-01	**1.80E-01**	1.90E-01
g07	Best	2.117	**2.504**	2.499
	Mean	1.656	1.919	**1.963**
	St. Dev.	3.60E-01	3.60E-01	**3.50E-01**
g08	Best	0.494	0.451	**0.556**
	Mean	0.317	0.304	**0.356**
	St. Dev.	9.10E-02	9.20E-02	**7.20E-02**
g09	Best	4.685	4.394	**4.768**
	Mean	**2.622**	2.209	2.510
	St. Dev.	1.29E+00	**1.12**E+00	1.24E+00
g10	Best	0.598	0.665	**0.678**
	Mean	0.360	**0.482**	0.468
	St. Dev.	1.30E-01	**1.00E-01**	1.10E-01
g11	Best	**0.143**	**0.143**	**0.143**
	Mean	0.088	**0.113**	0.101
	St. Dev.	**4.70E-02**	4.90E-02	5.20E-02
g12	Best	**0.342**	0.281	0.285
	Mean	**0.136**	0.119	0.096
	St. Dev.	7.20E-02	**6.70E-02**	7.40E-02

The nonparametric tests applied to the samples of results for the Progress Ratio showed no significant differences for the three algorithms in problems g04, g08, g12, g16 and g24 and in the comparison of the local best PSO with constriction factor and IPSO in problems g13 and g17. Problems g20 and g22 were discarded because no feasible solutions were found. In the remaining problems, the differences are significant.

Despite being very fast in reaching the feasible region, the global best PSO obtains the "best" improvement within the feasible region only in problem g02 and it is the most consistent in problem g09. The local best PSO obtains the "best" quality and most consistent results in problems g05, g06, g15, g17 and g18. Also, it presents the best result in problem g07 and the most consistent improvement in the feasible region in problems g10, and g19. IPSO is the best approach, based on quality and consistency in problems g01, g13, g14, g21 and g23. Besides, it presents the "best" quality results in problems g10 and g19 and it is the most consistent approach in problem g07.

As a conclusion for the Progress Ratio measure, IPSO does not significantly improve PSO's ability of moving inside the feasible region. However, IPSO is very competitive in problems with a combination of equality and inequality constraints (g21 and g23), but it is surpassed by the local best PSO with constriction factor in problem g05 as well as in other problems. A last finding to remark is the poor results obtained for the global best PSO with constriction factor. It seems that its fast approach to the feasible region leads to an inability to improve solutions inside it.

8.3 Convergence Behavior

The convergence behavior of the two original PSO variants and IPSO is graphically compared by plotting the run located in the mean value from a set of 30 independent runs. Problems where the behavior is very similar are omitted. The graphs are grouped in Figure 7. Based on the behavior found in those graphs, IPSO is able to converge faster than the two PSO variants in problems g01, g02 and g10, even local best PSO with constriction factor achieves similar results but in more generations. Global best PSO with constriction factor is trapped in a local optimum solution. In problem g03, the local best PSO provides the best convergence while IPSO and the global best PSO with constriction factor are trapped in local optima solutions. Finally, IPSO clearly shows a better convergence in problems g14, g17, g21 and g23. It is worth reminding that problems g21 and g23 have a combination of equality and inequality constraints. Therefore, the graphs suggest that the modified constraint-handling mechanism helps PSO in this kind of problems.

Table 10 Statistical results for the PROGRESS RATIO performance measure based on 30 independent runs in the last 12 test problems for IPSO and the two PSO variants with constriction factor. "(n)" means that in only "n" runs feasible solutions were found. Boldface remarks the best result per function. "-" means that no feasible solutions were found in any single run.

