Multiobjective-Based Concepts to Handle Constraints in Evolutionary Algorithms

Efrén Mezura-Montes and Carlos A. Coello Coello

CINVESTAV-IPN
Evolutionary Computation Group (EVOCINV)
Departamento de Ingeniería Eléctrica
Sección de Computación
Av. Instituto Politécnico Nacional No. 2508
Col. San Pedro Zacatenco
México D.F. 07300, MÉXICO
emezura@computacion.cs.cinvestav.mx
ccoello@cs.cinvestav.mx

Abstract

This paper presents the main multiobjective optimization concepts that have been used in evolutionary algorithms to handle constraints in global optimization problems. A review of some approaches developed under these concepts are discussed. Additionally, A comparison of four representative of these techniques using well-known benchmark test functions is shown. Finally, the analysis of the results obtained, based on three main points (quality, consistency and diversity) and also some conclusions and future trends are provided.

1 Introduction

Evolutionary Algorithms (EAs) are heuristics that have been successfully applied in a wide set of areas [6, 18, 10], both in single- and in multiobjective optimization. However, EAs lack a mechanism able to bias efficiently the search towards the feasible region in constrained search spaces. This has triggered a considerable amount of research and a wide variety of approaches have been suggested in the last few years to incorporate constraints into the fitness function of an evolutionary algorithm [2, 19].

The most common approach adopted to deal with constrained search spaces is the use of penalty functions. When using a penalty function, the amount of constraint violation is used to punish or "penalize" an infeasible solution so that feasible solutions are favored by the selection process. Despite the popularity of penalty functions, they have several drawbacks from which the main one is that they require a careful fine tuning of the penalty factors that accurately estimates the degree of penalization to be applied as to approach efficiently the feasible region [27, 2].

Among the several approaches that have been proposed as an alternative to the use of penalty functions, there is a group of techniques in which the constraints of a problem are handled as objective functions (i.e., a single-objective constrained problem is restated as an unconstrained multiobjective problem). This paper precisely focuses on these techniques.

With this study, we want to know how the different mechanisms taken from multiobjective optimization perform solving different types of problems. The analysis of results will be based on three aspects: quality, consistency and diversity in the population.

This paper is organized as follows. Section 2 presents the basic concepts both from global optimization and from multiobjective optimization that are going to be used in the remainder of this paper. In Section 3, the most popular multiobjective-based constraint-handling techniques are discussed. Section 4 presents a comparative study in which four of the techniques discussed in the previous section are tested on several benchmark problems taken from the standard constraint-handling literature [19, 4]. Section 5 discusses the results obtained, and Section 6 provides some conclusions and possible paths for future research.

2 Basic Concepts

We are interested in the general non linear programming problem in which we want to:Find $\vec{x}$ which optimizes $f(\vec{x})$ subject to: $g_i(\vec{x}) \leq 0, \quad i = 1, \ldots, n$ $h_j(\vec{x}) = 0, \quad j = 1, \ldots, p$ where $\vec{x}$ is the vector of solutions $\vec{x} = [x_1, x_2, \ldots, x_r]^T$, n is the number of inequality constraints and p is the number of equality constraints (in both cases, constraints could be linear or nonlinear). If we denote with $\mathcal{F}$ to the feasible region and with $\mathcal{S}$ to the whole search space, then it should be clear that $\mathcal{F} \subseteq \mathcal{S}$. For an inequality constraint that satisfies $g_i(\vec{x}) = 0$, we will say that is active at $\vec{x}$. All equality constraints h_j (regardless of the value of $\vec{x}$ used) are considered active at all points of $\mathcal{F}$. Now, we will define some basic concepts from multiobjective optimization.

In a general multiobjective optimization problem we want to find the vector $\vec{x}^* = [x_1^*, x_2^*, \ldots, x_r^*]^T$ which will satisfy the n inequality constraints: $g_i(\vec{x}) \geq 0 \quad i = 1, 2, \ldots, n$ the p equality constraints $h_i(\vec{x}) = 0 \quad i = 1, 2, \ldots, p$ and will optimize the vector function $\vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \ldots, f_k(\vec{x})]^T$ where $\vec{x} = [x_1, x_2, \ldots, x_r]^T$ is the vector of decision variables.

