A Numerical Comparison of some Multiobjective-Based Techniques to Handle Constraints in Genetic Algorithms. *

Efrén Mezura-Montes and Carlos A. Coello Coello

CINVESTAV-IPN
Evolutionary Computation Group
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 discusses the general multiobjective optimization concepts that can and have been used to incorporate constraints of any type (linear, nonlinear, equality and inequality) into the fitness function of a genetic algorithm used for global optimization. We also describe several approaches currently reported in the literature and four of them are compared using several test functions. The results obtained are widely discussed and some ideas about how to devise new approaches based on multiobjective optimization concepts are also briefly analyzed.

1 Introduction

Evolutionary Algorithms (EAs) are heuristics that have been successfully applied in a wide set of areas [14, 30, 1, 5, 19, 7, 4, 32, 45, 44, 28], 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 [9, 31].

*Technical Report EVOCINV-03-2002, Evolutionary Computation Group at CINVESTAV, Sección de Computación, Departamento de Ingeniería Eléctrica, CINVESTAV-IPN, México, September 2002.

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.[41, 9].

Among the several approaches that have been proposed as an alternative to the use of penalty functiones, 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.

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 [31, 11]. 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:

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

subject to:

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

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

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 non linear).

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 constaint that satisfies $g_i(\vec{x}) = 0$, then we will say that is active at $\vec{x}$. All equality constraints h_j (regardless of the value of $\vec{x}$ used) are considered active at all points of $\mathcal{F}$.

Now, we will define some basic concepts from multiobjective optimization.

Definition 1 **(General Multiobjective Optimization Problem):** *Find the vector* $\vec{x}^* = [x_1^*, x_2^*, \ldots, x_n^*]^T$ *which will satisfy the* m *inequality constraints:*

$$g_i(\vec{x}) \geq 0 \quad i = 1, 2, \ldots, m \tag{4}$$

the p *equality constraints*

$$h_i(\vec{x}) = 0 \quad i = 1, 2, \ldots, p \tag{5}$$

and will optimize the vector function

$$\vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \ldots, f_k(\vec{x})]^T \tag{6}$$

where $\vec{x} = [x_1, x_2, \ldots, x_n]^T$ *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 [17] and later generalized by Vilfredo Pareto (in 1896) [33]. This notion is normally referred to as "Pareto optimality" and is defined next.

Definition 2 **(Pareto Optimality:):** *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,*

$$\forall_{i \in I}(f_i(\vec{x}) = f_i(\vec{x}^*)) \tag{7}$$

or, there is at least one $i \in I$ *such that*

$$f_i(\vec{x}) > f_i(\vec{x}^*) \tag{8}$$

□

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 the following:

Definition 3 **(Pareto Dominance):** *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.,* $\forall i \in \{1, \ldots, k\},\ u_i \leq v_i \wedge \exists i \in \{1, \ldots, k\} : u_i < v_i$. □

Definition 4 **(Pareto Optimal Set):** *For a given multiobjective optimization problem,* $\vec{f}(x)$*, the Pareto optimal set (*$\mathcal{P}^*$*) is defined as:*

$$\mathcal{P}^* := \{x \in \mathcal{F} \mid \neg \exists\, x' \in \Omega \;\; \vec{f}(x') \preceq \vec{f}(x)\}. \tag{9}$$

□

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 [14] to the new vector $\bar{v} = (f(\vec{x}), f_1(\vec{x}), \dots, f_m(\vec{x}))$, where $f_1(\vec{x}), \dots, f_m(\vec{x})$ 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 [20] 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) [40].

In order to sample the feasible region of the search space widely enough to reach the global optima it is necessary to maintain a balance between feasible and infeasible solutions. If this diversity is not reached, the search will focus only in one area of the feasible region. Thus, it will lead to a local optima solution.

A multiobjective optimization technique aims to find a set of trade-off solutions which are considered good in all the objectives to be optimized. In global nonlinear optimization, the main goal is to find the global optimum. Therefore, some changes must be done to those approaches in order to adapt them to the new goal. 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 proceed to discuss the different approaches that have been proposed adopting the three main ideas previously indicated.

3.1 Pareto Set and Line Search

Camponogara & Talukdar [6] 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:

$$\Phi(\mathbf{x}) = \sum_{i=1}^{n} \max(0, g_i(\mathbf{x})) \tag{10}$$

Equation (10) tries to minimize the total amount of constraint violation of a solution (i.e., it tries to make it feasible). At each generation of the process, several Pareto sets are generated. An operator that substitutes crossover takes two Pareto sets S_i and S_j where $i < j$ and two solutions $x_i \in S_i$ and $x_j \in S_j$ where x_i dominates x_j. With these two points a search direction is defined using:

$$d = \frac{(x_i - x_j)}{|x_i - x_j|} \tag{11}$$

Line search begins by proyecting d over one variable axis on decision variable space in order to find a new solution x which dominates both x_i and x_j. At pre-defined intervals, the worst half of the population is replaced with new ramdom soutions 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. Also, it is not clear what is the impact of the segment chosen to search on the overall performance of the algorithm.

3.2 Min-Max

An approach similar to a min-max formulation used in multiobjective optimization [8] combined with tournament selection was proposed by Jiménez and Verdegay [25]. 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 on a 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 [43]. 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.

3.3 Pareto Ranking and Domain Knowledge

Ray et al. [37] 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 to ensure better constraint satisfaction in the offspring generated and a selection process that eliminates weaknesses in any of these spaces. To maintain diversity, a niche mechanism based on Euclidean distances is used. This approach can solve both constrained or unconstrained optimization problems with one or several objective functions.

Three different matrices must be computed: the objective matrix (containing the objective function values of each solution), the constraint matrix (that containts the constraint values of each individual), and a matrix that combines the two previous. These matrices are illustrated in equations (12), (13) and (14).

$$\begin{pmatrix} f_{11} & f_{12} & \cdots & f_{1k} \\ f_{21} & f_{22} & \cdots & f_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ f_{M1} & f_{M2} & \cdots & f_{Mk} \end{pmatrix} \tag{12}$$

$$\begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1s} \\ c_{21} & c_{22} & \cdots & c_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ c_{M1} & c_{M2} & \cdots & c_{Ms} \end{pmatrix} \tag{13}$$

$$\begin{pmatrix} f_{11} & f_{12} & \cdots & f_{1k} & c_{11} & c_{12} & \cdots & c_{1s} \\ f_{21} & f_{22} & \cdots & f_{2k} & c_{21} & c_{22} & \cdots & c_{2s} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ f_{M1} & f_{M2} & \cdots & f_{Mk} & c_{M1} & c_{M2} & \cdots & c_{Ms} \end{pmatrix} \tag{14}$$

The mating restrictions used by this method are based on the information that each individual has about its own feasibility. Such a scheme is based on an idea proposed by Hinterding and Michalewicz [22].

The main advantage of this approach is that it requires a very low number of fitness function evaluations (between 2% and 10% of the number of evaluations required by the homomorphous maps of Koziel and Michalewicz [27], which is one of the best constraint-handling techniques known to date). The technique has some problems to reach the global optima, but it produces very good approximations considering its low computation cost. The main drawback of the approach is that its implementation is considerably more complex than any of the other techniques previously discussed.

3.4 Pareto Dominance and Preselection

Jiménez et al. [24] proposed an algorithm that uses Pareto dominance inside a preselection scheme to solve several types of optimization problems (multi-objective, 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 nonuniform).

The authors argue that this preselection 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.