PROGRESS RATIO				
Problem		global best (k)	local best (k)	IPSO
g13	Best	-	1.165	**2.327**
	Mean	-	0.410	**0.549**
	St. Dev.	-	**3.40E-01**	5.70E-01
g14	Best	7.564E-04 (7)	0.077 (4)	**0.167**
	Mean	3.253E-04	0.028	**0.061**
	St. Dev.	**2.78E-04**	3.50-02	3.90E-02
g15	Best	1.520E-06 (5)	**5.460E-03**	5.419E-03
	Mean	6.866E-07	**2.805E-03**	2.571E-03
	St. Dev.	**5.79E-07**	1.82E-03	1.67E-03
g16	Best	0.379	**0.509**	0.412
	Mean	0.224	0.217	**0.246**
	St. Dev.	**7.20E-02**	9.20E-02	8.90E-02
g17	Best	-	**0.023**	0.022
	Mean	-	**8.342E-03**	6.015E-03
	St. Dev.	-	7.17E-03	**6.50E-03**
g18	Best	0.660 (5)	**1.540**	1.297
	Mean	0.348	**0.883**	0.690
	St. Dev.	**2.70E-01**	3.30E-01	2.90E+00
g19	Best	3.463	3.470	**3.580**
	Mean	2.975	**3.062**	3.050
	St. Dev.	2.70E-01	**1.80E-01**	3.2E-01
g20	Best	-	-	-
	Mean	-	-	-
	St. Dev.	-	-	-
g21	Best	7.252E-03 (7)	**0.819**	**0.819**
	Mean	1.036E-03	0.628	**0.646**
	St. Dev.	**2.74E-03**	1.60E-01	1.40E-01
g22	Best	-	-	-
	Mean	-	-	-
	St. Dev.	-	-	-
g23	Best	2.697E-03 (16)	0.691	**0.847**
	Mean	3.835E-04	0.139	**0.240**
	St. Dev.	**6.19E-04**	1.90E-01	2.10E-01
g24	Best	**1.498**	1.211	1.062
	Mean	**0.486**	0.443	0.481
	St. Dev.	3.50E-01	2.60E-01	**2.40E-01**

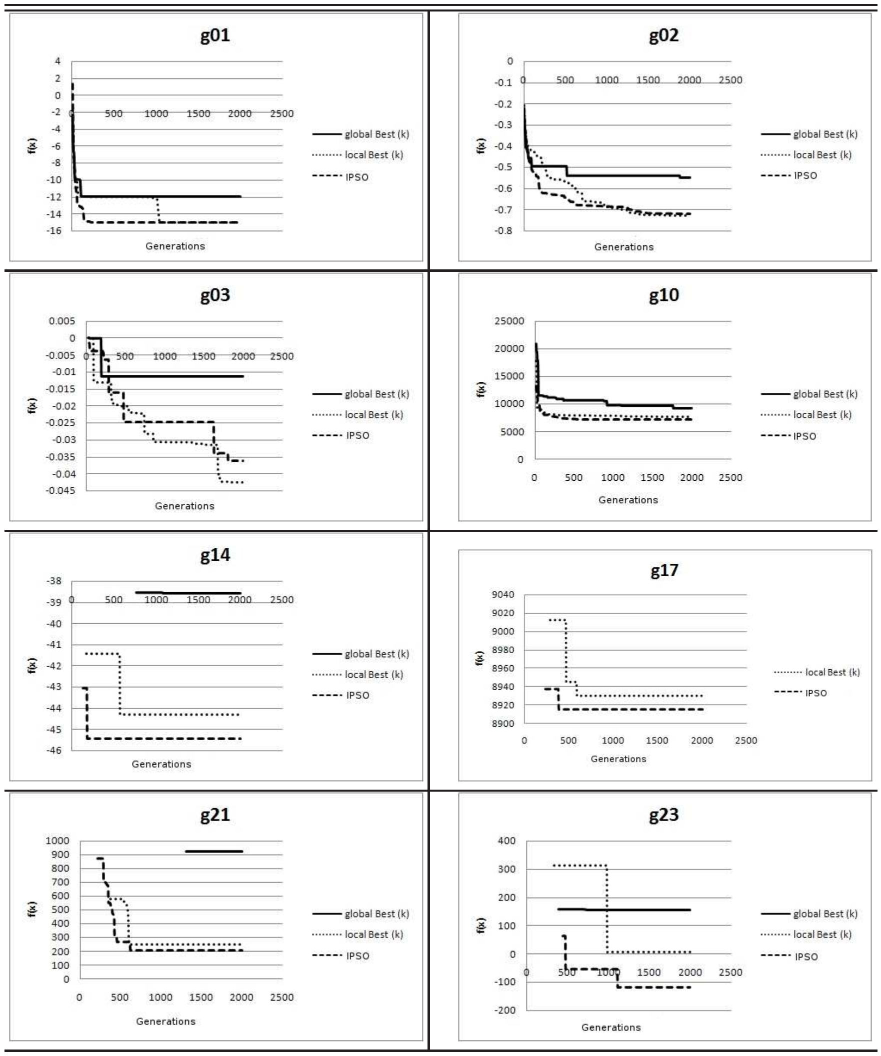

Fig. 7 Representative convergence graphs for the two compared PSO variants and IPSO.

