

An efficient constraint handling methodology for multi-objective evolutionary algorithms

Una metodología eficiente para manejo de restricciones en algoritmos evolutivos multi-objetivo

*Mauricio Granada Echeverri¹, *Jesús María López Lezama^{2*}, Ruben Romero³*

¹Departamento de Ingeniería Eléctrica, Universidad Tecnológica de Pereira, Vereda la Julita, Pereira, Risaralda, Colombia

²Grupo Gimel. Facultad de Ingeniería, Universidad de Antioquia, Calle 67 N.º 53-108, Medellín, Colombia

³Departamento de Ingeniería Eléctrica, Feis-Unesp-Ilha Solteira-Brasil. Avenida Brasil, 56 – Centro, 15385-000, Ilha Solteira – SP, Brasil

(Recibido el 29 de octubre de 2008. Aceptado el 26 de mayo de 2009)

Abstract

This paper presents a new approach for solving constraint optimization problems (COP) based on the philosophy of lexicographical goal programming. A two-phase methodology for solving COP using a multi-objective strategy is used. In the first phase, the objective function is completely disregarded and the entire search effort is directed towards finding a single feasible solution. In the second phase, the problem is treated as a bi-objective optimization problem, turning the constraint optimization into a two-objective optimization. The two resulting objectives are the original objective function and the constraint violation degree. In the first phase a methodology based on progressive hardening of soft constraints is proposed in order to find feasible solutions. The performance of the proposed methodology was tested on 11 well-known benchmark functions.

----- *Keywords:* Evolutionary algorithms, multi-objective algorithms, constraint optimization.

Resumen

Este artículo presenta un nuevo enfoque para resolver problemas de optimización restringidos (POR) basado en la filosofía de programación lexicográfica de objetivos. En este caso se utiliza una metodología de dos

* Autor de correspondencia: teléfono: + 57 + 4 + 219 55 55, fax: + 57 + 4 + 219 05 07, correo electrónico: lezama@udea.edu.co (J. López).

fases usando una estrategia multi-objetivo. En la primera fase se concentra el esfuerzo en encontrar por lo menos una solución factible, descartando completamente la función objetivo. En la segunda fase se aborda el problema como bi-objetivo, convirtiendo el problema de optimización restringida a un problema de optimización irrestricto de dos objetivos. Los dos objetivos resultantes son la función objetivo original y el grado de violación de las restricciones. En la primera fase se propone una metodología basada en el endurecimiento progresivo de restricciones blandas para encontrar soluciones factibles. El desempeño de la metodología propuesta es validado a través de 11 casos de prueba bastante conocidos en la literatura especializada.

----- *Palabras clave:* Algoritmos evolutivos, algoritmos multi-objetivo, optimización restringida.

Introduction

Evolutionary algorithms (EA) have been widely used in the solution of optimization problems. These techniques, compared with the traditional nonlinear programming methods, handle a smaller amount of information (gradients, and Hessians, among others), are of easy implementation, and constitute useful tools for global search. Additionally, they have a smaller probability of converging to a local optimal solution, and are able to obtain good quality results in problems of great size [1]. Many researchers have developed a great amount of EA to solve constraint optimization problems (COP). The different methodologies found in the literature to handle with COP can be classified in four main groups: 1) methods based on penalty functions, 2) methods based on the preference of feasible solutions instead of the non feasible ones, 3) hybrid methods, and 4) methods based on multi-objective optimization. This last group is currently of great scientific interest and becomes the state-of-the-art of the constraint optimization algorithms. A detail description of these methodologies is out of the scope of this paper. For a more in dept reading, the interested reader is referred to [2, 3] and [4].

Most of the real world problems involve equality and inequality constraints. The general problem formulation with continuous parameters and constraints is defined in [5] as shown in (1):

$$\text{Min } f(X), \quad X = (x_1, x_2, \dots, x_n) \in \mathfrak{R}^n$$

Subject to :

$$g_j(X) \leq 0, \quad (j = 1, \dots, q) \quad (1)$$

$$h_j(X) = 0, \quad (j = q + 1, \dots, m)$$

$$X \in F \subseteq S$$

The objective function f is defined on the search space $S \subseteq \mathfrak{R}^n$, and the set $F \subseteq S$ defines the feasible region. The feasible region F is restricted by a set of m constraints ($m \geq 0$) with q inequality constraints $g_j(X)$, and $m - q$ equality constraints $h_j(X)$.