3.5 Pareto Ranking and Robust Optimization

Ray [36] explored an extension of his previous work on constraint-handling [37] in which the emphasis was robustness. A robust optimized solution 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. Thus, a new matrix called "modified constraint matrix" is used to replace both the constraint matrix and the combined matrix of Ray's original proposal (see equations 13 and 14). An example of this modified constraint matrix is the following:

$$\begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1s} & c_{1s+1} & c_{1s+2} & \cdots & c_{12s} \\ c_{21} & c_{22} & \cdots & c_{2s} & c_{2s+2} & c_{2s+2} & \cdots & c_{22s} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{M1} & c_{M2} & \cdots & c_{Ms} & c_{Ms+1} & c_{Ms+2} & \cdots & c_{M2s} \end{pmatrix} \tag{15}$$

where $c_{is+1}, c_{is+2} \ldots c_{i2s}$ denotes the number of violations of the first constraint among k neighbors and so on. 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 p_1 and p_2. One parent (p_1) is selected based on a crowding factor using roulette wheel selection. The partner of this individual (p_2) is the result of the competition between two individuals (A and B) picked up by a roulette wheel selection process. To decide between A and B in order to obtain p_2 the following criteria are adopted:

1. If A is feasible and B is not: Select A as the partner p_2.
2. If B is feasible and A is not: Select B as the partner p_2.
3. If both A and B are feasible: Select one with a minimum **Rank Objective**. If the **Rank Objective** of A and B are the same, select one with minimum number of neighbors in the parametric space as a mate.
4. If both A and B are infeasible: Select A or B with a minimum **Rank Constraint**. If the **Rank Constraints** are the same, randomly select between A and B.

A real-coded GA with Simulated Binary Crossover [15] was used to implement this technique. The results reported in two well-known design problems [36] showed that the proposed approach did not reach solutions as good as the other techniques against which it was compared, but it turned out to be less sensitive to parametric variations, which was the main goal of the approach. In constrast, the other techniques analyzed showed significant changes when the parameters were perturbed. The main drawback of this approach is, again, its relative complexity (i.e., its difficulty to implement it), and it would also be desirable that the approach is further refined so that it can get closer to the global optimum than the current available version.

3.6 COMOGA

Surry & Radcliffe [42] used a combination of the Vector Evaluated Genetic Algorithm (VEGA) [40] 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., an 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 begining 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 requiere 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 [42].

The algorithm of COMOGA is the following [42]:

Begin
Calculate constraint violation for all solutions.
Rank solutions based on constraint violation (nondominance checking).
Evaluate the fitness of solutions.
Select a P_{cost} proportion of parents based on
fitness and the remaining $1 - P_{cost}$ based on constraint ranking.
Apply genetic operators
Adjust P_{cost}: Decreasing it favors feasible solutions; Increasing it
favors lower cost solutions (high fitness)
End

3.7 VEGA

Parmee & Purchase [34] proposed to use VEGA [40] 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. 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. Note that this approach does not really use Pareto dominance or any other multiobjective optimization concepts to exploit the search space. Instead, it uses VEGA just to reach the feasible region. 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 [12] used a population-based approach similar to VEGA [40] 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 additional 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 algorithm of this approach is the following:

Begin
Create M random solutions for the initial population.
Split the population into $m + 1$ sub-populations
Evaluate all M individuals
Assign a fitness value to all M individuals depending of their
corresponding subpopulation.
While stopping criterion is not satisfied **Do**
Insert the best individual of the current
population into the next population
While the next population is not full **Do**

Select 2 parents p_1 and p_2 based on **tournament selection with n candidates** from all the M individuals of the main population
Apply crossover to p_1 and p_2 to generate 2 offspring c_1 and c_2
Apply mutation to offspring c_1 and c_2
Insert c_1 and c_2 into the next population
End While
Split the population into $m+1$ subpopulations
Evaluate all M new individuals
Assign a fitness value to all M individuals depending of their corresponding subpopulation.
End While
End

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 [12] in which it produced competitive results. It has also been successfully used to solve combinational circuit design problems [13]. 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.

3.8 MOGA

Coello [11] 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 [18] (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).

The algorithm of this approach is the following:

Begin
  Create M random solutions for the initial population.
  Evaluate the M individuals in the population.
  Calculate the rank for each of the M individuals in the population.
  Assign a fitness value to all M individuals depending on rank
  While stopping criterion is not satisfied **Do**
    Insert the best individual of the current
    population into the next population
    While the next population is not full **Do**
      Select 2 parents p_1 and p_2 using **Universal Stochastic Sampling**
      Apply crossover to p_1 and p_2 to generate 2 offspring c_1 and c_2
      Apply mutation to offspring c_1 and c_2
      Insert c_1 and c_2 into the next population
    End While
    Evaluate the M new individuals in the population
    Calculate the rank for each of the M individuals in the population.
    Assign a fitness value to all M individuals depending on rank
  End While
End

To compute the **rank** of an individual $\vec{x}_i$ this approach uses the following procedure:

- Evaluate:
$$\text{rank}(\vec{x}_i) = \text{count}(\vec{x}_i) + 1 \tag{16}$$
  where $\text{count}(\vec{x}_i)$ is computed according to the following rules:
  1. Compare $\vec{x}_i$ against every other individual in the population. Assuming pairwise comparisons, we will call $\vec{x}_j$ ($j = 1, \ldots, pop\_size$ and $j \neq i$) the other individual against which x_i is being compared at any given time.
  2. Initialize $\text{count}(\vec{x}_i)$(for $i = 1, \ldots, pop\_size$) to zero.
  3. If both $\vec{x}_i$ and $\vec{x}_j$ are feasible, then both are given a rank of zero and $\text{count}(\vec{x}_i)$ remains without changes.
  4. If $\vec{x}_i$ is infeasible and $\vec{x}_j$ is feasible, then $\text{count}(\vec{x}_i)$ is incremented by one.

5. If both $\vec{x}_i$ and $\vec{x}_j$ are infeasible, but $\vec{x}_i$ violates more constraints than $\vec{x}_j$, then count($\vec{x}_i$) is incremented by one.
6. If both $\vec{x}_i$ and $\vec{x}_j$ are infeasible, and both violate the same number of constraints, but $\vec{x}_i$ has a total amount of constraint violation larger than the constraint violation of $\vec{x}_j$, then count($\vec{x}_i$) is incremented by one.

   If any constraint $g_k(\vec{x})$ ($k = 1, \ldots, m$, where m is the total number of constraints) is considered satisfied if $g_i(\vec{x}) \leq 0$, then, the total amount of constraint violation for an individual $\vec{x}_i$ (denoted as coef($\vec{x}_i$)) is given by:

$$\text{coef}(\vec{x}_i) = \sum_{k=1}^{p} g_k(\vec{x}_i) \quad \text{for all } g_k(\vec{x}_i) > 0 \tag{17}$$

To compute **fitness**, the following rules are adopted:

1. If $\vec{x}_i$ is feasible, then rank($\vec{x}_i$) = $fitness(\vec{x}_i)$, else
2. rank($\vec{x}_i$) = $\frac{1}{\text{rank}(\vec{x}_i)}$

Then, individuals are selected based on rank($\vec{x}_i$) (stochastic universal sampling is used). Note that the values produced by $fitness(\vec{x}_i)$ must be normalized to ensure that the rank of feasible individuals is always higher than the rank of infeasible ones.

This approach has been used to solve some engineering design problems [11] 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.

3.9 NPGA

Coello and Mezura [10] implemented a version of the Niched-Pareto Genetic Algorithm (NPGA) [23] 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 NPGA, Coello and Mezura's approach does not require niches (or fitnes sharing [16]) to maintain diversity in the population. The NPGA is a more efficient technique than traditional multiobjective optimization algorithms, since it does not compare every individual in the population with respect to each other (as in traditional Pareto ranking), but uses only a sample of the population to estimate Pareto dominance. This is the main advantage of this approach with respect to Coello's proposal [11].

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

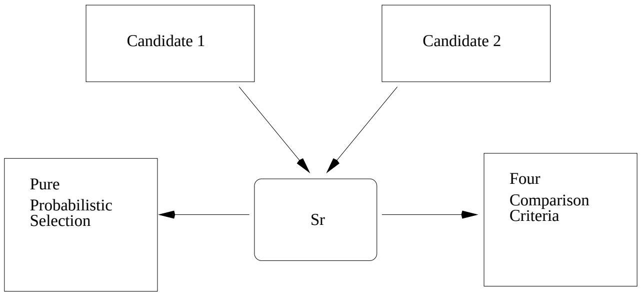

Figure 1: Diagram that illustrates the role of S_r in the selection process of Coello and Mezura's algorithm.