8.4 Comparison with State-Of-The-Art PSO-Based Approaches

As a final comparison, IPSO's final results are compared with respect to those reported by four state-of-the-art PSO-based approaches. The approaches are the PSO algorithms proposed by Toscano and Coello [42], Li et al. [20], Lu and Chen [25] and Cagnina et al. [2]. These approaches are selected because they were tested against the same set of test problems. Statistical results (best, mean and worst values) are shown in Table 11. Test problems g14 to g24 are omitted because no results are reported by the compared approaches.

The statistical results show that IPSO provides the most consistent results in problems g05 (with a combination of equality and inequality constraints), g06 and g13. IPSO also has a similar performance with respect to the PSOs compared in problems g01, g04, g08, g11 and g12. Moreover, IPSO obtains the second best performance in problems g02, g07, g09 and g10. In problem g03 IPSO is not competitive at all. Regarding the computational cost of the compared approaches, Toscano and Coello [42] and Cagnina et al. [2] use 340,000 evaluations, Li et al. [20] do not report the number of evaluations required and Lu and Chen [25] report 50,000 evaluations. However, Lu and Chens' approach requires the definition of an extra parameter called ω and the original problem is also modified. IPSO requires 160,000 evaluations and does not add any extra operator or complex mechanism to the original PSO, keeping its simplicity.

9 Conclusions and Future Work

This chapter presented a novel PSO-based approach to solve CNOPs. Unlike traditional design steps to generate an algorithm to deal with constrained search spaces, which in fact may produce a more complex technique, in this work a preliminary analysis of the behavior of the most known PSO variants was performed as to get an adequate search engine. Furthermore, empirical evidence about the convenience of using PSO local best variants in constrained search spaces was found (constriction factor was better than inertia weight). From this first experiment, the local best PSO with constriction factor was the most competitive variant and two simple modifications were added to it: (1) a dynamic adaptation mechanism to control k and c_2 parameters, these to be used for a dynamically adapted percentage of particles in the swarm and (2) the use of a dominance criterion to compare infeasible solutions in such a way that new solutions are accepted only if both, the sums of inequality and equality constraint violations (handled separately) are decreased. This Improved PSO (IPSO) was compared against original PSO variants based on their final results and also based on their on-line behavior. IPSO's final results were significantly improved with respect to the original variants. On the other hand, IPSO was not the fastest to reach the feasible region and it did not improve considerably the ability to move inside the feasible region. In other words, the way the original PSO works in constrained search spaces was modified in such a way that a slower approach to the feasible region allowed IPSO to enter it from a more promising area. However, this issue requires a more in-depth analysis. The convergence behavior shown by IPSO suggest that their mechanisms promote a better exploration of the search space to avoid local optimum solutions in most of the test problems. Finally IPSO, which does not add further complexity to PSO, provided competitive and even better results, with a moderate computational cost, when compared with four state-of-the-art PSO-based approaches. A final conclusion of this work is that,

Table 11 Comparison of results with respect to state-of-the-art PSO-based approaches. (|) indicates that the results for this function were not available.