The application of multi-objective evolutionary algorithms (MOEA) has additional advantages compared to other optimization methods, especially when solving COP. Some of these advantages are [6]:

- Constraint problems can be handled in a natural fashion. That is, it is not necessary to formulate artificially penalized objective functions, and additionally penalty parameters are not needed. These parameters introduce a subjective component to the problem solution.
- In real world problems, it is unusual to find rigid or hard constraints. Therefore, a constraint violation margin (soft constraints) is permitted as long as an important improvement in the objective function is obtained.

- MOEA allow obtaining a set of solutions, denominated Pareto-Optimal-Front (POF), with the best commitments between the objective functions involved in the problem. Thus, it is possible to find solutions that violate constraints marginally. Figure 1 shows a non-dominated set of solutions in the $f-v$ space, where f is the original objective function value, and v is the constraint violation index. In this scheme, one objective is the violation degree of constraints and the other is the original objective function value. The minimum feasible solution (point A), the minimum solution considering soft constraints within a violation margin ϵ (point B), and the original feasible solutions of the single-objective problem are also shown in Figure 1. All the solutions of the POF that are between the points A and B are of great interest.

The philosophy of the proposed methodology is inspired by the goal programming methods [7], where the main idea is to find solutions that reach a reference (predefined objective) for one or more objective functions. If these solutions do not exist, the task will be to find solutions where the difference with the reference is minimum. On the other hand, if there is a solution with the same value as the reference objective function, the task will be to identify this solution. The lexicographical method is among the goal programming methods. In this case the different goals are categorized within many levels of priority. In This way, the problem is first solved considering only one goal with the corresponding constraints of the first priority level.

If there are multiple solutions in the previous step, another goal programming problem is formulated considering the second level of priority. The goals of the first level of priority are used as equality constraints to assure that the second problem solution does not violate the first level constraints. The procedure is repeated sequentially for other priority levels.

Figure 2 illustrates the operation principle of the lexicographical goal programming method

for a minimization problem with two objective functions f_1 and f_2 . If it is considered that f_1 is more important than f_2 , the procedure consists of first minimizing the problem only considering f_1 and ignoring f_2 . In this way, the set of solutions is represented by the segments AB and CD for the first priority level. The solution of the second priority level will be the one that minimizes f_2 along the segments AB and CD . In this case, the solution of the second priority level will be D which is the global solution of the problem. If f_2 is more important than f_1 the problem solution changes to point E .