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. 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 [23]).
4. both are nondominated or dominated, the individual with the lowest amount of constraint violation wins.

The algorithm of this approach is the following:

Begin
  Create M random solutions for the initial population.
  Evaluate the M individuals in the population.
  While stopping criterion is not satisfied **Do**
    Insert the best individual of the current
    population into the next population
    While the next population is not full **Do**
      Select 2 parents p_1 and p_2 based on $\mathbf{S_r}$ value
      Apply crossover to p_1 and p_2 to generate 2 offspring c_1 and c_2
      Apply mutation to offspring c_1 and c_2
      Insert c_1 and c_2 into the next population
    End While

Evaluate the M new individuals in the population
End While
End

This approach has been tested with several benchmark problems and was compared against several types of penalty functions [29]. 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 varibles 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 [42], the use of VEGA proposed by Coello [12], the NPGA to handle constraints [10] and the approach that uses MOGA [11]. In order to simplify our notation, the last three techniques previously indicated will be called HCVEGA, HCNPGA and HCMOGA, respectively.

To evaluate the performance of the techniques selected, we decided to use the well-known benchmark proposed in [31] plus four engineering design problems used in [11]. The full description of the seventeen test functions is the following:

1. **Problem 1: (g01)**:

   Minimize:

$$f(\vec{x}) = 5\sum_{i=1}^{4} x_i - 5\sum_{i=1}^{4} x_i^2 - \sum_{i=5}^{13} x_i \tag{18}$$

   subject to:

$$\begin{aligned}
g_1(\vec{x}) &= 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \leq 0\\
g_2(\vec{x}) &= 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \leq 0\\
g_3(\vec{x}) &= 2x_2 + 2x_3 + x_{10} + x_{12} - 10 \leq 0\\
g_4(\vec{x}) &= -8x_1 + x_{10} \leq 0\\
g_5(\vec{x}) &= -8x_2 + x_{11} \leq 0\\
g_6(\vec{x}) &= -8x_3 + x_{12} \leq 0\\
g_7(\vec{x}) &= -2x_4 - x_5 + x_{10} \leq 0\\
g_8(\vec{x}) &= -2x_6 - x_7 + x_{11} \leq 0\\
g_9(\vec{x}) &= -2x_8 - x_9 + x_{12} \leq 0
\end{aligned} \tag{19}$$

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

2. **Problem 2: (g02)**:
   Maximize:

$$f(\vec{x}) = \left| \frac{\sum_{i=1}^{n} \cos^4(x_i) - 2\prod_{i=1}^{n} \cos^2(x_i)}{\sqrt{\sum_{i=1}^{n} i x_i^2}} \right| \tag{20}$$

   subject to:

$$\begin{aligned} g_1(\vec{x}) &= 0.75 - \prod_{i=1}^{n} x_i \leq 0 \\ g_2(\vec{x}) &= \sum_{i=1}^{n} x_i - 7.5n \leq 0 \end{aligned} \tag{21}$$

   where $n = 20$ and $0 \leq x_i \leq 10$ $(i = 1, \ldots, n)$. The global maximum is unknown; the best reported solution is [39] $f(x^*) = 0.803619$. Constraint g_1 is close to being active ($g_1 = -10^{-8}$).

3. **Problem 3: (g03)**:
   Maximize:

$$f(\vec{x}) = \left(\sqrt{n}\right)^n \prod_{i=1}^{n} x_i \tag{22}$$

   subject to:

$$h(\vec{x}) = \sum_{i=1}^{n} x_i^2 - 1 = 0 \tag{23}$$

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

4. **Problem 4: (g04)**:
   Minimize:

$$f(\vec{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141 \tag{24}$$

   subject to:

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

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

5. **Problem 5: (g5)**

   Minimize:

$$f(\vec{x}) = 3x_1 + 0.000001x_1^3 + 2x_2 + (0.000002/3)x_2^3 \tag{26}$$

   subject to:

$$\begin{aligned}
g_1(\vec{x}) &= -x_4 + x_3 - 0.55 \leq 0 \\
g_2(\vec{x}) &= -x_3 + x_4 - 0.55 \leq 0 \\
h_3(\vec{x}) &= 1000\sin(-x_3 - 0.25) + 1000\sin(-x_4 - 0.25) + 894.8 - x_1 = 0 \\
h_4(\vec{x}) &= 1000\sin(x_3 - 0.25) + 1000\sin(x_3 - x_4 - 0.25) + 894.8 - x_2 = 0 \\
h_5(\vec{x}) &= 1000\sin(x_4 - 0.25) + 1000\sin(x_4 - x_3 - 0.25) + 1294.8 = 0
\end{aligned} \tag{27}$$

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

6. **Problem 6: (g6)**

   Minimize:

$$f(\vec{x}) = (x_1 - 10)^3 + (x_2 - 20)^3 \tag{28}$$

   subject to:

$$\begin{aligned}
g_1(\vec{x}) &= -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \leq 0 \\
g_2(\vec{x}) &= (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \leq 0
\end{aligned} \tag{29}$$

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

7. **Problem 7: (g7)**

   Minimize:

$$\begin{aligned}
f(\vec{x}) = \;& x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 \\
& +2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 \\
& +(x_{10} - 7)^2 + 45
\end{aligned} \tag{30}$$

   Subject to:

$$
\begin{aligned}
g_1(\vec{x}) &= -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \leq 0 \\
g_2(\vec{x}) &= 10x_1 - 8x_2 - 17x_7 + 2x_8 \leq 0 \\
g_3(\vec{x}) &= -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \leq 0 \\
g_4(\vec{x}) &= 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \leq 0 \\
g_5(\vec{x}) &= 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \leq 0 \\
g_6(\vec{x}) &= x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \leq 0 \\
g_7(\vec{x}) &= 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 - 30 \leq 0 \\
g_8(\vec{x}) &= -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \leq 0 \qquad (31)
\end{aligned}
$$

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

8. **Problem 8: (g8)**

   Maximize:

$$
f(\vec{x}) = \frac{\sin^3(2\pi x_1)\sin(2\pi x_2)}{x_1^3(x_1 + x_2)} \qquad (32)
$$

   subject to:

$$
\begin{aligned}
g_1(\vec{x}) &= x_1^2 - x_2 + 1 \leq 0 \\
g_2(\vec{x}) &= 1 - x_1 + (x_2 - 4)^2 \leq 0 \qquad (33)
\end{aligned}
$$

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

9. **Problem 9: (g9)**

   Minimize:

$$
\begin{aligned}
f(\vec{x}) = \; & (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 \\
& +10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7 \qquad (34)
\end{aligned}
$$

   subject to:

$$
\begin{aligned}
g_1(\vec{x}) &= -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \leq 0 \\
g_2(\vec{x}) &= -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \leq 0 \\
g_3(\vec{x}) &= -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \leq 0 \\
g_4(\vec{x}) &= 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \leq 0 \qquad (35)
\end{aligned}
$$

10. **Problem 10: (g10)** Minimize:

$$f(\vec{x}) = x_1 + x_2 + x_3 \tag{36}$$

Subject to:

$$\begin{aligned} g_1(\vec{x}) &= -1 + 0.0025(x_4 + x_6) \leq 0 \\ g_2(\vec{x}) &= -1 + 0.0025(x_5 + x_7 - x_4) \leq 0 \\ g_3(\vec{x}) &= -1 + 0.01(x_8 - x_5) \leq 0 \\ g_4(\vec{x}) &= -x_1 x_6 + 833.33252 x_4 + 100 x_1 - 83333.333 \leq 0 \\ g_5(\vec{x}) &= -x_2 x_7 + 1250 x_5 + x_2 x_4 - 1250 x_4 \leq 0 \\ g_6(\vec{x}) &= -x_3 x_8 + 1250000 + x_3 x_5 - 2500 x_5 \leq 0 \end{aligned} \tag{37}$$

where $100 \leq x_1 \leq 10000$, $1000 \leq x_i \leq 10000$, $(i = 2, 3)$, $10 \leq x_i \leq 1000$, $(i = 4, \ldots, 8)$. The global optimum is: $x^* = (579.3167, 1359.943, 5110.071, 182.0174, 295.5985, 217.9799, 286.4162, 395.5979)$, where $f(x^*) = 7049.3307$. g_1, g_2 and g_3 are active.