Having several objective functions, the notion of "optimum" changes, because in multiobjective optimization problems, the aim is to find good compromises (or "trade-offs") rather than a single solution as in global optimization. The notion of "optimum" most commonly adopted is that originally proposed by Francis Ysidro Edgeworth in 1881 [8] and later generalized by Vilfredo Pareto (in 1896) [20]. This notion is normally referred to as "Pareto optimality" and is defined as: A point $\vec{x}^* \in \mathcal{F}$ is Pareto optimal if for every $\vec{x} \in \mathcal{F}$ and $I = \{1, 2, \ldots, k\}$ either,$\forall_{i \in I}(f_i(\vec{x}) = f_i(\vec{x}^*))$ or, there is at least one $i \in I$ such that $f_i(\vec{x}) > f_i(\vec{x}^*)$. In words, this definition says that $\vec{x}^*$ is Pareto optimal if there exists no feasible vector $\vec{x}$ which would decrease some criterion without causing a simultaneous increase in at least one other criterion. The phrase "Pareto optimal" is considered to mean with respect to the entire decision variable space unless otherwise specified.

Other important definitions associated with Pareto optimality are: Pareto Dominance that is defined follows: A vector $\vec{u} = (u_1, \ldots, u_k)$ is said to dominate $\vec{v} = (v_1, \ldots, v_k)$ (denoted by $\vec{u} \preceq \vec{v}$) if and only if u is partially less than v, i.e., $\forall i \in \{1, \ldots, k\}$, $u_i \leq v_i \wedge \exists i \in \{1, \ldots, k\} : u_i < v_i$, and Pareto Optimal Set: For a given multiobjective optimization problem, $\vec{f}(x)$, the Pareto optimal set ($\mathcal{P}^*$) is defined as: $\mathcal{P}^* := \{x \in \mathcal{F} \mid \neg\exists\ x' \in \mathcal{F} \ \ \vec{f}(x') \preceq \vec{f}(x)\}$.

3 Multiobjective-based Constraint Handling techniques

The main idea of adopting multiobjective concepts to handle constraints is to redefine the single-objective optimization of $f(\vec{x})$ as a multiobjective optimization problem in which we will have $m + 1$ objectives, where m is the total number of constraints (the additional objective is obviously the original objective function of the problem). Then, we can apply any multiobjective optimization technique [6] to the new vector $\bar{v} = (f(\vec{x}), f_1(\vec{x}), \ldots, f_m(\vec{x}))$, where $f_1(\vec{x}), \ldots, f_m(\vec{x})$ are the original constraints of the problem. An ideal solution $\vec{x}$ would thus have $f_i(\vec{x})$=0 for $1 \leq i \leq m$ and $f(\vec{x}) \leq f(\vec{y})$ for all feasible $\vec{y}$ (assuming minimization).

Three are the mechanisms taken from evolutionary multiobjective optimization that are the most frequently incorporated into constraint-handling techniques:

1. Use of Pareto dominance as a selection criterion.
2. Use of Pareto ranking [10] to assign fitness in such a way that nondominated individuals (i.e., feasible individuals in this case) are assigned a higher fitness value.
3. Split the population in subpopulations that are evaluated either with respect to the objective function or with respect to a single constraint of the problem. This is the selection mechanism adopted in the Vector Evaluated Genetic Algorithm (VEGA) [26]. In the remaining of the paper We will refer to this mechanism as a "population-based"

To solve this type of problems it is necessary to maintain a balance between feasible and infeasible solutions in order to sample the feasible region of the search space widely enough to reach the global optimum solution.

In multiobjective optimization the goal is to find a set of trade-off solutions which are considered good in all the objectives to be optimized. In global optimization we want to reach only the global optimum. Therefore, some changes must be done to those approaches to adapt them to reach only this global optimum. These new criteria are the following: Feasibility of solutions must be considered better than infeasible solutions and the number of violated constraints and the amount of constraint violation will emerge as selection criteria. Furthermore, a mechanism to maintain diversity should be considered.

We will now introduce a description of some approaches that have been designed to apply the concepts discussed before. Camponogara & Talukdar [1] proposed an approach in which a global optimization problem was transformed into a bi-objective problem where the first objective is to optimize the original objective function and the second is to minimize:

$$\Phi(\mathbf{x}) = \sum_{i=1}^{n} \max(0, g_i(\mathbf{x})) \quad (1)$$

Equation (1) tries to minimize the total amount of constraint violation of a solution. At each generation, based on the generated Pareto sets, a search direction is defined and a linear search is performed. At pre-defined intervals, the worst half of the population is replaced with new random solutions to avoid premature convergence. This indicates some of the problems of the approach to maintain diversity. Additionally, the use of line search within a GA adds some extra computational cost. The authors of this approach validated it using a benchmark consisting of five test functions. The results obtained were either optimal or very close to it. The main drawback of this approach is its additional computational cost.