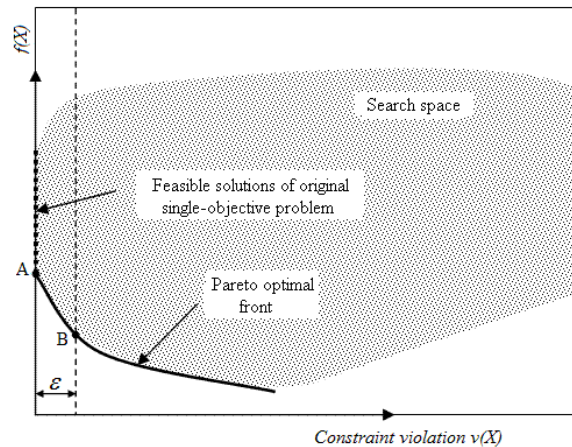


Figure 1 Search space of a two-objective problem with hard and soft constraints

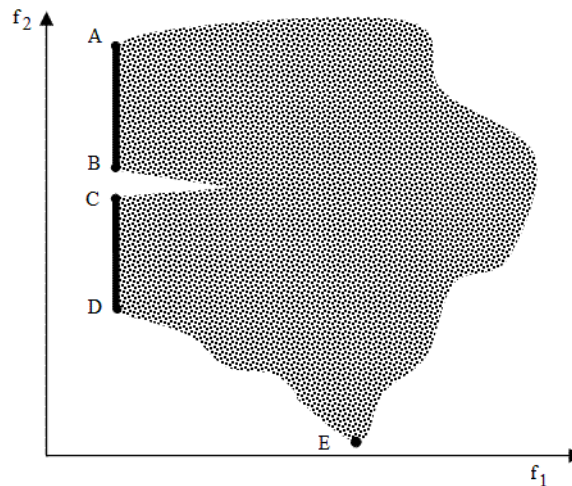


Figure 2 Lexicographical goal programming method

Proposed methodology for COP

The proposed methodology for COP is based on the philosophy of lexicographical goal programming. It consists of turning a COP into a bi-objective problem, where one objective function is the original one $f(X)$ and the other, is the constrained violation degree $v(X)$. In other words, one objective function considers optimality and the other one considers feasibility. The algorithm is composed by two phases. In the first stage, the original objective function is completely discarded and the optimization problem is concentrated on minimizing the constrained violation degree of the solutions. Thus, the algorithm might find a feasible solution because the search is concentrated only on the minimization of the constrained violation degree. The second phase consists of optimizing simultaneously the original objective function and the constrained violation degree using a multi-objective strategy.

Phase I: Constraint enforcement algorithm

In this phase, the objective function is completely discarded and all the algorithm effort is directed towards finding at least one feasible solution. For each alternative i of the population, a fitness function, according to $v_i(X)$, is assigned. Then, an elitist strategy is used to assure that the solution with smaller $v(X)$ is included in the following generation. This phase allows obtaining a solution that satisfies all the constraints (usable solution in the real world).

This technique is appropriate to solve highly constrained problems, where finding a feasible solution can be difficult. In order to find the constrained violation degree of an alternative X in the constraint j , the first step of the proposed strategy consists of turning the equality constraints into soft constraints using a tolerance δ . Thus, the constraint violation degree of the alternative will be given by (2)

$$c_j(X) = \begin{cases} \max \{0, g_j(X)\} & j = 1, \dots, q \\ \max \{0, |h_j(X)| - \delta\} & j = q + 1, \dots, m \end{cases} \quad (2)$$

Where $| \cdot |$ denotes the absolute value.

In order to give the same importance degree to all constraints, each alternative violated must be normalized dividing it by the greatest violation value of the population. In this case the greatest violation value for each constraint j is calculated using (3).

$$c_{\max}(j) = \max_X \{c_j(X)\} \quad (3)$$

The maximum violation value for each constraint in the whole population is used to normalize each violated constraint calculated in (2). Finally, to produce a scalar number that represents the constraint violation degree for each alternative of the population (in a range between 0 and 1), the normalized values are added and then divided by the total number of m constraints as shown in (4).

$$v(X) = \frac{\sum_{j=1}^m \frac{c_j(X)}{c_{\max}(j)}}{m} \quad (4)$$

Obtaining the fitness function for phase one

In order to illustrate the calculus of the fitness function, the constraint optimization problem presented in [8] and defined by the set of equations (5) is considered.

$$\begin{aligned} \min \quad & G5(X) \equiv 3x_1 + 0.000001x_1^3 + 2x_2 + \frac{0.000002}{3}x_2^3 \\ \text{st} \quad & r_1 \equiv x_3 - x_4 - 0.55 \leq 0 \\ & r_2 \equiv x_4 - x_3 - 0.55 \leq 0 \\ & r_3 \equiv 1000\sin(-x_3 - 0.25) + 1000\sin(-x_4 - 0.25) + 894.8 - x_1 = 0 \\ & r_4 \equiv 1000\sin(x_3 - 0.25) + 1000\sin(x_4 - 0.25) + 894.8 - x_2 = 0 \\ & r_5 \equiv 1000\sin(x_4 - 0.25) + 1000\sin(x_3 - 0.25) + 1294.8 = 0 \\ & r_6 \equiv 0 \leq x_i \leq 1200, \quad i = 1, 2 \\ & r_7 \equiv -0.55 \leq x_i \leq 0.55, \quad i = 3, 4 \end{aligned} \quad (5)$$