Comparison with state-of-the-art PSO-based approaches.						
Problem & best-known solution		Toscano & Coello [42]	Li, Tian & Kong [20]	Lu & Chen [25]	Cagnina et al. [2]	IPSO
g01	Best	**-15.000**	**-15.000**	**-15.000**	**-15.000**	**-15.000**
-15.000	Mean	**-15.000**	**-15.000**	-14.418	**-15.000**	**-15.000**
	Worst	**-15.000**	**-15.000**	-12.453	-134.219	**-15.000**
g02	Best	**-0.803432**	|	-0.664	-0.801	-0.802629
-0.803619	Mean	**-0.790406**	|	-0.413	0.765	-0.713879
	Worst	**-0.750393**	|	-0.259	0.091	-0.600415
g03	Best	-1.004	|	-1.005	**-1.000**	-0.641
-1.000	Mean	-1.003	|	-1.002	**-1.000**	-0.154
	Worst	-1.002	|	-0.934	**-1.000**	-3.747E-03
g04	Best	-30665.500	-30665.600	**-30665.539**	**-30665.659**	**-30665.539**
-30665.539	Mean	-30665.500	-30665.594	**-30665.539**	-30665.656	**-30665.539**
	Worst	-30665.500	-30665.500	**-30665.539**	-25555.626	**-30665.539**
g05	Best	5126.640	5126.495	**5126.484**	5126.497	5126.498
5126.498	Mean	5461.081	**5129.298**	5241.054	5327.956	5135.521
	Worst	6104.750	5178.696	5708.225	2300.5443	**5169.191**
g06	Best	-6961.810	**-6961.837**	-6961.813	-6961.825	-6961.814
-6961.814	Mean	-6961.810	**-6961.814**	-6961.813	-6859.075	**-6961.814**
	Worst	-6961.810	-6961.644	-6961.813	64827.544	**-6961.814**
g07	Best	24.351	|	**24.306**	24.400	24.366
24.306	Mean	25.355	|	**24.317**	31.485	24.691
	Worst	27.316	|	**24.385**	4063.525	25.15
g08	Best	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**
-0.095825	Mean	**-0.095825**	**-0.095825**	**-0.095825**	-0.095800	**-0.095825**
	Worst	**-0.095825**	**-0.095825**	**-0.095825**	-0.000600	**-0.095825**
g09	Best	680.638	**680.630**	**680.630**	680.636	680.638
680.630	Mean	680.852	680.654	**680.630**	682.397	680.674
	Worst	681.553	680.908	**680.630**	18484.759	680.782
g10	Best	7057.900	|	**7049.248**	7052.852	7053.963
7049.248	Mean	7560.047	|	**7049.271**	8533.699	7306.466
	Worst	8104.310	|	**7049.596**	13123.465	7825.478
g11	Best	**0.749**	**0.749**	**0.749**	**0.749**	**0.749**
0.749	Mean	0.750	**0.749**	**0.749**	0.750	0.753
	Worst	0.752	**0.749**	**0.749**	0.446	0.776
g12	Best	**-1.000**	|	**-1.000**	**-1.000**	**-1.000**
-1.000	Mean	**-1.000**	|	**-1.000**	**-1.000**	**-1.000**
	Worst	**-1.000**	|	**-1.000**	9386	**-1.000**
g13	Best	0.068	|	**0.053**	0.054	0.066
0.053949	Mean	1.716	|	0.681	0.967	**0.430**
	Worst	13.669	|	2.042	1.413	**0.948**

regarding PSO to solve CNOPs, the previous knowledge about the heuristic used as a search engine led to a less complex but competitive approach. Part of the future work is to improve the dynamic adaptation proposed in this chapter i.e. adaptive mechanism, testing more recent PSO variants such as the Fully Informed PSO [26], to test PSO variants with other constraint-handling mechanisms such as adaptive penalty functions [41] and to use IPSO in real-world problems.

Acknowledgements The first author acknowledges support from CONACyT through project number 79809-Y. The second author acknowledges support from CONACyT through a scholarship to pursue graduate studies at LANIA A.C.