An approach similar to a min-max formulation used in multiobjective optimization combined with tournament selection was proposed by Jiménez and Verdegay [14]. The selection criteria is based on the following rules:

- Between two feasible individuals, the one with a higher fitness wins.
- A feasible individual wins over an infeasible individual.
- Between two infeasible individuals, the one with the lowest amount of constraint violation wins.

This approach was validated using four test functions, and the results obtained in most cases were very close to the optima. A subtle problem with this approach is that the evolutionary process first concentrates only on the constraint satisfaction problem and therefore it samples points in the feasible region essentially at random [28]. This means that in some cases (e.g., when the feasible region is disjoint) we might land in an inappropriate part of the feasible region from which we will not be able to escape. However, this approach may be a good alternative to find a feasible point in a heavily constrained search space. The relative simplicity of this approach is another advantage of this technique.

Ray et al. [23] proposed the use of a Pareto ranking approach that operates on three spaces: objective space, constraint space and the combination of the two previous spaces. This approach also uses mating restrictions based on the feasibility of each individual in order to ensure better constraint satisfaction in the offspring generated and a selection process that eliminates weaknesses in any of these spaces. Also, a niche mechanism based on Euclidian distances is used. This approach only requires between 2% and 10% of the number of fitness function evaluations required by the homomorphous maps of Koziel and Michalewicz [15] which is one of the best techniques to handle constraints known to date. The main drawback of Ray et al.'s approach is that its implementation is considerably more complex than any of the other techniques previously discussed.

Jiménez et al. [13] proposed an algorithm that uses Pareto dominance inside a pre-selection scheme to solve several types of optimization problems (multiobjective, constraint satisfaction, global optimization, and goal programming problems). The approach redefines the problem as an unconstrained multiobjective optimization problem in which objectives are given priorities. Feasible solutions with a good objective function value are given the highest priority. The authors use a real-coded nongenerational GA with two types of crossover operators (uniform and arithmetic) and two mutation operators (uniform and non-uniform). The authors argue that this pre-selection mechanism is an implicit niche formation technique because individuals are replaced only by similar ones (i.e., their offspring). As only the best individuals are inserted into the new population, this scheme is also an elitist strategy. This approach was validated with eleven test functions, producing very good results. Note however, that the authors do not specify the computational cost of the approach and it is not clear if the approach is competitive with other techniques in that regard.

Ray [22] extended his previous work on constraint-handling [23] in which the emphasis was to find a robust optimized solution which is not sensitive to parametric variations due to incomplete information of the problem or to changes on it. This approach is capable of handling constraints and finds feasible solutions that are robust to parametric variations produced over time. This is achieved using the individual's self-feasibility and its neighborhood feasibility. A mechanism based on raking values in both spaces (objective space and constraint space) is used to select the best individuals and copy them into the next population. The remaining portion of the new population is filled by mating two parents and using different criteria based on feasibility of the solutions. Ray used a real-coded GA with Simulated Binary Crossover. The results reported in two well-known design problems [22] showed that the proposed approach did reach less sensitive, but not close to the optimum solutions. In contrast, the other techniques analyzed showed significant changes when the parameters were perturbed. The main drawback of this approach is its difficulty to implement it.

Surry & Radcliffe [28] combined the Vector Evaluated Genetic Algorithm (VEGA) [26] and Pareto Ranking to handle constraints in an approach called COMOGA (Constrained Optimization by Multi-Objective Genetic Algorithms). In this technique, individuals are ranked depending of their sum of constraint violation (number of individuals dominated by a solution). However, the selection process is based not only on ranks, but also on the fitness of each solution. COMOGA uses a non-generational GA and extra parameters defined by the user (e.g., a parameter called ϵ is used to define the change rate of P_{cost}). One of these parameters is P_{cost}, that sets the rate of selection based on fitness. The remaining $1 - P_{cost}$ individuals are selected based on ranking values. P_{cost} is defined by the user at the beginning of the process and it is adapted during evolution using as a basis the percentage of feasible individuals that one wishes to have in the population. COMOGA was applied on a gas network design problem and it was compared against a penalty function approach. Although COMOGA showed a slight improvement in the results with respect to a penalty function, its main advantage is that it does not require a fine tuning of penalty factors or any other additional parameter. The main drawback of COMOGA is that it requires several extra parameters, although its authors argue that the technique is not particularly sensitive to their values [28].

Parmee & Purchase [21] proposed to use VEGA [26] to guide the search of an evolutionary algorithm to the feasible region of an optimal gas turbine design problem with a heavily constrained search space. Nevertheless, they did not use it to reach the global optimum solution. After having a feasible point, they generated an optimal hypercube around it in order to avoid leaving the feasible region after applying the genetic operators. The use of special operators that preserve feasibility make this approach highly specific to one application domain rather than providing a general methodology to handle constraints.