Where r_1, r_2, \dots, r_7 are the problem constraints. Table 1 shows a population of 5 alternatives randomly generated. The small letters (x_1, x_2, \dots, x_5) stand for the variables and the capital

letters (X_1, X_2, \dots, X_5) stand for the solution alternatives. During the generation of the population it is guaranteed that constraints r_6 and r_7 are satisfied (limits of the decision variables). Thus, the problem is only limited by the first 5 constraints (r_1, r_2, \dots, r_5).

Evaluating each alternative of the population for each of the constraints in problem (5) and discarding the objective function completely, the data registered in table 2 are obtained. Applying (2) and assuming a tolerance $\delta = 0.0001$ the data presented in table 3 are obtained, and the term $c_{\max}(j)$ is calculated using (3).

Table 1 Randomly generated population

	x_1	x_2	x_3	x_4
X_1	1156.4	133.69	0.14667	0.3424
X_2	867.67	791.77	-0.3331	0.08332
X_3	955.76	982.42	0.36467	0.01741
X_4	722.69	153.13	0.37178	0.055625
X_5	747.24	966.49	0.12927	-0.4855

Table 2 Violation values for each alternative and each constraint

	r_1	r_2	r_3	r_4	r_5
X_1	-0.74573	-0.35427	-1206.3	226.85	1332.8
X_2	-0.9665	-0.1335	-216.97	-1065.9	1294.6
X_3	-0.20274	-0.89726	-901.88	123.9	501.93
X_4	-0.23384	-0.86616	-711.27	929.26	565.26
X_5	0.064772	-11.648	10.647	164.62	-137.11

Then, when a new alternative is generated, a comparison between the constraint violations for each alternative, and the maximum violations calculated in (3) allows keeping the values of the vector $c_{\max}(j)$ updated. It is advisable to generate an additional vector $i_{\max}(j)$ containing the index of the alternatives that produce each $c_{\max}(j)$. Thus, for example, the maximum violation of

constraint 1 is caused by individual 5 (X_5) as shown table 3.

Table 3 Constraint violations considering $\delta = 0.0001$

	C_1	C_2	C_3	C_4	C_5
X_1	0	0	1206.2	226.84	1332.8
X_2	0	0	216.96	1065.9	1294.6
X_3	0	0	901.87	123.89	501.92
X_4	0	0	711.26	929.25	565.25
X_5	0.064772	0	10.637	164.61	137.1
$C_{\max}(j)$	0.064772	-	1206.2	1065.9	1332.8
$I_{\max}(j)$	5	-	1	2	1

Finally, applying expression (4) to the data shown in table 3, a scalar vector $v(X)$ that quantifies the infeasible degree of each individual of the population is obtained as shown in (6).

$$v(X) = [0.44256, 0.43024, 0.2481, 0.37711, 0.25322]^T \quad (6)$$

The vector $v(X)$ corresponds to the fitness function of phase one, which will be used in the selection process. For the feasible solution search, a traditional genetic algorithm (GA) with real codification is used incorporating progressive hardening of soft constraints.

Progressive hardening of soft constraints (PHSC) – Phase one

Figure 1 shows the soft constraints handled through a violation margin \mathcal{E} . The technique used to find the feasible solutions consists of considering an interval of the violation margin ($\mathcal{E}_{\min} \leq \mathcal{E} \leq \mathcal{E}_{\max}$). In this way, the initial objective of the GA is to minimize the parameter $v(X)$ of each alternative, calculated with expression (4) considering \mathcal{E}_{\max} . The algorithm is initially run with a high violation margin, and therefore, the GA reaches its objective with a low computational effort. Next, the violation margin is reduced every time the GA reaches a partial objective, until a

constraint violation margin smaller or equal to ϵ_{\min} is finally reached. In this point, the GA has found a feasible solution.