References

1. Bäck T (1996) Evolutionary Algorithms in Theory and Practice. Oxford University Press, New York
2. Cagnina LC, Esquivel SC, Coello CAC (2006) A Particle Swarm Optimizer for Constrained Numerical Optimization. In: et al TR (ed) Parallel Problem Solving from Nature (PPSN IX). 9th International Conference, Reykjavik, Iceland, Springer, Reykjavik, Iceland, pp 910–919, Lecture Notes in Computer Science Vol. 4193
3. Clerc M, Kennedy J (2002) The Particle Swarm-Explosion, Stability, and Convergence in a Multidimensional Complex Space. IEEE Transactions on Evolutionary Computation 6(1):58–73
4. Coello CAC (2002) 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
5. Deb K (2000) An Efficient Constraint Handling Method for Genetic Algorithms. Computer Methods in Applied Mechanics and Engineering 186(2/4):311–338
6. Dorigo M, Maniezzo V, Colorni A (1996) The Ant System: Optimization by a Colony of Cooperating Agents. IEEE Transactions of Systems, Man and Cybernetics-Part B 26(1):29–41
7. Eiben A, Smith JE (2003) Introduction to Evolutionary Computing. Natural Computing Series, Springer Verlag
8. Eiben G, Schut MC (2008) New Ways to Calibrate Evolutionary Algorithms. In: Siarry P, Michalewicz Z (eds) Advances in Metaheuristics for Hard Optimization, Springer, Natural Computing Series, Heidelberg, Germany, pp 153–177
9. Engelbrecht A (2005) Fundamentals of Computational Swarm Intelligence. John Wiley & Sons
10. Fogel L (1962) Autonomous Automata. Industrial Research v.4 (12):14–19
11. Glover F, Laguna F (1997) Tabu Search. Kluwer Academic Publishers, Norwell, MA, USA
12. Glover F, Laguna M, Marti Ri (2003) Scatter Search. In: Advances in evolutionary computing: theory and applications, Springer-Verlag New York, Inc., New York, NY, USA, pp 519–537
13. He S, Prempain E, QHWu (2004) An Improved Particle Swarm Optimizer for Mechanical Design Optimization Problems. Engineering Optimization 36(5):585–605
14. Holland JH (1962) Concerning Efficient Adaptive Systems. in M C Yovits, G T Jacobi and G D Goldstein, eds, Self-Organizing Systems, Spartan Books, Washington DC pp 215–230
15. Kennedy J, Eberhart RC (2001) Swarm Intelligence. Morgan Kaufmann, UK
16. Kirkpatrick S, C D Gelatt J, Vecchi MP (1983) Optimization by Simulated Annealing. Science 220:671–680

17. Koza JR (1992) Genetic Programming. On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge, Massachusetts
18. Krohling RA, dos Santos Coelho L (2006) Coevolutionary Particle Swarm Optimization Using Gaussian Distribution for Solving Constrained Optimization Problems. IEEE Transactions on Systems, Man and Cybernetics Part B 36(6):1407–1416
19. Lampinen J (2002) A Constraint Handling Approach for the Diifferential Evolution Algorithm. In: Proceedings of the Congress on Evolutionary Computation 2002 (CEC'2002), IEEE Service Center, Piscataway, New Jersey, vol 2, pp 1468–1473
20. Li H, Jiao YC, Wang Y (2005) Integrating the Simplified Interpolation into the Genetic Algorithm for Constrained Optimization problems. In: et al YH (ed) Computational Intelligence and Security. International Conference, CIS 2005, Springer-Verlag, Xi'an, China, vol 3801, pp 247–254, Lecture Notes in Artificial Intelligence
21. Li X, Tian P, Kong M (2005) Novel Particle Swarm Optimization for Constrained Optimization Problems. In: ZZhang, Jarvis R (eds) Proceedings of the Australian Conference on Artificial Intelligence, Springer-Verlag, Heidelberg, Germany, pp 1305–1310, lNAI No. 3809
22. Li X, Tian P, Min X (2006) A Hierarchical Particle Swarm Optimization for Solving Bilevel Programming Problems. Artificial Intelligence and Soft Computing-ICAISC, Lecture Notes in Computer Science 4029 Springer pp 1169–1178
23. Liang JJ, Suganthan PN (2006) Dynamic Multi-Swarm Particle Swarm Optimizer with a Novel Constrain-Handling Mechanism. In: 2006 IEEE Congress on Evolutionary Computation (CEC'2006), IEEE, Vancouver, BC, Canada, pp 316–323
24. Liang JJ, Runarsson T, Mezura-Montes E, Clerc M, Suganthan PN, Coello CC, Deb K (2006) Problem Definitions and Evaluation Criteria for the CEC 2006 Special Session on Constrained Real-Parameter Optimization. Tech. rep., available at: http://www3.ntu.edu.sg/home/EPNSugan/
25. Lu H, Chen W (2006) Dynamic-Objective Particle Swarm Optimization for Constrained Optimization Problems. Journal of Combinatorial Optimization 12(4):409–419
26. Mendes R, Kennedy J, Neves J (2004) The Fully Informed Particle Swarm: Simpler, Maybe Better. IEEE Transactions on Evolutionary Computation 8(3):204–210
27. Mezura-Montes E, Coello CAC (2005) Identifying On-line Behavior and Some Sources of Difficulty in Two Competitive Approaches for Constrained Optimization. In: 2005 IEEE Congress on Evolutionary Computation (CEC'2005), IEEE Press, Edinburgh, Scotland, vol 2, pp 1477–1484
28. Mezura-Montes E, Coello-Coello CA (2008) Multiobjective Problems Solving from Nature: From Concepts to Applications, Springer-Verlag, Natural Computing Series, 2008, ISBN: 978-3-540-72963-1., chap Constrained Optimization via Multiobjective Evolutionary Algorithms, p pp:5376
29. Mezura-Montes E, López-Ramírez BC (2007) Comparing Bio-Inspired Algorithms in Constrained Optimization problems. In: 2007 IEEE Congress on Evolutionary Computation (CEC'2007), IEEE Press, Singapore, pp 662–669
30. Michalewicz Z, Schoenauer M (1996) Evolutionary Algorithms for Constrained Parameter Optimization Problems. Evolutionary Computation 4(1):1–32
31. Paquet U, Engelbrecht AP (2003) A New Particle Swarm Optimiser for Linearly Constrained Optimization. In: Proceedings of the Congress on Evolutionary Computation 2003 (CEC'2003), Canberra, Australia, IEEE Service Center, Piscataway, New Jersey, vol 1, pp 227–233
32. Parsopoulos K, Vrahatis M (2005) Unified Particle Swarm Optimization for Solving Constrained Engineering Optimization Problems. Advances in Natural Computation, Pt 3 pp 582–591, Lecture Notes in Computer Science Vol. 3612
33. Powell D, Skolnick MM (1993) Using Genetic Algorithms in Engineering Design Optimization with Non-Linear Constraints. In: Forrest S (ed) Proceedings of the Fifth International Conference on Genetic Algorithms (ICGA-93), University of Illinois at Urbana-Champaign, Morgan Kaufmann Publishers, San Mateo, California, pp 424–431