Coello [5] used a population-based approach similar to VEGA [26] to handle constraints in single-objective optimization problems. At each generation, the population was split into $m+1$ subpopulations of equal fixed size, where m is the number of constraints of the problem. The remaining subpopulation handles the objective function of the problem and the individuals contained within it are selected based on the unconstrained objective function value. The m remaining subpopulations take one constraint of the problem each as their fitness function. The aim is that each of the subpopulations tries to reach the feasible region corresponding to one individual constraint. By combining these different subpopulations, the approach will reach the feasible region of the problem considering all of its constraints. The fitness assignment scheme of the approach is the following:

if $g_j(\mathbf{x}) < 0.0$	**then**	fitness = $g_j(\mathbf{x})$
else if $v \neq 0$	**then**	fitness = $-v$
else		fitness = $f(\mathbf{x})$

where $g_j(\mathbf{x})$ refers to the jth constraint of the problem, v is the number of violated constraints ($v \leq m$) and $f(\mathbf{x})$ is the value of the objective function of the individual.

As can be seen above, each subpopulation tries to satisfy one single constraint. If the encoded solution does not violate the constraint of its corresponding subpopulation, then the fitness of an individual will be determined by the total number of constraints violated. Finally, if the solution is feasible, then the feasible criterion is to optimize the objective function. Therefore, any feasible individuals will be merged with the subpopulation on charge of optimizing the original (unconstrained) objective function.

The genetic operators are applied to the entire population and it is allowed to every individual in a subpopulation to mate with any other in any subpopulation (including its own, of course). In this way, individuals who satisfy constraints are combined with individuals with a good fitness value. At the end, it is expected to have a population of feasible individuals with high fitness values.

This approach was tested with some engineering problems [5] in which it produced competitive results. It has also been successfully used to solve combinational circuit design problems. The main drawback of this approach is that the number of subpopulations required increases linearly with the number of constraints of the problem. This has some obvious scalability problems when dealing with highly constrained search spaces. Furthermore, it is not clear how to determine appropriate sizes for each of the subpopulations used.

Coello [4] proposed the use of Pareto dominance selection to handle constraints in a genetic algorithm. This is an application of Fonseca and Fleming's Pareto ranking process [9] (called Multi-Objective Genetic Algorithm, or MOGA) to constraint-handling. In this approach, feasible individuals are always ranked higher than infeasible ones. Based on this rank, a fitness value is assigned to each individual. This technique also includes a self-adaptation mechanism that avoids the usual empirical fine-tuning of the main genetic operators.

Coello's approach uses a real-coded GA with universal stochastic sampling selection (to reduce the selection pressure caused by the Pareto ranking process).

This approach has been used to solve some engineering design problems [4] in which it produced very good results. Furthermore, the approach showed great robustness and a relatively low number of fitness function evaluations with respect to traditional penalty functions. Additionally, it does not require any extra parameters. Its main drawback is the computational cost ($O(M^2)$, where M is the population size) derived from the Pareto ranking process.

Coello and Mezura [3] implemented a version of the Niched-Pareto Genetic Algorithm (NPGA) [12] to handle constraints in single-objective optimization problems. The NPGA is a multiobjective optimization approach in which individuals are selected through a tournament based on Pareto dominance. However, unlike the [original] NPGA,

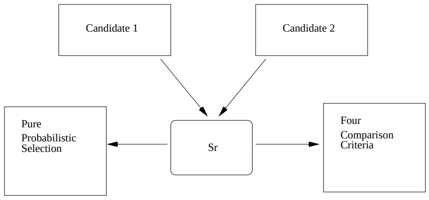

Figure 1. Diagram that illustrates the role of S_r in the selection process of Coello and Mezura's algorithm.

Problem	n	Type of function	ρ	LI	NI	LE	NE
g8	2	non linear	0.8581%	0	2	0	0
g9	7	non linear	0.5199%	0	4	0	0
g11	2	quadratic	0.0973%	0	0	0	1
g12	3	quadratic	4.7697%	0	9^3	0	0
eng1	4	quadratic	2.6859%	6	1	0	0
eng2	4	quadratic	39.6762%	3	1	0	0
eng3	3	quadratic	0.7537%	1	3	0	0
eng4	10	non linear	46.8070%	0	22	0	0

Table 1. Values of ρ for the eight test problems chosen.

Coello and Mezura's approach does not require niches (or fitness sharing [7]) to maintain diversity in the population. The NPGA is a more efficient technique than traditional multiobjective optimization algorithms, because it only uses a sample of the population to estimate Pareto dominance. This is the main advantage of this approach with respect to Coello's [previous] proposal [4].

