Use of Multiobjective Optimization Concepts to Handle Constraints in Single-Objective Optimization

Arturo Hernández Aguirre[1], Salvador Botello Rionda[1],
Carlos A. Coello Coello[2] and Giovanni Lizárraga Lizárraga[1]

[1] Center for Research in Mathematics (CIMAT)
Department of Computer Science
Guanajuato, Gto. 36240, México
artha,botello,giovanni@cimat.mx
[2] CINVESTAV-IPN
Evolutionary Computation Group
Depto. de Ingeniería Eléctrica
Sección de Computación
Av. Instituto Politécnico Nacional No. 2508
Col. San Pedro Zacatenco
México, D. F. 07300
ccoello@cs.cinvestav.mx

Abstract. In this paper, we propose a new constraint-handling technique for evolutionary algorithms which is based on multiobjective optimization concepts. The approach uses Pareto dominance as its selection criterion, and it incorporates a secondary population. The new technique is compared with respect to an approach representative of the state-of-the-art in the area using a well-known benchmark for evolutionary constrained optimization. Results indicate that the proposed approach is able to match and even outperform the technique with respect to which it was compared at a lower computational cost.

1 Introduction

The success of Evolutionary Algorithms (EAs) in global optimization has triggered a considerable amount of research regarding the development of mechanisms able to incorporate information about the constraints of a problem into the fitness function of the EA used to optimize it [7]. So far, the most common approach adopted in the evolutionary optimization literature to deal with constrained search spaces is the use of penalty functions [10]. 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 indicates the degree of penalization to be applied [12].

Recently, some researchers have suggested the use of multiobjective optimization concepts to handle constraints in EAs. This paper introduces a new approach that is based on an evolution strategy that was originally proposed for multiobjective optimization: the Pareto Archived Evolution Strategy (PAES) [5]. Our approach (which is an extension of PAES) can be used to handle constraints in single-objective optimization problems and does not present the scalability problems of the original PAES.

Besides using Pareto-based selection, our approach uses a secondary population (one of the most common notions of elitism in evolutionary multiobjective optimization), and a mechanism that reduces the constrained search space so that our technique can approach a optimum more efficiently.

2 Problem Statement

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

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

subject to:

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

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

where $\boldsymbol{x}$ is the vector of solutions $\boldsymbol{x} = [x_1, x_2, \ldots, x_r]^T$, 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). For an inequality constraint that satisfies $g_i(\boldsymbol{x}) = 0$, then we will say that is active at $\boldsymbol{x}$. All equality constraints h_j (regardless of the value of $\boldsymbol{x}$ used) are considered active at all points of $\mathcal{F}$ ($\mathcal{F}$ = feasible region).

3 Basic Concepts

A multiobjective optimization problem (MOP) has the following the form:

$$\text{Minimize } [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x})] \tag{4}$$

subject to the m inequality constraints:

$$g_i(\boldsymbol{x}) \geq 0 \quad i = 1, 2, \ldots, m \tag{5}$$

and the p equality constraints:

$$h_i(\boldsymbol{x}) = 0 \quad i = 1, 2, \ldots, p \tag{6}$$

where k is the number of objective functions $f_i : \mathbb{R}^n \to \mathbb{R}$. We call $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T$ the vector of decision variables. We wish to determine from among the set $\mathcal{F}$ of all vectors which satisfy (5) and (6) the particular set of values $x_1^*, x_2^*, \ldots, x_n^*$ which yield the optimum values of all the objective functions.

3.1 Pareto Optimality

A vector $\boldsymbol{u} = (u_1, \ldots, u_k)$ is said to dominate $\boldsymbol{v} = (v_1, \ldots, v_k)$ (denoted by $\boldsymbol{u} \preceq \boldsymbol{v}$) if and only if u is partially less than v, i.e., $\forall i \in \{1, \ldots, k\},\ u_i \leq v_i \wedge \exists i \in \{1, \ldots, k\} : u_i < v_i$. For a given multiobjective optimization problem, $\boldsymbol{f}(x)$, the Pareto optimal set ($\mathcal{P}^*$) is defined as:

$$\mathcal{P}^* := \{x \in \mathcal{F} \mid \neg\exists\, x' \in \mathcal{F} \;\; \boldsymbol{f}(x') \preceq \boldsymbol{f}(x)\}. \tag{7}$$