The expression $\epsilon = (1 - \tau)\epsilon$ is used as a reduction strategy of the violation margin. ϵ_{\min} corresponds to the tolerance used to evaluate the equality constraints fulfillment (a typical value is $\epsilon_{\min} = \delta = 0.0001$). ϵ_{\max} is a “bait” value that allows the GA to easily find a population with a reasonable infeasibility margin. From this population, the optimization process guides the search towards feasible regions of better quality until finding a feasible solution with the desired precision degree. Figure 3 shows the search process of feasible solutions for problem (5), starting with a random population with $\epsilon_{\max} = 0.4$ and $\tau = 0.5$. The white circles correspond to the initial population, the asterisks indicate the evolution of the population after several generations, and the vertical dashed lines indicate the current violation margin.

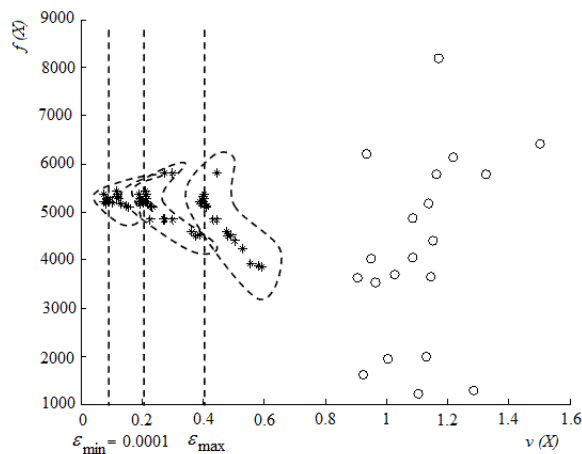


Figure 3 Search process of feasible solutions. Evolution of the population alternatives for different ϵ

Phase II: Optimization algorithm for constraint problems

Phase II is activated when at least one feasible solution has been found by phase one. In phase one the fitness function corresponds to the constraint violation degree, and the evolution of

the alternatives considers the quality of each non-dominated set to which each alternative belongs. In phase II the constraint violation and the original objective function must be minimized simultaneously within a modified objective space as shown in figure 1 (space f - v). The feasible alternative with the best objective function will be the current incumbent of the space search.

A GA and an elitist operator based on a non-dominated sorting (NSGA-II [9]) are used in this paper for solving the bi-objective problem. In addition, to preserve diversity in the alternatives belonging to the non-dominated solutions set, a niches scheme is used, taking into consideration the normalized Euclidean distance between two objective vectors. This distance is known as *crowding distance metric*. A detailed description of the calculus of this distance is presented in [7], and [10]. The multi-objective theory introduces the concept of *dominance*, which defines that a solution X_1 dominates another solution X_2 if both conditions 1) and 2) are true:

- 1) The solution X_1 is not worse than X_2 in all objectives.
- 2) The solution X_1 is strictly better than X_2 in at least one objective.

If any of the above conditions is violated, the solution X_1 does not dominate the solution X_2 . It is possible to apply this definition in an iterative way to any set of solutions belonging to a multi-objective optimization problem to establish the dominant and non-dominant sets of alternatives.

The set of dominant solutions through all the objective space is called the Pareto-optimal front. Therefore, the GA (or any other evolutionary approach) aims to move the current front in each iteration towards regions of better quality. Figure 4 shows the optimal front evolution for problem (5) using NSGA-II approach.

There are a bunch of different strategies that can be used to intensify the exploration in the target search region. Some of these strategies as reported in the specialized literature can be: the guided domination approach, dominance principle

by weights, and in general, modifications on the crowding distance metric. In this paper the guided domination approach described in [7] was implemented. Thus, a different dominance concept for minimization problems is formulated. A weighted function of the objectives is defined as shown in (7)

$$\Omega_i(f(x)) = f_i(x) + \sum_{j=1, j \neq i}^M a_{ij} f_j(x), \quad i = 1, 2, \dots, M$$