Note however that Coello and Mezura's approach requires an additional parameter called S_r that controls the diversity of the population. S_r indicates the proportion of parents selected by four comparison criteria described below. The remaining $1 - S_r$ parents will be selected by a pure probabilistic approach (a coin toss with both candidates with a 50% of probability of being chosen). Thus, this mechanism is responsible for keeping infeasible individuals in the population (i.e., the source of diversity that keeps the algorithm from converging to a local optimum too early in the evolutionary process). A graphical illustration of the role of the parameter S_r is shown in Figure 1.

Tournaments in this approach are decided using as a basis four comparison criteria:

If

1. both individuals are feasible, the individual with the higher fitness wins.

2. one is feasible and the other is infeasible, the feasible individual wins.

3. both are infeasible: Nondominance checking is applied (tournament selection as in the NPGA [12]).

4. both are nondominated or dominated, the individual with the lowest amount of constraint violation wins.

This approach has been tested with several benchmark problems and was compared against several types of penalty functions [16]. Results indicated that the approach was robust, efficient and effective. However, it was also found that the approach had scalability problems (its performance degrades as the number of decision variables increases).

4 A Comparative Study

Four techniques were selected from those discussed before to perform a small comparative study that aims to illustrate some practical issues of constraint-handling techniques. The techniques selected are the following: COMOGA [28], the use of VEGA proposed by Coello [5], the NPGA to handle constraints [3] and the approach that uses MOGA [4]. In order to simplify our notation, the last three techniques previously indicated will be called HCVEGA, HCNPGA and HCMOGA, respectively. The algorithmic detail of these four approaches can be found in [17]

To evaluate the performance of the techniques selected, we decided to use some test problems taken from the well-known benchmark proposed in [19] plus four engineering design problems used in [4]. The full description of the eight test functions can be found in [17]:

To get an idea of the difficulty of solving each of these problems, a ρ metric (as suggested by Koziel and Michalewicz [15]) was computed using the following expression:

$$\rho = |F|/|S| \tag{2}$$

where $|F|$ is the number of feasible solutions and $|S|$ is the total number of solutions randomly generated. In this work, $S = 1,000,000$ random solutions.

The different values of ρ for each of the functions chosen are shown in Table 1, where n is the number of decision variables, LI is the number of linear inequalities, NI the number of nonlinear inequalities, LE is the number of linear equalities and NE is the number of nonlinear equalities. It can be clearly seen that problems g8, g9, g11 and eng3 should be the most difficult to solve since they present the lowest value of ρ.

In our comparative study, we used a binary-gray-coded GA with two-point crossover and uniform mutation. Equality constraints were transformed into inequalities using a tolerance value of 0.001 (see [2] for details of this transformation). The number of fitness function evaluations is the same for all the approaches under study ($80,000$). The parameters adopted for each of the methods were the following:

P	COMOGA						
	Optimal	Best	Median	Mean	St. Dev.	Worst	F_p
g8*(99)	0.095825	0.095782	0.094613	0.094343	0.001336	0.089900	0.002883
g9	680.6300573	733.001221	979.748016	983.625328	115.570129	1314.535767	0.011225
g11	0.750	0.749058	0.749787	0.749829	0.000495	0.751753	0.000286
g12	1.000000	0.999999	0.999835	0.999679	0.000930	0.993070	0.070262
eng1	1.728226	1.835381	2.039148	2.030360	0.094810	2.328703	0.000580
eng2	6059.946341	6369.428223	7889.838867	7795.411538	701.363966	9147.520508	0.003989
eng3	0.012681	0.012929	0.014263	0.014362	0.000864	0.017113	0.021144
eng4	5152.636136	6283.198730	6675.126709	6660.455649	126.289250	6968.627441	0.006516

Table 2. Experimental results using COMOGA with the 8 test problems. The symbol "*" and the number in parentheses "(n)"means that in n runs feasible solutions were found (of the 100 runs performed)

P	HCVEGA						
	Optimal	Best	Median	Mean	St. Dev.	Worst	F_p
g8	0.095825	0.095826	0.095826	0.095826	0.000000	0.095826	0.393393
g9	680.6300573	693.642029	736.190613	739.306726	25.164813	806.855469	0.046137
g11	0.750	0.749621	0.812369	0.798690	0.025821	0.847242	0.011206
g12	1.000000	1.000000	1.000000	1.000000	0.000000	1.000000	0.516501
eng1	1.728226	1.726772	1.735865	1.736439	0.005927	1.769726	0.346061
eng2	6059.946341	6064.723633	6238.489746	6259.963745	170.254024	6820.944824	0.425354
eng3	0.012681	0.012688	0.012789	0.012886	0.000209	0.013784	0.250408
eng4	5152.636136	5327.418457	5453.446045	5455.871895	56.744241	5569.240723	0.676408

Table 3. Experimental results using HCVEGA to handle constraints with the 8 test problems.