Thus, we say that a vector of decision variables $\boldsymbol{x}^* \in \mathcal{F}$ is *Pareto optimal* if there does not exist another $\boldsymbol{x} \in \mathcal{F}$ such that $f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{x}^*)$ for all $i = 1, \ldots, k$ and $f_j(\boldsymbol{x}) < f_j(\boldsymbol{x}^*)$ for at least one j. In words, this definition says that $\boldsymbol{x}^*$ is Pareto optimal if there exists no feasible vector of decision variables $\boldsymbol{x} \in \mathcal{F}$ which would decrease some criterion without causing a simultaneous increase in at least one other criterion. Unfortunately, this concept almost always gives not a single solution, but rather a set of solutions called the *Pareto optimal set*. The vectors $\boldsymbol{x}^*$ correspoding to the solutions included in the Pareto optimal set are called *nondominated*. The image of the Pareto optimal set under the objective functions is called *Pareto front*.

4 Related Work

The main idea of adopting multiobjective optimization concepts to handle constraints is to redefine the single-objective optimization of $f(\boldsymbol{x})$ as a multiobjective optimization problem in which we will have $m + 1$ objectives, where m is the total number of constraints. Then, we can apply any multiobjective optimization technique [3] to the new vector $\bar{v} = (f(\boldsymbol{x}), f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))$, where $f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})$ are the original constraints of the problem. An ideal solution $\boldsymbol{x}$ would thus have $f_i(\boldsymbol{x})$=0 for $1 \leq i \leq m$ and $f(\boldsymbol{x}) \leq f(\boldsymbol{y})$ for all feasible $\boldsymbol{y}$ (assuming minimization).

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

1. Use of Pareto dominance as a selection criterion.
2. Use of Pareto ranking [4] 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.

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 on 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. Our main concern is that feasibility takes precedence, in this case, over nondominance. Therefore, good "trade-off" solutions that are not feasible cannot be considered as good as bad "trade-off" solutions that are feasible. Furthermore, a mechanism to maintain diversity must normally be added to any evolutionary multiobjective optimization technique. In our proposal, diversity is kept by using an adaptable grid, and by a selection process applied to the external file that maintains a mixture of both good "trade-off" and feasible individuals.

There are several approaches that have been developed using multiobjective optimization concepts to handle constraints, but due to space limitations we will not discuss them here (see for example [2, 13, 8, 9]).

5 Description of IS-PAES

Our approach (called Inverted-Shrinkable Pareto Archived Evolution Strategy, or IS-PAES) has been implemented as an extension of the Pareto Archived Evolution Strategy (PAES) proposed by Knowles and Corne [5] for multiobjective optimization. PAES's main feature is the use of an adaptive grid on which objective function space is located using a coordinate system. Such a grid is the diversity maintenance mechanism of PAES and it constitutes the main feature of this algorithm. The grid is created by bisecting k times the function space of dimension $d = g+1$. The control of 2^{kd} grid cells means the allocation of a large amount of physical memory for even small problems. For instance, 10 functions and 5 bisections of the space produce 2^{50} cells. Thus, the first feature introduced in IS-PAES is the "inverted" part of the algorithm that deals with this space usage problem. IS-PAES's fitness function is mainly driven by a feasibility criterion. Global information carried by the individuals surrounding the feasible region is used to concentrate the search effort on smaller areas as the evolutionary process takes place. In consequence, the search space being explored is "shrunk" over time. Eventually, upon termination, the size of the search space being inspected will be very small and will contain the solution desired. The main algorithm of IS-PAES is shown in Figure 1.

```
maxsize: max size of file
c: current parent ∈ X (decision variable space)
h:child of c ∈ X, a_h: individual in file that dominates h
a_d: individual in file dominated by h
current: current number of individuals in file
cnew: number of individuals generated thus far
current = 1; cnew=0;
c = newindividual();
add(c);
While cnew≤MaxNew do
    h = mutate(c); cnew+ =1;
    if (c⪯h) then Label A
    else if (h⪯c) then { remove(c); add(g); c=h; }
    else if (∃ a_h ∈ file | a_h ⪯ h) then Label A
    else if (∃ a_d ∈ file | h ⪯ a_d) then {
        add( h ); ∀ a_d { remove(a_d); current− =1 }
    else test(h,c,file)
Label A
    if (cnew % g==0) then c = individual in less densely populated region
    if (cnew % r==0) then shrinkspace(file)
End While
```