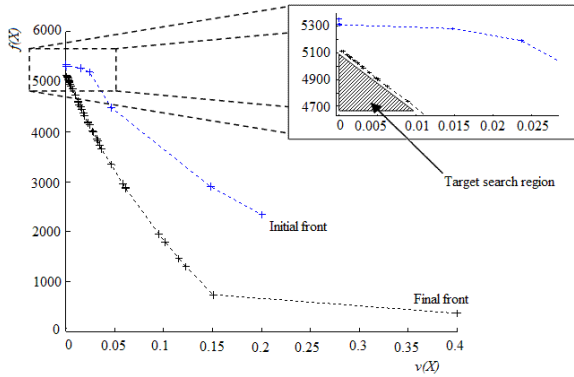


Figure 4 Pareto-Optimal-Front and target search region

Where a_{ij} represents the improvement in the j -th objective function for a one-unit loss in the i -th objective function. Then, the new dominance concept is:

A solution X_1 dominates another solution X_2 if $\Omega_i(f(x_1)) \leq \Omega_i(f(x_2))$ for all $i=1, 2, \dots, M$ and the strict inequality is satisfied at least for one objective.

In this problem, there are two objective functions ($M=2$). The two weighted functions are shown in (8) and (9).

$$\Omega_1(f_1, f_2) = f_1 + a_{12}f_2 \quad (8)$$

$$\Omega_2(f_1, f_2) = a_{21}f_1 + f_2 \quad (9)$$

Thus, the modified definition of dominance allows a larger region to become dominated

by any solution than the one allowed by the traditional definition.

Besides, by choosing appropriate values of the coefficients a_{12} and a_{21} , a section of the Pareto-optimal region can be emphasized (see Figure 5). In this paper, in order to intensify the exploration of an interesting region of the Pareto-optimal front (as shown in Figure 4) the following coefficients were used: $a_{12} = 0$ and $a_{21} = 1.33$.

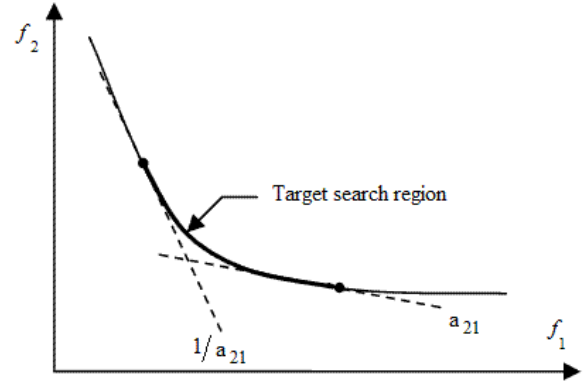


Figure 5 The non-dominated portion of the Pareto-optimal region

Genetic Algorithm

The multi-objective technique (NSGA-II) requires the incorporation of a GA that improves the Pareto-optimal front quality during the iterative process. A GA with the following characteristics is used:

Real codification: binary chain codification is not used, which implies a modification in the recombination and mutation operators.

Linear crossover: the implemented crossover operator creates three solutions (offspring) in each generation t from two parent solutions $X_i^{1,t}$ and $X_i^{2,t}$, as shown in expressions (10), (11) and (12).

$$1/2 X_i^{1,t} + 1/2 X_i^{2,t} \quad (10)$$

$$3/2 X_i^{1,t} - 1/2 X_i^{2,t} \quad (11)$$

$$-1/2 X_i^{1,t} + 3/2 X_i^{2,t} \quad (12)$$

Out of these three solutions, one is eliminated by tournament.

Random mutation: the mutation scheme consists of creating a random alternative $Y_i^{(1,t+1)}$ considering all search space $Y_i^{(1,t+1)} = r_i (X_i^{(U)} - X_i^{(L)})$. Where r_i is a random number between $[0,1]$ and superscripts U and L indicate the superior and inferior search space limit, respectively.