11. **Problem 11: (g11)**
Minimize:

$$f(\vec{x}) = x_1^2 + (x_2 - 1)^2 \tag{38}$$

subject to:

$$h(\vec{x}) = x_2 - x_1^2 = 0 \tag{39}$$

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

12. **Problem 12: (g12)**
Maximize:

$$f(\vec{x}) = \frac{100 - (x_1 - 5)^2 - (x_2 - 5)^2 - (x_3 - 5)^2}{100} \tag{40}$$

Subject to:

$$g_1(\vec{x}) = (x_1 - p)^2 + (x_2 - q)^2 + (x_3 - r)^2 - 0.0625 \leq 0 \tag{41}$$

where $0 \leq x_i \leq 10$ $(i = 1, 2, 3)$ and $p, q, r = 1, 2, \ldots, 9$. The feasible region of the search space consists of 9^3 disjointed spheres. A point (x_1, x_2, x_3) is feasible if and only if there exist p, q, r such the above inequality (41) holds. The global optimum is located at $x^* = (5, 5, 5)$ where $f(x^*) = 1$. The solution lies within the feasible region.

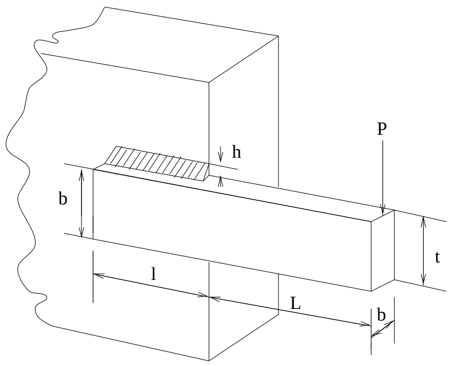

Figure 2: The welded beam used for problem 14 .

13. **Problem 13: (g13)**

    Minimize:

$$f(\vec{x}) = e^{x_1 x_2 x_3 x_4 x_5} \tag{42}$$

    subject to:

$$\begin{aligned} h_1(\vec{x}) &= x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0 \\ h_2(\vec{x}) &= x_2 x_3 - 5 x_4 x_5 = 0 \\ h_3(\vec{x}) &= x_1^3 + x_2^3 + 1 = 0 \end{aligned} \tag{43}$$

    where $-2.3 \leq x_i \leq 2.3$ $(i = 1, 2)$ and $-3.2 \leq x_i \leq 3.2$ $(i = 3, 4, 5)$. The optimum solution is $x^* = (-1.717143, 1.595709, 1.827247, -0.7636413, -0.763645)$ where $f(x^*) = 0.0539498$.

14. **Problem 14: (Design of a Welded Beam)**

    A welded beam is designed for minimum cost subject to constraints on shear stress (τ), bending stress in the beam (σ), buckling load on the bar (P_c), end deflection of the beam (δ), and side constraints [35]. There are four design variables as shown in Figure 2 [35]: h (x_1), l (x_2), t (x_3) and b (x_4).

    The problem can be stated as follows:

    Minimize:

$$f(\vec{x}) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14.0 + x_2) \tag{44}$$

Subject to:

$$
\begin{aligned}
g_1(\vec{x}) &= \tau(\vec{x}) - \tau_{max} \leq 0 \\
g_2(\vec{x}) &= \sigma(\vec{x}) - \sigma_{max} \leq 0 \\
g_3(\vec{x}) &= x_1 - x_4 \leq 0 \\
g_4(\vec{x}) &= 0.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \leq 0 \\
g_5(\vec{x}) &= 0.125 - x_1 \leq 0 \\
g_6(\vec{x}) &= \delta(\vec{x}) - \delta_{max} \leq 0 \\
g_7(\vec{x}) &= P - P_c(\vec{x}) \leq 0
\end{aligned}
\tag{45}
$$

where

$$\tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}$$

$$\tau' = \frac{P}{\sqrt{2}x_1x_2}, \tau'' = \frac{MR}{J}, M = P\left(L + \frac{x_2}{2}\right)$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}$$

$$J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}$$

$$\sigma(\vec{x}) = \frac{6PL}{x_4x_3^2}, \delta(\mathbf{X}) = \frac{4PL^3}{Ex_3^3x_4}$$

$$P_c(\vec{x}) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right) \tag{46}$$

$$P = 6000\ lb, \quad L = 14\ in, \quad E = 30 \times 10^6\ psi, \quad G = 12 \times 10^6\ psi$$

$$\tau_{max} = 13,600\ psi, \quad \sigma_{max} = 30,000\ psi, \quad \delta_{max} = 0.25\ in$$

where $0.1 \leq x_1 \leq 2.0$, $0.1 \leq x_2 \leq 10.0$, $0.1 \leq x_3 \leq 10.0$ y $0.1 \leq x_4 \leq 2.0$.

15. **Problem 15: (Design of a Pressure Vessel)**

A cylindrical vessel is capped at both ends by hemispherical heads as shown in Figure 3. The objective is to minimize the total cost, including the cost of the material, forming and welding. There are four design variables: T_s (thickness of the shell), T_h (thickness of the head), R (inner radius) and L (length of the cylindrical section of the vessel, not including the head). T_s and T_h are integer multiples of 0.0625 inch, which are the available thicknesses of rolled steel plates, and R and L are continuous. Using the same notation given by Kannan and Kramer [26], the problem can be stated as follows:

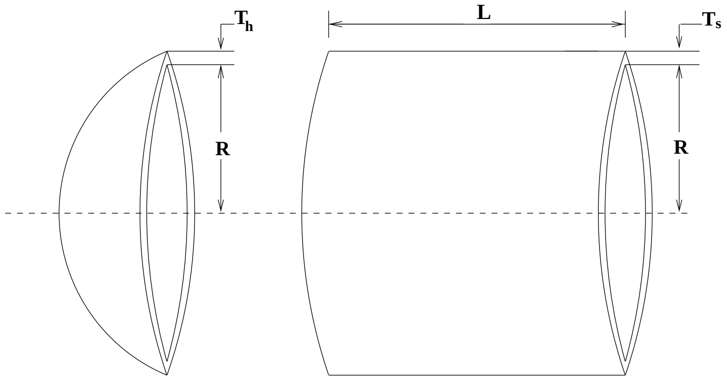

Figure 3: Center and end section of the pressure vessel used for problem 15.

Minimize :

$$f(\vec{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \quad (47)$$

Subject to :

$$\begin{aligned} g_1(\vec{x}) &= -x_1 + 0.0193x_3 \le 0 \\ g_2(\vec{x}) &= -x_2 + 0.00954x_3 \le 0 \\ g_3(\vec{x}) &= -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1,296,000 \le 0 \\ g_4(\vec{x}) &= x_4 - 240 \le 0 \end{aligned} \quad (48)$$

where $1 \le x_1 \le 99$, $1 \le x_2 \le 99$, $10 \le x_3 \le 200$ y $10 \le x_4 \le 200$.

16. **Problem 16: (Minimization of the Weight of a Tension/Compression String)**

This problem was described by Arora [2] and Belegundu [3], and it consists of minimizing the weight of a tension/compression spring (see Figure 4) subject to constraints on minimum deflection, shear stress, surge frequency, limits on outside diameter and on design variables. The design variables are the mean coil diameter D (x_2), the wire diameter d (x_1) and the number of active coils N (x_3).