Fig. 1. Main algorithm of IS-PAES

The function **test(h,c,file)** determines if an individual can be added to the external memory or not. Here we introduce the following notation: $x_1 \square x_2$ means x_1 is located in a less populated region of the grid than x_2. The pseudo-code of this function is depicted in Figure 2.

```
if (current < maxsize) then {
    add(h);
    if (h □ c) then c=h }
else if (∃a_p∈file | h □ a_p) then {
    remove(a_p); add(h)
    if (h □ c) then c = h; }
```

Fig. 2. Pseudo-code of **test(h,c,file)**

5.1 Inverted "ownership"

PAES keeps a list of individuals on every grid location, but in IS-PAES each individual knows its position on the grid. Therefore, building a sorted list of the most dense populated areas of the grid only requires to sort the k elements of the external memory. In PAES, this procedure needs to inspect all 2^{kd} locations in order to generate a list of the individuals sorted by the density of their current location in the grid. The advantage of the inverted relationship is clear when the optimization problem has many functions (more than 10), and/or the granularity of the grid is fine, for in this case only IS-PAES is able to deal with any number of functions and granularity level.

5.2 Shrinking the objective space

Shrinkspace(file) is the most important function of IS-PAES since its task is the reduction of the search space. The pseudo-code of **Shrinkspace(file)** is shown in Figure 3.

$\underline{\boldsymbol{x}}_{pob}$: vector containing the smallest value of either $x_i \in X$
$\overline{\boldsymbol{x}}_{pob}$: vector containing the largest value of either $x_i \in X$
select(file);
getMinMax(file, $\underline{\boldsymbol{x}}_{pob}$, $\overline{\boldsymbol{x}}_{pob}$);
trim($\underline{\boldsymbol{x}}_{pob}$, $\overline{\boldsymbol{x}}_{pob}$);
adjustparameters(file);

Fig. 3. Pseudo-code of **Shrinkspace(file)**

```
m: number of constraints
i: constraint index
maxsize: max size of file
listsize: 15% of maxsize
constraintvalue(x,i): value of individual at constraint i
sortfile(file): sort file by objective function
worst(file,i): worst individual in file for constraint i
validconstraints={1,2,3,...,m};
i=firstin(validconstraints);
While (size(file) > listsize and size(validconstraints) > 0) {
    x=worst(file,i)
    if (x violates constraint i)
        file=delete(file,x)
    else validconstraints=removeindex(validconstraints,i)
if (size(validconstraints) > 0) i=nextin(validconstraints)
}
if (size(file) == listsize))
    list=file
else
    file=sort(file)
    list=copy(file,listsize) *pick the best listsize elements*
```

Fig. 4. Pseudo-code of **select(file)**

The function **select(file)** returns a list whose elements are the best individuals found in *file*. The size of the list set to 15% of *maxsize*. Thus, the goal of **select(file)** is to create a list with: 1) only the best feasible individuals, 2) a combination of feasible and partially feasible individuals, or 3) the "best" infeasible individuals. The selection algorithm is shown in Figure 4. Note that $validconstraints$ (a list of indexes to the problem constraints) indicates the order in which constraints are tested. One individual (the worst) is removed at a time in this loop of constraint testing till there is none to delete (all factible), or 15% of the file is reached. The function **getMinMax(file)** finds the extreme values of the decision variables represented by the individuals of the list. Thus, the vectors $\underline{\boldsymbol{x}}_{pob}$ and $\overline{\boldsymbol{x}}_{pob}$ are found. Function **trim**($\underline{\boldsymbol{x}}_{pob}$, $\overline{\boldsymbol{x}}_{pob}$) shrinks the feasible space around the potential solutions enclosed in the hypervolume defined by the vectors $\underline{\boldsymbol{x}}_{pob}$ and $\overline{\boldsymbol{x}}_{pob}$. Thus, the function **trim** ($\underline{\boldsymbol{x}}_{pob}$, $\overline{\boldsymbol{x}}_{pob}$) (see Figure 5) determines the new boundaries for the decision variables.