COMOGA: Population Size = 200, crossover rate = 1.0, mutation rate = 0.05, desired proportion of feasible solutions = 10 %, $\epsilon = 0.01$ **HCVEGA**: Population Size = 200, number of generations = 400, crossover rate = 0.6, mutation rate = 0.05, tournament size= 5 **HCNPGA**: Population Size = 200, number of generations = 400, crossover rate = 0.6, mutation rate = 0.05, size of sample of the population = 10, selection Ratio = 0.8 **HCMOGA**: Population Size = 200, number of generations = 400, crossover rate = 0.6, mutation rate = 0.05

A total of 100 runs per technique per problem were performed. Statistical results are presented in Tables 2, 3, 4 and 5, where $0.0 \leq F_p \leq 1.0$ is the average rate of feasible solutions found during a single run (with respect to the full population).

5 Discussion of Results

After the experimental phase, several results were obtained and they are discussed next: **Quality of the results**: HCNPGA gave the best (or tied with the best) results in 6 (g8, g9, , g11, g12, eng2 and eng4) problems and in 5 of them it reached the global optimum or best known solution (g8, g11, g12, eng1 and eng2). HCMOGA is superior (or is tied) in 4 of them (g8, g11, g12 and eng3) and it reached the global optimum in 4 (g8, g11, g12 and eng3). HCVEGA got better results only in 3 problems (g8, g12 and eng1) and it found the global optimum in 4 (g8, g11, g12 and eng1). COMOGA was clearly surpassed because it did not give the best results in any given problem. However it reached the global optimum in problem (g11). **Consistency**: The statistics indicate that HCNPGA presented the lowest standard deviation (all but g11 and eng1). Also, the best median and average solutions in 7 problems were found by HCNPGA (all problems but g11). The second best approach was HCMOGA which produced the best results in 3 problems (g8,g11 and g12). HCVEGA found better results than HCMOGA in all four engineering problems. COMOGA shown a similar behavior than HCNPGA in problem g11. **Diversity**: An utopical behavior for an ideal constraint handling technique is defined in our case as keeping at all times half of the population with feasible solutions and the other half with infeasible ones. However, in practice such balance may be very difficult to achieve. Therefore, we provide a relative comparison among the approaches under study. The less balanced approach based on the number of feasible and infeasible solutions in the population during all the evolutionary process was COMOGA. The remaining three approaches shown a similar behavior, not necessarily closer to the ideal performance but good enough to find good results.

Some issues can be stated based on the previous study: It seems that all three multiobjective concepts (Pareto dominance, Pareto ranking and population-based) are able to maintain an acceptable balance between feasible and infeasible solutions. Also, the four approaches could deal with different feasible region sizes and shapes. It is important to note that all the compared approaches always found the feasible region of the search space in all the proposed problems. Pareto dominance as a selection criterion has proved to give better results (in terms of optimality) than Pareto Ranking or a population-based approach. Finally, the overall results of the COMOGA's steady-state GA suggests that it is not a good option to solve nonlinear global optimization

P	HCNPGA						
	Optimal	Best	Median	Mean	St. Dev.	Worst	F_p
g8	0.095825	0.095826	0.095826	0.095826	0.000000	0.095826	0.424794
g9	680.6300573	680.951172	682.121002	682.335237	0.836150	684.816284	0.240264
g11	0.750	0.749001	0.749613	0.753909	0.012147	0.832940	0.025842
g12	1.000000	1.000000	1.000000	1.000000	0.000000	1.000000	0.446223
eng1	1.728226	1.727014	1.751112	1.766413	0.043733	1.978103	0.261526
eng2	6059.946341	6059.926270	6127.618408	6172.527373	123.897547	6845.770508	0.330693
eng3	0.012681	0.012683	0.012736	0.012752	0.000062	0.013132	0.104895
eng4	5152.636136	5179.740723	5256.108154	5259.013174	37.658930	5362.890625	0.503566

Table 4. Experimental results using HCNPGA to handle constraints with the 8 test problems.

problems. This is mainly due to its high selection pressure which tends to produce premature convergence. Although not conclusive, this study seems to indicate that Pareto dominance, Pareto ranking and population-based mechanisms are promising approaches to handle constraints. HCNPGA looks like the most robust approach from the four compared in this study. This approach aims to obtain a better performance out of Pareto dominance used as a selection criteria. Also, the use of Pareto Ranking in HCMOGA gave good results. Furthermore, HCVEGA found better statistical results than HCMOGA in some problems. Finally, these results also seem to suggest that a traditional (i.e., generational) GA performs better in optimization problems than nongenerational GAs. An open question is if the advantages of each of these techniques can be combined into a single approach.