Formally, the problem can be expressed as:

Minimize:

$$(N + 2)Dd^2 \quad (49)$$

Subject to:

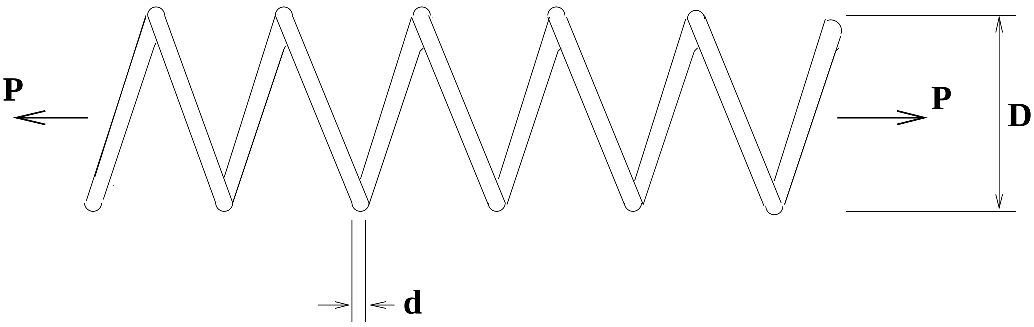

Figure 4: Tension/compression string used for problem 16.

$$
\begin{aligned}
g_1(\vec{x}) &= 1 - \frac{D^3 N}{71785 d^4} \leq 0 \\
g_2(\vec{x}) &= \frac{4D^2 - dD}{12566(Dd^3 - d^4)} + \frac{1}{5108 d^2} - 1 \leq 0 \\
g_3(\vec{x}) &= 1 - \frac{140.45 d}{D^2 N} \leq 0 \\
g_4(\vec{x}) &= \frac{D + d}{1.5} - 1 \leq 0 \qquad (50)
\end{aligned}
$$

where $0.05 \leq x_1 \leq 2$, $0.25 \leq x_2 \leq 1.3$ y $2 \leq x_3 \leq 15$.

17. **Problem 17: (Design of a 10-bar plane truss)**

Consider the 10-bar plane truss shown in Figure 5 [3]. The problem is to find the moment of inertia of each member of this truss, such that we minimize its weight, subject to stress and displacement constraints. The weight of the truss is given by:

$$
f(x) = \sum_{j=1}^{10} \rho\, A_j\, L_j \qquad (51)
$$

where x is the candidate solution, A_j is the cross-sectional area of the jth member, L_j is the length of the jth member, and ρ is the weight density of the material.

The assumed data are: modulus of elasticity, $E = 1.0 \times 10^4$ ksi 68965.5 MPa), $\rho = 0.10$ lb/in^3 (2768.096 kg/m^3), and a load of 100 kips (45351.47 Kg) in the negative y-direction is applied at nodes 2 and 4. The maximum allowable stress of each member is called σ_a, and it is assumed to be ± 25 ksi (172.41 MPa). The maximum allowable displacement of each node

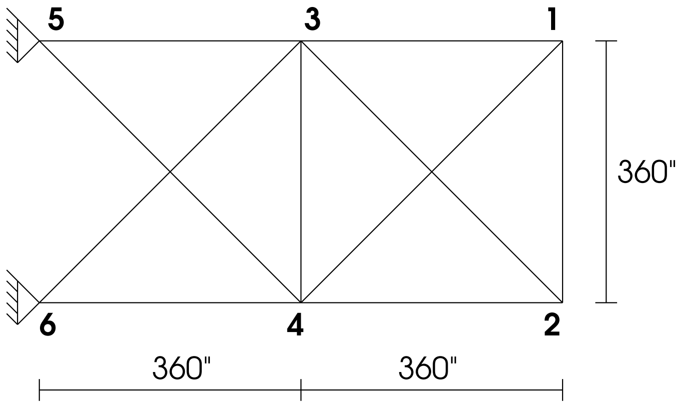

Figure 5: 10-bar plane truss used for problem 17.

(horizontal and vertical) is represented by u_a, and is assumed to be 2 inches (5.08 cm).

There are 10 stress constraints, and 12 displacement constraints (we can really assume only 8 displacement constraints because there are two nodes with zero displacement, but they will nevertheless be considered as additional constraints by the new approach). The moment of inertia of each element can be different, thus the problem has 10 design variables.

To get a measure of the difficulty of solving each of these problems, a ρ metric (as suggested by Koziel and Michalewicz [27]) was computed using the following expression:

$$\rho = |F|/|S| \tag{52}$$

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 5 and 13 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 [9] 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:

Problem	n	Type of function	ρ	LI	NI	LE	NE
1	13	quadratic	0.0003%	9	0	0	0
2	20	non linear	99.9973%	2	0	0	0
3	10	non linear	0.0026%	0	0	0	1
4	5	quadratic	27.0079%	4	2	0	0
5	4	non linear	0.0000%	2	0	0	3
6	2	non linear	0.0057%	0	2	0	0
7	10	quadratic	0.0000%	3	5	0	0
8	2	non linear	0.8581%	0	2	0	0
9	7	non linear	0.5199%	0	4	0	0
10	8	linear	0.0020%	6	0	0	0
11	2	quadratic	0.0973%	0	0	0	1
12	3	quadratic	4.7697%	0	9^3	0	0
13	5	non linear	0.0000%	0	0	1	2
14	4	quadratic	2.6859%	6	1	0	0
15	4	quadratic	39.6762%	3	1	0	0
16	3	quadratic	0.7537%	1	3	0	0
17	10	non linear	46.8070%	0	22	0	0

Table 1: Values of ρ for the four test problems chosen.

- **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 3, 4, 5 and 6, where P_i refers to the problem solved ($1 \leq i \leq 17$), the symbol "*" and the number between parenthesis "(n)"mean that only in n runs feasible solutions were found , $0.0 \leq F_p \leq 1.0$ is the average rate of feasible solutions found during a single run (with respect to the full population). A summary of all results (for a better visualization to compare them) is shown in Table 2

5 Discussion of Results

After the experimental phase, several results were obtained and they are discussed next:

5.1 Quality of the results

HCNPGA gave the best results in 9 problems and in six of them it reached the global optimum. HCMOGA is superior in 7 of them and it reached the global optimum in 4. HCVEGA got better results only in 4 problems and it found the global optimum in 3. COMOGA was clearly surpassed because it did not give the best results in any given problem.

5.2 Consistency

The statistics indicate that HCNPGA and HCVEGA presented the lowest standard deviation and the best median and average solutions in the same number of problems (7). The second best approach was HCMOGA which produced the best results in 4 problems. COMOGA was the best in only 2 functions. Despite the fact that there was a tie between HCNPGA and HCVEGA, the last one showed premature convergence in most ot the test problems.