The value of β is the percentage by which the boundary values of either $x_i \in X$ must be reduced such that the resulting hypervolume H is a fraction α of its previous value. In IS-PAES all objective variables are reduced at the same rate β. Therefore, β can be deduced from α as discussed next. Since we need the new hypervolume to be a fraction α of the previous one, then

$$H_{\text{new}} \geq \alpha H_{\text{old}}$$

```
n: size of decision vector;
x̄_i: actual upper bound of the i_th decision variable
x̲_i: actual lower bound of the i_th decision variable
x̄_pob,i: upper bound of i_th decision variable in population
x̲_pob,i: lower bound of i_th decision variable in population
∀i : i ∈ { 1, ..., n }
    slack_i = 0.05 × (x̄_pob,i − x̲_pob,i)
    width_pob_i = x̄_pob,i − x̲_pob,i; width_i^t = x̄_i^t − x̲_i^t
    deltaMin_i = (β * width_i^t − width_pob_i) / 2
    delta_i = max(slack_i, deltaMin_i);
    x̄_i^{t+1} = x̄_pob,i + delta_i; x̲_i^{t+1} = x̲_pob,i − delta_i;
    if (x̄_i^{t+1} > x̄_original,i) then
        x̲_i^{t+1} −= x̄_i^{t+1} − x̄_original,i; x̄_i^{t+1} = x̄_original,i;
    if (x̲_i^{t+1} < x_original,i) then x̄_i^{t+1} += x̲_original,i − x̲_i^{t+1};
        x̲_i^{t+1} = x̲_original,i;
    if (x̄^{t+1} > x̄_original,i) then x̄_i^{t+1} = x̄_original,i;
```

Fig. 5. Pseudo-code of **trim**

$$\prod_{i=1}^{n}(\overline{x}_i^{t+1} - \underline{x}_i^{t+1}) = \alpha \prod_{i=1}^{n}(\overline{x}_i^{t} - \underline{x}_i^{t})$$

Either x_i is reduced at the same rate β, thus

$$\prod_{i=1}^{n}\beta(\overline{x}_i^{t} - \underline{x}_i^{t}) = \alpha \prod_{i=1}^{n}(\overline{x}_i^{t} - \underline{x}_i^{t})$$

$$\beta^n \prod_{i=1}^{n}(\overline{x}_i^{t} - \underline{x}_i^{t}) = \alpha \prod_{i}^{n}(\overline{x}_i^{t} - \underline{x}_{i=1}^{t})$$

$$\beta^n = \alpha$$

$$\beta = \alpha^{\frac{1}{n}}$$

Summarizing, the search interval of each decision variable x_i is adjusted as follows (the complete algorithm is shown in Figure 3):

$$width_{new} \geq \beta \times width_{old}$$

In our experiments, $\alpha = 0.90$ worked well in all cases. Since α controls the shrinking speed, it can prevent the algorithm from finding the optimum if small values are chosen. In our experiments, values in the range [85%,95%] were tested with no effect on the performance. The last step of **shrinkspace()** is a call to **adjustparameters(file)**. The goal is to re-start the control variable σ using: $\sigma_i = (\overline{x}_i - \underline{x}_i)/\sqrt{n} \;\; i \in (1, \ldots, n)$. This expression is also used during the initial generation of the EA. In that case, the upper and lower bounds take the initial values of the search space indicated by the problem. The variation of the mutation probability follows the exponential behavior suggested by Bäck [1].

6 Comparison of Results

We have validated our approach with the benchmark for constrained optimization proposed in [7]. Our results are compared against the homomorphous maps [6], which is representative of the state-of-the-art in constrained evolutionary optimization. The following parameters were adopted for IS-PAES in all the experiments reported next: $maxsize = 200$, $bestindividuals = 15\%$, $slack = 0.05$, $r = 400$. The maximum number of fitness function evaluations was set to 350,000, whereas the results of Koziel and Michalewicz [6] were obtained with 1,400,000 fitness function evaluations.