Test cases and results

The multi-objective NSGA-II method and the PHSC approach proposed in this paper were applied to 11 test cases reported in [8] and [11]. In Table 4 a summary of the 11 test cases is presented. LI, NE, and NI represent the number of linear inequalities, nonlinear equations and nonlinear inequalities, respectively, n is the number of decision variables involved and a is the active constraint. The mutation rate, for all cases, is 1% and the recombination rate is 90%. The population size for all cases is 15 individuals. The maximum generation number is 5000 and 30 runs for each case were executed. For cases G2 and G3, $k=30$ was used. For all cases $\epsilon_{\max} = 0.4$, and $\epsilon_{\min} = 0.0001$. The obtained results are shown

in table 5. It can be observed that for most of the cases, the proposed methodology reaches values of the objective function equal to those reported in the specialized literature. Particularly, for case G5 (represented in this paper by the set of equations (5)) there were found three alternatives, all of them with objective values better than the best value reported in the literature. Table 6 presents detail information of the variables and constraint values for the best alternatives found by the proposed approach for problem G5.

In general terms, the use of PHSC showed a better performance of the algorithm in phase I. Particularly, the G5 case is highly constrained and to find a feasible solution is a difficult task. Venkatraman reports, for the G5 case, an average number of generations of 1807.82 to find the first feasible solution on 50 runs with $\epsilon = 0.001$. Using PHSC an average of 405.3 generations is obtained on 30 runs with $\epsilon = 0.0001$. The use of a smaller tolerance allows obtaining a greater number of non-dominated solutions in the target search region. Another strongly constraint case is G10, for this case the average number of generations reported by Venkatraman is 99.86. However, the average number obtained applying PHSC was 38.7.

Table 4 Summary of eleven test cases. (for G2 and G3 it is assumed k=30)

Test case	n	Type of function	LI	NE	NI	a
G1	13	quadratic	9	0	0	6
G2	k	nonlinear	0	0	2	1
G3	k	polynomial	0	1	0	1
G4	5	quadratic	0	0	6	2
G5	4	cubic	2	3	0	3
G6	2	cubic	0	0	2	2
G7	10	quadratic	3	0	5	6
G8	2	nonlinear	0	0	2	0
G9	7	polynomial	0	0	4	2
G10	8	linear	3	0	3	6
G11	2	quadratic	0	1	0	1

The algorithm performance in phase II is similar to the one reported by Venkatraman in [11]. Nevertheless, for the G5 case the proposed method was able to find, in 30 runs, 3 solutions with a better objective function than the one reported in [11]. These solutions belong to the dominated front and

have an acceptable constraint violation degree (see table 6). Comparing the best alternative reported with the 3 alternatives found, it can be noticed that the alternatives 1 and 3 exactly satisfy constraints $R3=0$ and $R4=0$. All alternatives satisfy the inequality constraints $R1$ and $R2$.

Table 5 Comparison of best results. ANG= average number of generations when the first feasible solution is found.

<i>Function</i>	<i>Best known solution</i>	<i>Venkatraman and Yen [1]</i>	<i>Koziel and Michalewicz [2]</i>	<i>Proposed methodology</i>	<i>ANG Venkatraman and Yen</i>	<i>ANG Proposed methodology</i>
Min. G1	-15	-14.9999	-14.7864	-14.9999	11.24	0
Max. G2	0.803553	0.803190	0.799530	0.803190	0	0
Max. G3	1.0	1.0	0.9997	1.0	31.68	5.5
Min. G4	-30665	-30665.5312	-30664.9	-30665.5312	0	0
Min. G5	5126.4981	5126.63049	NR	5126.331	1807.82	405.3
Min. G6	-6961.8	-6961.17856	-6952.1	-6961.17856	289.52	45.6
Min. G7	24.306	24.410977	24.620	24.410977	53.22	9.9
Max. G8	0.095582	0.095825	0.095825	0.095825	9.28	1.5
Min. G9	680.63	680.7622	680.91	680.7622	5.84	0
Min. G10	7049.33	7060.5528	7147	7060.5528	99.86	38.7
Min. G11	0.750	0.7490	0.9	0.7490	13.32	0