6 Conclusions and Future Work

A set of constraint-handling techniques based on multiobjective concepts were presented in this paper. In each case, advantages and disadvantages were discussed. We also presented a comparative study in which four of the techniques discussed were implemented and evaluated using eight test functions. Our results provided some insights regarding the behavior of each type of technique. Note however, that comparisons with respect to traditional penalty functions [24, 27] and with the most competitive constraint-handling techniques used with EAs (e.g., stochastic ranking [25], the homomorphous maps [15], and the adaptive segregational constrained handling evolutionary algorithm (ASCHEA) [11]) are still lacking.

The results obtained seem to indicate that techniques based on multiobjective optimization can properly deal with constrained search spaces. However, such results also seem to indicate that additional mechanisms should be used to improve the effectiveness of these approaches, since they have obvious difficulties to reach the global optimum in some of the test functions used in the current study.

Acknowledgments

The first author acknowledges support from the mexican Consejo Nacional de Ciencia y Tecnología (CONACyT) through a scholarship to pursue graduate studies at CINVESTAV-IPN's Electrical Engineering Department. The second author acknowledges support from (CONACyT) through project number 32999-A. Both authors acknowledge the valuable help provided by Dr. Patrick D. Surry.

References

[1] E. Camponogara and S. N. Talukdar. A Genetic Algorithm for Constrained and Multiobjective Optimization. In J. T. Alander, editor, *3rd Nordic Workshop on Genetic Algorithms and Their Applications (3NWGA)*, pages 49–62, Vaasa, Finland, August 1997. University of Vaasa.

[2] C. A. C. Coello. Theoretical and Numerical Constraint Handling Techniques used with Evolutionary Algorithms: A Survey of the State of the Art. *Computer Methods in Applied Mechanics and Engineering*, 191(11-12):1245–1287, January 2002.

[3] C. A. C. Coello and E. Mezura-Montes. Handling Constraints in Genetic Algorithms Using Dominance-Based Tournaments. In I. Parmee, editor, *Proceedings of the Fifth International Conference on Adaptive Computing Design and Manufacture (ACDM 2002)*, volume 5, pages 273–284, University of Exeter, Devon, UK, April 2002. Springer-Verlag.

[4] C. A. Coello Coello. Constraint-handling using an evolutionary multiobjective optimization technique. *Civil Engineering and Environmental Systems*, 17:319–346, 2000.

[5] C. A. Coello Coello. Treating Constraints as Objectives for Single-Objective Evolutionary Optimization. *Engineering Optimization*, 32(3):275–308, 2000.

[6] C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York, June 2002. ISBN 0-3064-6762-3.

[7] K. Deb and D. E. Goldberg. An Investigation of Niche and Species Formation in Genetic Function Optimization. In J. D. Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 42–50, San Mateo, California, June 1989. George Mason University, Morgan Kaufmann Publishers.

Problem	HCMOGA						
	Optimal	Best	Median	Mean	St. Dev.	Worst	F_p
g8	0.095825	0.095825	0.095825	0.095825	0.000000	0.095825	0.074362
g9	680.6300573	681.707947	689.955780	692.966086	10.957636	734.002258	0.048562
g11	0.750	0.749001	0.749155	0.749393	0.000595	0.752445	0.016913
g12	1.000000	1.000000	1.000000	1.000000	0.000000	1.000000	0.090064
eng1	1.728226	1.729384	1.791587	1.825219	0.079812	2.126669	0.064106
eng2	6059.946341	6066.969727	6561.483154	6629.064048	385.110736	7547.403320	0.452672
eng3	0.012681	0.012680	0.012815	0.012960	0.000363	0.014754	0.047492
eng4	5152.636136	5336.618652	5745.238281	5748.839526	210.696096	6474.041992	0.603598

Table 5. Experimental results using HCMOGA to handle constraints with the 8 test problems.

[8] F. Y. Edgeworth. *Mathematical Physics*. P. Keagan, London, England, 1881.

[9] C. M. Fonseca and P. J. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In S. Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, San Mateo, California, 1993. University of Illinois at Urbana-Champaign, Morgan Kaufmann Publishers.

[10] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Co., Reading, Massachusetts, 1989.