1. **g01**: Minimize: $f(\boldsymbol{x}) = 5\sum_{i=1}^{4} x_i - 5\sum_{i=1}^{4} x_i^2 - \sum_{i=5}^{13} x_i$ subject to: $g_1(\boldsymbol{x}) = 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \leq 0$, $g_2(\boldsymbol{x}) = 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \leq 0$, $g_3(\boldsymbol{x}) = 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \leq 0$, $g_4(\boldsymbol{x}) = -8x_1 + x_{10} \leq 0$, $g_5(\boldsymbol{x}) = -8x_2 + x_{11} \leq 0$, $g_6(\boldsymbol{x}) = -8x_3 + x_{12} \leq 0$, $g_7(\boldsymbol{x}) = -2x_4 - x_5 + x_{10} \leq 0$, $g_8(\boldsymbol{x}) = -2x_6 - x_7 + x_{11} \leq 0$, $g_9(\boldsymbol{x}) = -2x_8 - x_9 + x_{12} \leq 0$ where the bounds are $0 \leq x_i \leq 1$ $(i = 1, \ldots, 9)$, $0 \leq x_i \leq 100$ $(i = 10, 11, 12)$ 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. **g02**: Maximize: $f(\boldsymbol{x}) = \left| \frac{\sum_{i=1}^{n} \cos^4(x_i) - 2\prod_{i=1}^{n} \cos^2(x_i)}{\sqrt{\sum_{i=1}^{n} i x_i^2}} \right|$ subject to:

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

$$g_2(\boldsymbol{x}) = \sum_{i=1}^{n} x_i - 7.5n \leq 0 \tag{8}$$

   where $n = 20$ and $0 \leq x_i \leq 10$ $(i = 1, \ldots, n)$. The global maximum is unknown; the best reported solution is [11] $f(x^*) = 0.803619$. Constraint g_1 is close to being active ($g_1 = -10^{-8}$).
3. **g03**: Maximize: $f(\boldsymbol{x}) = (\sqrt{n})^n \prod_{i=1}^{n} x_i$ subject to: $h(\boldsymbol{x}) = \sum_{i=1}^{n} x_i^2 - 1 = 0$ 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. **g04**: Minimize: $f(\boldsymbol{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$ subject to:

$$\begin{aligned}
g_1(\boldsymbol{x}) &= 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 - 92 \leq 0 \\
g_2(\boldsymbol{x}) &= -85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 + 0.0022053x_3x_5 \leq 0 \\
g_3(\boldsymbol{x}) &= 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 - 110 \leq 0 \\
g_4(\boldsymbol{x}) &= -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 - 0.0021813x_3^2 + 90 \leq 0 \\
g_5(\boldsymbol{x}) &= 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 - 25 \leq 0 \\
g_6(\boldsymbol{x}) &= -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \leq 0
\end{aligned} \tag{9}$$

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

5. **g05** Minimize: $f(\boldsymbol{x}) = 3x_1 + 0.000001x_1^3 + 2x_2 + (0.000002/3)x_2^3$ subject to:

$$\begin{aligned}
g_1(\boldsymbol{x}) &= -x_4 + x_3 - 0.55 \leq 0 \\
g_2(\boldsymbol{x}) &= -x_3 + x_4 - 0.55 \leq 0 \\
h_3(\boldsymbol{x}) &= 1000\sin(-x_3 - 0.25) + 1000\sin(-x_4 - 0.25) + 894.8 - x_1 = 0 \\
h_4(\boldsymbol{x}) &= 1000\sin(-x_3 - 0.25) + 1000\sin(x_3 - x_4 - 0.25) + 894.8 - x_2 = 0 \\
h_5(\boldsymbol{x}) &= 1000\sin(-x_4 - 0.25) + 1000\sin(x_4 - x_3 - 0.25) + 1294.8 = 0
\end{aligned} \tag{10}$$

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

6. **g06** Minimize: $f(\boldsymbol{x}) = (x_1 - 10)^3 + (x_2 - 20)^3$ subject to:

$$\begin{aligned}
g_1(\boldsymbol{x}) &= -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \leq 0 \\
g_2(\boldsymbol{x}) &= (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \leq 0
\end{aligned} \tag{11}$$