Table 6 Variables and constraint values for the best alternatives of problem G

	<i>Alternative 1</i>	<i>Alternative 2</i>	<i>Alternative 3</i>	<i>Best alternative reported</i>	
Objective function value	5126.4945	5126.4691	5126.331	5126.4975	
Variables	x_1	680.3447227503	678.0676632572	681.5644559478	679.9453
	x_2	1025.639306343	1028.06265646	1024.293474739	1026.067
	x_3	0.000118591184	0.12021181155	0.11770661330	0.1188764
	x_4	-0.000396368384	-0.39559444472	-0.39676709250	-0.3962336
Constraints	R_1	-0.035040431265	-0.03419374372	-0.03552629420	-0.03489
	R_2	-1.064959568734	-1.06580625627	-1.06447370579	-1.06511
	R_3	0	0.000045898864	0	0.000033030076
	R_4	0	0.000522908572	0	0.000247240858
	R_5	0.000817247899	0.008254636764	0.03317435154	-0.00009672673

Conclusions

In this paper a new methodology to deal with constraint optimization problems was presented. The main contribution of the proposed methodology consists of an efficient constraint handling approach using progressive hardening of soft constraints along with an intensive exploration of a target search region of the Pareto-optimal front.

The multi-objective NSGA-II method along with the proposed methodology was implemented on 11 test cases widely studied in the specialized literature. Results showed that the proposed methodology is competitive with the state-of-the-art constraint optimization algorithms. In particular, for test case G5, three different alternatives better than the one reported in the literature were found. For the other test cases the algorithm found the best solution already reported. However, in some cases, a considerable reduction of the number of generations was achieved.

Future work will consider other recombination and mutation strategies using real codification, such as blend crossover, simulated binary crossover, simplex crossover, non-uniform mutation and polynomial mutation, among others. The use of algorithms that incorporate the lateral diversity concept, such as NSGAI-controlled, allows a search of greater quality in the target search region and can be implemented with the purpose of improving some results. This philosophy can be applied to highly constrained problems.

References

1. D. Powell, M. Skolnick. "Using genetic algorithms in engineering design optimization with nonlinear constraints". *Proceedings of the 5th International conference on Genetic Algorithms*. Urbana-Champaign. 1993. pp. 424-431.
2. J. Kim, H. Myung. "Evolutionary programming techniques for constraint optimization problems". *IEEE Transactions on Evolutionary Computation*. Vol 1. 1997. pp. 129-140.
3. K. Deb. "An efficient constraint handling method for genetic algorithms". *Computational Methods Applied on Mechanical Engineering*. Vol. 186. 2000. pp. 311-338.
4. A. Kurri, J. Gutierrez. "Penalty function methods for constrained optimization with genetic algorithms: A statistical analysis". *Proceedings of the 2nd Mexican international conference on artificial intelligence*. Mérida Mx. 2002. pp. 108-117.
5. Z. Michalewicz, M. Schoemauer. "Evolutionary algorithms for constrained parameter optimization problems". *Evolutionary Computation*. Vol. 4. 1996. pp. 1-32.
6. C. Zixing, W. Yong. "A multi-objective optimization-based evolutionary algorithm for constrained optimization". *IEEE Trans. Evolutionary Computation*. Vol. 10. 2006. pp. 659-675.
7. K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. Department of mechanical Engineering. Indian Institute of technology, Kanpur, India. 2^a ed. John Wiley and Sons. New York. 2004. pp. 408-424.
8. S. Koziel, Z. Michalewicz. "Evolutionary algorithms, homomorphous mappings and constrained parameter optimization". *Evolutionary Computation*. Vol. 7. 1999. pp. 19-44.
9. K. Deb, A. Pratap, S. Agarwal, T. Meyarivan. "A fast and elitist multi-objective genetic algorithm: NSGA-II". *IEEE Transactions on Evolutionary Computation*. Vol. 6. 2002. pp. 182-197.
10. C. A. Peñuela, M. Granada. "Optimización multiobjetivo usando un algoritmo genético y un operador elitista basado en un ordenamiento no-dominado (NSGA-II)". *Revista Scientia et Technica*. Vol 35. 2007. pp. 175-170.
11. S. Venkatraman, G. Yen. "A generic framework for constrained optimization using genetic algorithms". *IEEE Transactions on Evolutionary Computation*. Vol. 9. 2005. pp. 424-432.