		GLOBAL						
Ap.	P	Optimal	Best	Median	Mean	St. Dev.	Worst	F_p
1	P_1	−15.000000	−5.000041	0.000000	−0.673341	1.353731	0.000000	0.001290
2	P_1	−15.000000	−11.221095	−9.702318	−9.715928	0.493894	−8.387313	0.146110
3	$P_1 * (52)$	−15.000000	−10.565151	−6.460820	−6.496296	2.184355	−1.275637	0.000012
4	P_1	−15.000000	−13.967349	−12.900057	−12.752793	0.815685	−9.328688	0.013744
1	P_2	0.803619	0.027258	0.021285	0.020641	0.003602	0.008432	0.998641
2	P_2	0.803619	−0.000037	−0.000270	−0.000315	0.000178	−0.000983	0.999955
3	P_2	0.803619	0.785141	0.731922	0.724332	0.034197	0.528319	0.914850
4	P_2	0.803619	−0.425258	−0.512270	−0.515214	0.040292	−0.618569	0.998833
1	P_3	1.000000	0.125546	0.000509	−1399999.995091	3469870.316560	−	0.000223
2	P_3	1.000000	−0.000000	−0.000000	0.000000	0.000000	−0.000000	0.002225
3	P_3	1.000000	0.974879	0.866315	0.861310	0.053738	0.715349	0.002336
4	P_3	1.000000	−0.035369	−0.219031	−0.246764	0.143303	−0.601319	0.009780
1	P_4	−30665.539	−30533.056641	−30328.199219	−30329.563398	74.793290	−30141.033203	0.002358
2	P_4	−30665.539	−30647.246094	−30628.587891	−30628.468711	7.877054	−30607.240234	0.408108
3	P_4	−30665.539	−30661.033203	−30635.346680	−30630.883145	20.466057	−30544.324219	0.345432
4	P_4	−30665.539	−30649.958984	−30570.754883	−30568.917930	53.531272	−30414.773438	0.344896
1	P_5	5126.4981	−	−	−	−	−	−
2	P_5	5126.4981	−	−	−	−	−	−
3	P_5	5126.4981	−	−	−	−	−	−
4	P_5	5126.4981	8879.123047	8879.977051	8879.944961	0.100387	8879.999023	0.000000
1	P_6	−6961.81388	−6808.695801	−5408.742920	−5255.104592	994.761331	−1341.317627	0.002026
2	P_6	−6961.81388	−6942.747070	−6758.276855	−6762.048252	101.608629	−6516.471191	0.042749
3	P_6	−6961.81388	−6941.307129	−6748.525146	−6644.539102	336.222702	−5164.000000	0.017528
4	P_6	−6961.81388	−6939.439941	−6699.261963	−6678.926099	156.695161	−6258.591309	0.017084
1	$P_7 * (8)$	24.3062091	485.579163	1987.980957	1567.294106	924.459907	3149.045898	0.000330
2	P_7	24.3062091	28.491827	34.527794	34.558270	2.931860	41.785904	0.153518
3	P_7	24.3062091	26.985840	30.882214	31.248952	2.324165	38.459484	0.049250
4	P_7	24.3062091	29.572760	40.376375	45.589268	15.171002	114.547325	0.013262
1	$P_8 * (99)$	0.095825	0.095782	0.094613	0.094343	0.001336	0.089900	0.002883
2	P_8	0.095825	0.095826	0.095826	0.095826	0.000000	0.095826	0.393393
3	P_8	0.095825	0.095826	0.095826	0.095826	0.000000	0.095826	0.424794
4	P_8	0.095825	0.095825	0.095825	0.095825	0.000000	0.095825	0.074362
1	P_9	680.6300573	733.001221	979.748016	983.625328	115.570129	1314.535767	0.011225
2	P_9	680.6300573	693.642029	736.190613	739.306726	25.164813	806.855469	0.046137
3	P_9	680.6300573	680.951172	682.121002	682.335237	0.836150	684.816284	0.240264
4	P_9	680.6300573	681.707947	689.955780	692.966086	10.957636	734.002258	0.048562
1	$P_{10} * (71)$	7049.3307	10865.433594	18994.846680	18924.576419	3851.982032	26625.984375	0.000000
2	$P_{10} * (63)$	7049.3307	9842.453125	17708.879883	17605.587860	3877.896934	27627.634766	0.000050
3	$P_{10} * (29)$	7049.3307	8183.303711	12691.962402	13716.703697	4804.769185	23585.121094	0.000460
4	P_{10}	7049.3307	7578.336426	9026.873047	9504.358618	1506.820708	16473.984375	0.011467
1	P_{11}	0.750	0.749058	0.749787	0.749829	0.000495	0.751753	0.000286
2	P_{11}	0.750	0.749621	0.812369	0.798690	0.025821	0.847242	0.011206
3	P_{11}	0.750	0.749001	0.749613	0.753909	0.012147	0.832940	0.025842
4	P_{11}	0.750	0.749001	0.749155	0.749393	0.000595	0.752445	0.016913
1	P_{12}	1.000000	0.999999	0.999835	0.999679	0.000930	0.993070	0.070262
2	P_{12}	1.000000	1.000000	1.000000	1.000000	0.000000	1.000000	0.516501
3	P_{12}	1.000000	1.000000	1.000000	1.000000	0.000000	1.000000	0.446223
4	P_{12}	1.000000	1.000000	1.000000	1.000000	0.000000	1.000000	0.090064
1	P_{13}	0.0539498	−	−	−	−	−	−
2	P_{13}	0.0539498	−	−	−	−	−	−
3	P_{13}	0.0539498	−	−	−	−	−	−
4	P_{13}	0.0539498	−	−	−	−	−	−
1	P_{14}	1.728226	1.835381	2.039148	2.030360	0.094810	2.328703	0.000580
2	P_{14}	1.728226	1.726772	1.735865	1.736439	0.005927	1.769726	0.346061
3	P_{14}	1.728226	1.727014	1.751112	1.766413	0.043733	1.978103	0.261526
4	P_{14}	1.728226	1.729384	1.791587	1.825219	0.079812	2.126669	0.064106
1	P_{15}	6059.946341	6369.428223	7889.838867	7795.411538	701.363966	9147.520508	0.003989
2	P_{15}	6059.946341	6064.723633	6238.489746	6259.963745	170.254024	6820.944824	0.425354
3	P_{15}	6059.946341	6059.926270	6127.618408	6172.527373	123.897547	6845.770508	0.330693
4	P_{15}	6059.946341	6066.969727	6561.483154	6629.064048	385.110736	7547.403320	0.452672
1	P_{16}	0.012681	0.012929	0.014263	0.014362	0.000864	0.017113	0.021144
2	P_{16}	0.012681	0.012688	0.012789	0.012886	0.000209	0.013784	0.250408
3	P_{16}	0.012681	0.012683	0.012736	0.012752	0.000062	0.013132	0.104895
4	P_{16}	0.012681	0.012680	0.012815	0.012960	0.000363	0.014754	0.047492
1	P_{17}	5152.636136	6283.198730	6675.126709	6660.455649	126.289250	6968.627441	0.006516
2	P_{17}	5152.636136	5327.418457	5453.446045	5455.871895	56.744241	5569.240723	0.676408
3	P_{17}	5152.636136	5179.740723	5256.108154	5259.013174	37.658930	5362.890625	0.503566
4	P_{17}	5152.636136	5336.618652	5745.238281	5748.839526	210.696096	6474.041992	0.603598

Table 2: Experimental results using the four approaches with the 17 test problems. The "-" symbol means that no feasible solutions were found during the experiments.

	COMOGA						
P	**Optimal**	**Best**	**Median**	**Mean**	**St. Dev.**	**Worst**	F_p
P_1	-15.000000	-5.000041	0.000000	-0.673341	1.353731	0.000000	0.001290
P_2	0.803619	0.027258	0.021285	0.020641	0.003602	0.008432	0.998641
P_3	1.000000	0.125546	0.000509	-1399999.995091	3469870.316560	–	0.000223
P_4	-30665.539	-30533.056641	-30328.199219	-30329.563398	74.793290	-30141.033203	0.002358
P_5	5126.4981	–	–	–	–	–	–
P_6	-6961.81388	-6808.695801	-5408.742920	-5255.104592	994.761331	-1341.317627	0.002026
$P_7 * (8)$	24.3062091	485.579163	1987.980957	1567.294106	924.459907	3149.045898	0.000330
$P_8 * (99)$	0.095825	0.095782	0.094613	0.094343	0.001336	0.089900	0.002883
P_9	680.6300573	733.001221	979.748016	983.625328	115.570129	1314.535767	0.011225
$P_{10} * (71)$	7049.3307	10865.433594	18994.846680	18924.576419	3851.982032	26625.984375	0.000000
P_{11}	0.750	0.749058	0.749787	0.749829	0.000495	0.751753	0.000286
P_{12}	1.000000	-0.999999	-0.999835	-0.999679	0.000930	-0.993070	0.070262
P_{13}	0.0539498	–	–	–	–	–	–
P_{14}	1.728226	1.835381	2.039148	2.030360	0.094810	2.328703	0.000580
P_{15}	6059.946341	6369.428223	7889.838867	7795.411538	701.363966	9147.520508	0.003989
P_{16}	0.012681	0.012929	0.014263	0.014362	0.000864	0.017113	0.021144
P_{17}	5152.636136	6283.198730	6675.126709	6660.455649	126.289250	6968.627441	0.006516

Table 3: Experimental results using COMOGA with the 17 test problems. The "-" symbol means that no feasible solutions were found during the experiments.