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

7. **g07** Minimize:

$$\begin{aligned}
f(\boldsymbol{x}) = \; & x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 \\
& + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 \\
& + (x_{10} - 7)^2 + 45
\end{aligned} \tag{12}$$

subject to:

$$\begin{aligned}
g_1(\boldsymbol{x}) &= -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \leq 0 \\
g_2(\boldsymbol{x}) &= 10x_1 - 8x_2 - 17x_7 + 2x_8 \leq 0 \\
g_3(\boldsymbol{x}) &= -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \leq 0 \\
g_4(\boldsymbol{x}) &= 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \leq 0 \\
g_5(\boldsymbol{x}) &= 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \leq 0 \\
g_6(\boldsymbol{x}) &= x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \leq 0 \\
g_7(\boldsymbol{x}) &= 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 - 30 \leq 0 \\
g_8(\boldsymbol{x}) &= -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \leq 0
\end{aligned} \tag{13}$$

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. **g08** Maximize: $f(\boldsymbol{x}) = \frac{\sin^3(2\pi x_1)\sin(2\pi x_2)}{x_1^3(x_1+x_2)}$ subject to: $g_1(\boldsymbol{x}) = x_1^2 - x_2 + 1 \leq 0$, $g_2(\boldsymbol{x}) = 1 - x_1 + (x_2 - 4)^2 \leq 0$ 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. **g09** Minimize:

$$f(\boldsymbol{x}) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 \\ +10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7 \tag{14}$$

subject to:

$$\begin{aligned} g_1(\boldsymbol{x}) &= -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \leq 0 \\ g_2(\boldsymbol{x}) &= -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \leq 0 \\ g_3(\boldsymbol{x}) &= -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \leq 0 \\ g_4(\boldsymbol{x}) &= 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \leq 0 \end{aligned} \tag{15}$$

10. **g10** Minimize: $f(\boldsymbol{x}) = x_1 + x_2 + x_3$ subject to:

$$\begin{aligned} g_1(\boldsymbol{x}) &= -1 + 0.0025(x_4 + x_6) \leq 0 \\ g_2(\boldsymbol{x}) &= -1 + 0.0025(x_5 + x_7 - x_4) \leq 0 \\ g_3(\boldsymbol{x}) &= -1 + 0.01(x_8 - x_5) \leq 0 \\ g_4(\boldsymbol{x}) &= -x_1x_6 + 833.33252x_4 + 100x_1 - 83333.333 \leq 0 \\ g_5(\boldsymbol{x}) &= -x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \leq 0 \\ g_6(\boldsymbol{x}) &= -x_3x_8 + 1250000 + x_3x_5 - 2500x_5 \leq 0 \end{aligned} \tag{16}$$

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. **g11** Minimize: $f(\boldsymbol{x}) = x_1^2 + (x_2 - 1)^2$ subject to: $h(\boldsymbol{x}) = x_2 - x_1^2 = 0$ where: $-1 \leq x_1 \leq 1$, $-1 \leq x_2 \leq 1$. The optimum solution is $x^* = (\pm 1/\sqrt{2}, 1/2)$ where $f(x^*) = 0.75$.

The comparison of results is summarized in Table 1. It is worth indicating that IS-PAES converged to a feasible solution in all of the 30 independent runs performed. The discussion of results for each test function is provided next (HM stands for homomorphous maps):

For **g01** both the best and the mean results found by IS-PAES are better than the results found by HM, although the difference between the worst and the best result is higher for IS-PAES. For **g02**, again both the best and mean results found by IS-PAES are better than the results found by HM, but IS-PAES has a (slightly) higher difference between its worst and best results. In the case of **g03**, IS-PAES obtained slightly better results than HM, but in this case it also has a lower variability. It can be clearly seen that for **g04**, IS-PAES had a better performance with respect to all the statistical measures evaluated. The same applies to **g05** in which HM was not able to find any feasible solutions (the best result found by IS-PAES was very close to the global optimum). For **g06**, again IS-PAES found better values than HM (IS-PAES practically converges to the optimum in all cases). For **g07** both the best and the mean values produced by IS-PAES were better than those produced by HM, but the difference between the worst and best

Table 1. Comparison of the results for the test functions from [7]. Our approach is called IS-PAES and the homomorphous maps approach [6] is denoted by HM. N.A. = Not Available.