[11] S. B. Hamida and M. Schoenauer. ASCHEA: New Results Using Adaptive Segregational Constraint Handling. In *Proceedings of the Congress on Evolutionary Computation 2002 (CEC'2002)*, volume 1, pages 884–889, Piscataway, New Jersey, May 2002. IEEE Service Center.

[12] J. Horn, N. Nafpliotis, and D. E. Goldberg. A Niched Pareto Genetic Algorithm for Multiobjective Optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, volume 1, pages 82–87, Piscataway, New Jersey, June 1994. IEEE Service Center.

[13] F. Jiménez, A. Gómez-Skarmeta, and G. Sánchez. How Evolutionary Multi-objective Optimization can be used for Goals and Priorities based Optimization. In E. Alba, F. Fernández, J. Gómez, F. Herrera, J. Hidalgo, J. Lanchares, J. Merelo, and J. Sánchez, editors, *Primer Congreso Espaõl de Algoritmos Evolutivos y Bioinspirados (AEB'02)*, pages 460–465, Mérida España, 2002. Universidad de la Extremadura, España.

[14] F. Jiménez and J. L. Verdegay. Evolutionary techniques for constrained optimization problems. In H.-J. Zimmermann, editor, *7th European Congress on Intelligent Techniques and Soft Computing (EUFIT'99)*, Aachen, Germany, 1999. Verlag Mainz. ISBN 3-89653-808-X.

[15] S. Koziel and Z. Michalewicz. Evolutionary Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization. *Evolutionary Computation*, 7(1):19–44, 1999.

[16] E. Mezura-Montes. Uso de la Técnica Multiobjetivo NPGA para el Manejo de Restricciones en Algoritmos Genéticos. Master's thesis, Universidad Veracruzana, Xalapa, México, 2001. (In Spanish).

[17] E. Mezura-Montes and C. A. C. Coello. A Numerical Comparison of some Multiobjective-Based Techniques to Handle Constraints in Genetic Algorithms. Technical Report EVOCINV-03-2002, Evolutionary Computation Group at CINVESTAV, Sección de Computación, Departamento de Ingeniería Eléctrica, CINVESTAV-IPN, México D.F., México, 2002. Available in the Constraint Handling Techniques in Evolutionary Algorithms Repository at http://www.cs.cinvestav.mx/˜constraint/.

[18] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, third edition, 1996.

[19] Z. Michalewicz and M. Schoenauer. Evolutionary Algorithms for Constrained Parameter Optimization Problems. *Evolutionary Computation*, 4(1):1–32, 1996.

[20] V. Pareto. *Cours D'Economie Politique*, volume I and II. F. Rouge, Lausanne, 1896.

[21] I. C. Parmee and G. Purchase. The development of a directed genetic search technique for heavily constrained design spaces. In I. C. Parmee, editor, *Adaptive Computing in Engineering Design and Control-'94*, pages 97–102, Plymouth, UK, 1994. University of Plymouth.

[22] T. Ray. Constraint Robust Optimal Design using a Multiobjective Evolutionary Algorithm. In *Proceedings of the Congress on Evolutionary Computation 2002 (CEC'2002)*, volume 1, pages 419–424, Piscataway, New Jersey, May 2002. IEEE Service Center.

[23] T. Ray, T. Kang, and S. K. Chye. An Evolutionary Algorithm for Constrained Optimization. In D. Whitley, D. Goldberg, E. Cantú-Paz, L. Spector, I. Parmee, and H.-G. Beyer, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2000)*, pages 771–777, San Francisco, California, July 2000. Morgan Kaufmann.

[24] J. T. Richardson, M. R. Palmer, G. Liepins, and M. Hilliard. Some Guidelines for Genetic Algorithms with Penalty Functions. In J. D. Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms (ICGA-89)*, pages 191–197, San Mateo, California, June 1989. George Mason University, Morgan Kaufmann Publishers.

[25] T. P. Runarsson and X. Yao. Stochastic Ranking for Constrained Evolutionary Optimization. *IEEE Transactions on Evolutionary Computation*, 4(3):284–294, September 2000.

[26] J. D. Schaffer. Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, pages 93–100. Lawrence Erlbaum, 1985.

[27] A. E. Smith and D. W. Coit. Constraint Handling Techniques—Penalty Functions. In T. Bäck, D. B. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation*, chapter C 5.2. Oxford University Press and Institute of Physics Publishing, 1997.

[28] P. D. Surry and N. J. Radcliffe. The COMOGA Method: Constrained Optimisation by Multiobjective Genetic Algorithms. *Control and Cybernetics*, 26(3):391–412, 1997.