	HCVEGA						
P	**Optimal**	**Best**	**Median**	**Mean**	**St. Dev.**	**Worst**	F_p
P_1	-15.000000	-11.221095	-9.702318	-9.715928	0.493894	-8.387313	0.146110
P_2	0.803619	-0.000037	-0.000270	-0.000315	0.000178	-0.000983	0.999955
P_3	1.000000	-0.000000	-0.000000	0.000000	0.000000	-0.000000	0.002225
P_4	-30665.539	-30647.246094	-30628.587891	-30628.468711	7.877054	-30607.240234	0.408108
P_5	5126.4981	–	–	–	–	–	–
P_6	-6961.81388	-6942.747070	-6758.276855	-6762.048252	101.608629	-6516.471191	0.042749
P_7	24.3062091	28.491827	34.527794	34.558270	2.931860	41.785904	0.153518
P_8	0.095825	0.095826	0.095826	0.095826	0.000000	0.095826	0.393393
P_9	680.6300573	693.642029	736.190613	739.306726	25.164813	806.855469	0.046137
$P_{10} * (63)$	7049.3307	9842.453125	17708.879883	17605.587860	3877.896934	27627.634766	0.000050
P_{11}	0.750	0.749621	0.812369	0.798690	0.025821	0.847242	0.011206
P_{12}	1.000000	1.000000	1.000000	1.000000	0.000000	1.000000	0.516501
P_{13}	0.0539498	–	–	–	–	–	–
P_{14}	1.728226	1.726772	1.735865	1.736439	0.005927	1.769726	0.346061
P_{15}	6059.946341	6064.723633	6238.489746	6259.963745	170.254024	6820.944824	0.425354
P_{16}	0.012681	0.012688	0.012789	0.012886	0.000209	0.013784	0.250408
P_{17}	5152.636136	5327.418457	5453.446045	5455.871895	56.744241	5569.240723	0.676408

Table 4: Experimental results using HCVEGA to handle constraints with the 17 test problems. The "-" symbol means that no feasible solutions were found during the experiments.

	HCNPGA						
P	**Optimal**	**Best**	**Median**	**Mean**	**St. Dev.**	**Worst**	F_p
$P_1 * (52)$	-15.000000	-10.565151	-6.460820	-6.496296	2.184355	-1.275637	0.000012
P_2	0.803619	0.785141	0.731922	0.724332	0.034197	0.528319	0.914850
P_3	1.000000	0.974879	0.866315	0.861310	0.053738	0.715349	0.002336
P_4	-30665.539	-30661.033203	-30635.346680	-30630.883145	20.466057	-30544.324219	0.345432
P_5	5126.4981	–	–	–	–	–	–
P_6	-6961.81388	-6941.307129	-6748.525146	-6644.539102	336.222702	-5164.000000	0.017528
P_7	24.3062091	26.985840	30.882214	31.248952	2.324165	38.459484	0.049250
P_8	0.095825	0.095826	0.095826	0.095826	0.000000	0.095826	0.424794
P_9	680.6300573	680.951172	682.121002	682.335237	0.836150	684.816284	0.240264
$P_{10} * (29)$	7049.3307	8183.303711	12691.962402	13716.703697	4804.769185	23585.121094	0.000460
P_{11}	0.750	0.749001	0.749613	0.753909	0.012147	0.832940	0.025842
P_{12}	1.000000	1.000000	1.000000	1.000000	0.000000	1.000000	0.446223
P_{13}	0.0539498	–	–	–	–	–	–
P_{14}	1.728226	1.727014	1.751112	1.766413	0.043733	1.978103	0.261526
P_{15}	6059.946341	6059.926270	6127.618408	6172.527373	123.897547	6845.770508	0.330693
P_{16}	0.012681	0.012683	0.012736	0.012752	0.000062	0.013132	0.104895
P_{17}	5152.636136	5179.740723	5256.108154	5259.013174	37.658930	5362.890625	0.503566

Table 5: Experimental results using HCNPGA to handle constraints with the 17 test problems. The "-" symbol means that no feasible solutions were found during the experiments.

	HCMOGA						
Problem	Optimal	Best	Median	Mean	St. Dev.	Worst	F_p
P_1	−15.000000	−13.967349	−12.900057	−12.752793	0.815685	−9.328688	0.013744
P_2	0.803619	−0.425258	−0.512270	−0.515214	0.040292	−0.618569	0.998833
P_3	1.000000	−0.035369	−0.219031	−0.246764	0.143303	−0.601319	0.009780
P_4	−30665.539	−30649.958984	−30570.754883	−30568.917930	53.531272	−30414.773438	0.344896
P_5	5126.4981	8879.123047	8879.977051	8879.944961	0.100387	8879.999023	0.000000
P_6	−6961.81388	−6939.439941	−6699.261963	−6678.926099	156.695161	−6258.591309	0.017084
P_7	24.3062091	29.572760	40.376375	45.589268	15.171002	114.547325	0.013262
P_8	0.095825	0.095825	0.095825	0.095825	0.000000	0.095825	0.074362
P_9	680.6300573	681.707947	689.955780	692.966086	10.957636	734.002258	0.048562
P_{10}	7049.3307	7578.336426	9026.873047	9504.358618	1506.820708	16473.984375	0.011467
P_{11}	0.750	0.749001	0.749155	0.749393	0.000595	0.752445	0.016913
P_{12}	1.000000	−1.000000	−1.000000	−1.000000	0.000000	−1.000000	0.090064
P_{13}	0.0539498	–	–	–	–	–	–
P_{14}	1.728226	1.729384	1.791587	1.825219	0.079812	2.126669	0.064106
P_{15}	6059.946341	6066.969727	6561.483154	6629.064048	385.110736	7547.403320	0.452672
P_{16}	0.012681	0.012680	0.012815	0.012960	0.000363	0.014754	0.047492
P_{17}	5152.636136	5336.618652	5745.238281	5748.839526	210.696096	6474.041992	0.603598

Table 6: Experimental results using HCMOGA to handle constraints with the 17 test problems. The "-" symbol means that no feasible solutions were found during the experiments.

Category	Problem
Very difficult	P_5, P_{13}
Difficult	P_1, P_2, P_3, P_{10}
Average	P_4, P_6, P_7, P_9, P_{15}, P_{17}
Easy	P_8, P_{11}, P_{12}, P_{14}, P_{16}

Table 7: Clasification of test functions based on the experimental phase.

5.3 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 most balanced approach based on the number of feasible and infeasible solutions in the population during all the evolutionary process was HCVEGA. HCNPGA ranked second in this category. HCMOGA had results very close to those of HCNPGA. This suggests that a population-based approach should help to maintain diversity. Pareto ranking as a selection criteria is helpful too, based on the results obtained by HCNPGA. However, Pareto Ranking seems to give a less balanced population, probably because of its higher selection pressure.

5.4 Difficult of test problems

Based on the results obtained, all seventeen functions are classified in three categories: Very difficult, difficult, average and easy. The classification is presented in Table 7.