34. Price KV, Storn RM, Lampinen JA (2005) Differential Evolution. A practical Approach to Global Optimization. Springer
35. Runarsson TP, Yao X (2000) Stochastic Ranking for Constrained Evolutionary Optimization. IEEE Transactions on Evolutionary Computation 4(3):284–294
36. Schoenauer M, Xanthakis S (1993) Constrained GA Optimization. In: Forrest S (ed) Proceedings of the Fifth International Conference on Genetic Algorithms (ICGA-93), University of Illinois at Urbana-Champaign, Morgan Kauffman Publishers, San Mateo, California, pp 573–580
37. Schwefel HP (ed) (1995) Evolution and Optimization Seeking. Wiley, New York
38. Shi Y, Eberhart R (1998) A Modified Particle Swarm Optimizer. Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence v. 4-9:69–73
39. Shi Y, Eberhart RC (1998) Parameter Selection in Particle Swarm Optimization. http://wwwengriupuiedu/ shi/PSO/Paper/EP98/psof6/ep98_psohtml
40. Takahama T, Sakai S, Iwane N (2006) Solving Nonlinear Constrained Optimization Problems by the ε Constrained Differential Evolution. In: Proceedings of the 2006 IEEE International Conference on Systems, Man, and Cybernetics, IEEE Press, Taipei, Taiwan, pp 2322–2327
41. Tessema B, Yen GG (2006) A Self Adaptative Penalty Function Based Algorithm for Constrained Optimization. In: 2006 IEEE Congress on Evolutionary Computation (CEC'2006), IEEE, Vancouver, BC, Canada, pp 950–957
42. Toscano-Pulido G, Coello Coello CA (2004) A Constraint-Handling Mechanism for Particle Swarm Optimization. In: Proceedings of the Congress on Evolutionary Computation 2004 (CEC'2004), Portland, Oregon, USA, IEEE Service Center, Piscataway, New Jersey, vol 2, pp 1396–1403
43. Wei J, Wang Y (2006) A Novel Multi-objective PSO Algorithm for Constrained Optimization Problems. In: et al TDW (ed) Simulated Evolution and Learning (SEAL 2006), Springer. Lecture Notes in Computer Science Vol. 4247, pp 174–180