5.5 Final Remarks

Some issues can be stated based on the previous study:

- The four Multiobjective-Based techniques to handle constraints compared in this study do not work too well with large feasible regions. It means that these approaches have problems to move towards the global optimum once the feasible region is reached (e.g.P_2 where the feasible region covers 99% of the search space).
- When the problem to be solved has more than one nonlinear equality (P_5 and P_{13}) it turns out to be very difficult to solve by any of the approaches compared in this study.
- The performance of all the techniques compared degrades in the presence of high dimensionality (P_1 with 13 decision variables, P_2 with 20 decision variables, P_3 with 10 decision variables and P_{10} with 8 decision variables).
- The four compared techniques are efficient in problems with small feasible regions (joint, disjoint, non convex) formed by a significant number of inequality constraints (both linear and nonlinear).
- With low dimensionality (2 or 3 decision variables) and constraints on any kind, The four compared approaches find very good results (P_8, P_{11}, and P_{12}).
- HCVEGA prematurely converges to local optima.
- HCMOGA was the only one to find feasible solutions in one very difficult and one difficult function (P_5 and P_{13}). However the solutions found are far from the global optimum.
- The results suggest that spliting the population into subpopulations aims to a better diversity balance.
- Pareto dominance as a selection criterion has proved to give better results (in terms of optimality) than Pareto Ranking or a population-based approach. However it can not reach the global optima in problems with either high dimensionality, large feasible regions or several nonlinear equality constraints.
- The overall results of the COMOGA's steady-state GA suggests that it is not a good option to solve nonlinear global optimization 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. However, The use of Pareto Ranking in HCMOGA gave better results in difficult problems. Furthermore, HCVEGA could maintain diversity better than the 3 other approaches. 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 seventeen 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 [38, 41] and with the most competitive constraint-handling techniques used with EAs (e.g., stochastic ranking [39], the homomourphous maps [27], and the adaptive segregational constrained handling evolutionary algorithm (ASCHEA) [21]) 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

This work is representative of the research performed by the Evolutionary Computation Group at CINVESTAV-IPN (EVOCINV). 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 acknowledges valuable help from Dr. Patrick D. Surry.

References

[1] Dirk V. Arnold. *Noisy Optimization with Evolution Strategies.* Kluwer Academic Publishers, New York, June 2002. ISBN 1-4020-7105-1.

[2] Jasbir S. Arora. *Introduction to Optimum Design.* McGraw-Hill, New York, 1989.

[3] Ashok Dhondu Belegundu. *A Study of Mathematical Programming Methods for Structural Optimization*. Department of civil and environmental engineering, University of Iowa, Iowa, Iowa, 1982.

[4] Jürgen Branke. *Evolutionary Optimization in Dynamic Environments*. Kluwer Academic Publishers, New York, December 2001. ISBN 0-7923-7631-5.

[5] Martin V. Butz. *Anticipatory Learning Classifier Systems*. Kluwer Academic Publishers, New York, December 2001. ISBN 0-7923-7630-7.

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

[7] Erick Cantú-Paz. *Efficient and Accurate Parallel Genetic Algorithms*. Kluwer Academic Publishers, New York, November 2000. ISBN 0-7923-7221-2.

[8] V. Chankong and Y.Y Haimes. Multiobjective Decision Making: Theory and Methodology. In Andrew P. Sage, editor, *Systems Science and Engineering*. North-Holland, 1983.

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

[10] Carlos A. Coello Coello and Efrén Mezura-Montes. Handling Constraints in Genetic Algorithms Using Dominance-Based Tournaments. In I.C. 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.

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

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

[13] Carlos A. Coello Coello and Arturo Hernández Aguirre. Design of Combinational Logic Circuits through an Evolutionary Multiobjective Optimization Approach. *Artificial Intelligence for Engineering, Design, Analysis and Manufacture*, 16(1):39–53, 2002.

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

[15] Kalyanmoy Deb and R. W. Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9:115–148, 1995.

[16] Kalyanmoy Deb and David E. Goldberg. An Investigation of Niche and Species Formation in Genetic Function Optimization. In J. David 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.

[17] Francis Ysidro Edgeworth. *Mathematical Physics*. P. Keagan, London, England, 1881.

[18] Carlos M. Fonseca and Peter J. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In Stephanie 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.

[19] David Goldberg. *The Design of Innovation*. Kluwer Academic Publishers, New York, June 2002. ISBN 1-4020-7098-5.

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

[21] S. Ben Hamida and Marc Schoenauer. An Adaptive Algorithm for Constrained Optimization Problems. In Marc Schoenauer, Kalyanmoy Deb, Günter Rudolph, Xin Yao, Evelyne Lutton, Juan Julian Merelo, and Hans-Paul Schwefel, editors, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 529–538, Paris, France, 2000. Springer. Lecture Notes in Computer Science No. 1917.

[22] Robert Hinterding and Zbigniew Michalewicz. Your Brains and My Beauty: Parent Matching for Constrained Optimisation. In *Proceedings of the 5th International Conference on Evolutionary Computation*, pages 810–815, Anchorage, Alaska, May 1998.

[23] Jeffrey Horn, Nicholas Nafpliotis, and David 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.

[24] F. Jiménez, A.F. 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.A. Gómez, F. Herrera, J.I. Hidalgo, J. Lanchares, J.J. Merelo, and J.M. Sánchez, editors, *Primer Congreso Espaõl de Algoritmos Evolutivos y Bioinspirados (AEB'02)*, pages 460–465, Mérida España, 2002. Universidad de la Extremadura, España.

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

[26] B. K. Kannan and S. N. Kramer. An Augmented Lagrange Multiplier Based Method for Mixed Integer Discrete Continuous Optimization and Its Applications to Mechanical Design. *Journal of Mechanical Design. Transactions of the ASME*, 116:318–320, 1994.

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

[28] Nivedita Sumi Majumdar and Dipankar Dasgupta. Determining Optimal Configuration for Turbine Generator Cooler. In *Proceedings of the Congress on Evolutionary Computation 2001 (CEC'2001)*, volume 2, pages 1009–1014, Piscataway, New Jersey, May 2001. IEEE Service Center.

[29] Efrén 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).

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

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

[32] Shigeru Obayashi. Pareto Solutions of Multipoint Design of Supersonic Wings using Evolutionary Algorithms. In I.C. Parmee, editor, *Adaptive Computing in Design and Manufacture V*, pages 3–15, London, 2002. Springer-Verlag.

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

[34] 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.

[35] Singiresu S. Rao. *Engineering Optimization*. John Wiley and Sons, third edition, 1996.

[36] Tapabrata 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.

[37] Tapabrata Ray, Tai Kang, and Seow Kian Chye. An Evolutionary Algorithm for Constrained Optimization. In Darrell Whitley, David Goldberg, Erick Cantú-Paz, Lee Spector, Ian Parmee, and Hans-Georg Beyer, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2000)*, pages 771–777, San Francisco, California, July 2000. Morgan Kaufmann.

[38] Jon T. Richardson, Mark R. Palmer, Gunar Liepins, and Mike Hilliard. Some Guidelines for Genetic Algorithms with Penalty Functions. In J. David 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.

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

[40] J. David 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.

[41] Alice E. Smith and David W. Coit. Constraint Handling Techniques—Penalty Functions. In Thomas Bäck, David B. Fogel, and Zbigniew Michalewicz, editors, *Handbook of Evolutionary Computation*, chapter C 5.2. Oxford University Press and Institute of Physics Publishing, 1997.

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

[43] Patrick D. Surry, Nicholas J. Radcliffe, and Ian D. Boyd. A Multi-Objective Approach to Constrained Optimisation of Gas Supply Networks : The COMOGA Method. In Terence C. Fogarty, editor, *Evolutionary Computing. AISB Workshop. Selected Papers*, Lecture Notes in Computer Science, pages 166–180. Springer-Verlag, Sheffield, U.K., 1995.

[44] K.C. Tan, T.H. Lee, J. Cai, and Y. H. Chew. Automating the Drug Scheduling of Cancer Chemotherapy via Evolutionary Computation. In *Proceedings of the Congress on Evolutionary Computation 2001 (CEC'2001)*, volume 2, pages 908–913, Piscataway, New Jersey, May 2001. IEEE Service Center.

[45] K. Tesh, M.A. Atherton, and M.W. Collins. Genetic Algorithms Search for Stent Design Improvements. In I.C. Parmee, editor, *Adaptive Computing in Design and Manufacture V*, pages 99–107, London, 2002. Springer-Verlag.