		BEST RESULT		MEAN RESULT		WORST RESULT	
TF	**OPTIMAL**	**IS-PAES**	**HM**	**IS-PAES**	**HM**	**IS-PAES**	**HM**
g01	-15.0	-14.995	-14.7864	-14.909	-14.7082	-12.4476	-14.6154
g02	-0.803619	-0.8035376	-0.79953	-0.798789	-0.79671	-0.7855539	-0.79199
g03	-1.0	-1.00050019	-0.9997	-1.00049986	-0.9989	-1.00049952	-0.9978
g04	-30665.539	-30665.539	-30664.5	-30665.539	-30655.3	-30665.539	-30645.9
g05	5126.498	5126.99795	N.A.	5210.22628	N.A.	5497.40441	N.A.
g06	-6961.814	-6961.81388	-6952.1	-6961.81387	-6342.6	-6961.81385	-5473.9
g07	24.306	24.3410221	24.620	24.7051034	24.826	25.9449662	25.069
g08	-0.095825	-0.09582504	-0.0958250	-0.09582504	-0.0891568	-0.09582504	-0.0291438
g09	680.630	680.638363	680.91	680.675002	681.16	680.727904	683.18
g10	7049.331	7055.11415	7147.9	7681.59187	8163.6	9264.35787	9659.3
g11	0.750	0.75002984	0.75	0.74992803	0.75	0.74990001	0.75

result is slightly lower for HM. For **g08** the best result found by the two approaches is the optimum of the problem, but IS-PAES found this same solution in all the runs performed, whereas HM presented a much higher variability of results. In **g09**, IS-PAES had a better performance than HM with respect to all the statistical measures adopted. For **g10** none of the two approaches converged to the optimum, but IS-PAES was much closer to the optimum and presented better statistical measures than HM. Finally, for **g11**, HM presented slighly better results than IS-PAES, but the difference is practically negligible.

Summarizing, we can see that IS-PAES either outperformed or was very close to the results produced by HM even when it only performed 25% of the fitness functions evaluations of HM. IS-PAES was also able to approach the global optimum of **g05**, for which HM did not find any feasible solutions.

7 Conclusions and Future Work

We have presented a new constraint-handling approach that combines multiobjective optimization concepts with an efficient reduction mechanism of the search space and a secondary population. We have shown how our approach overcomes the scalability problem of the original PAES (which was proposed exclusively for multiobjective optimization) from which it was derived, and we also showed that the approach is highly competitive with respect to a constraint-handling approach that is representative of the state-of-the-art in the area.

The proposed approach illustrates the usefulness of multiobjective optimization concepts to handle constraints in evolutionary algorithms used for single-objective optimization. Note however, that this mechanism can also be used for multiobjective optimization and that is in fact part of our future work.

Another aspect that we want to explore in the future is the elimination of all of the parameters of our approach using online or self-adaptation. This task, however, requires

a careful analysis of the algorithm because any online or self-adaptation mechanism may interefere with the mechanism used by the approach to reduce the search space.

Acknowledgments

The first and second authors acknowledge support from CONACyT project No. P-40721-Y. The third author acknowledges support from NSF-CONACyT project No. 32999-A.

References

1. Thomas Bäck. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, New York, 1996.
2. 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.
3. Carlos A. Coello Coello, David A. Van Veldhuizen, and Gary B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York, May 2002. ISBN 0-3064-6762-3.
4. David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.
5. Joshua D. Knowles and David W. Corne. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8(2):149–172, 2000.
6. Slawomir Koziel and Zbigniew Michalewicz. Evolutionary Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization. *Evolutionary Computation*, 7(1):19–44, 1999.
7. Zbigniew Michalewicz and Marc Schoenauer. Evolutionary Algorithms for Constrained Parameter Optimization Problems. *Evolutionary Computation*, 4(1):1–32, 1996.
8. 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, University of Plymouth.
9. Tapabrata Ray, Tai Kang, and Seow Kian Chye. An Evolutionary Algorithm for Constrained Optimization. In Darrell Whitley et al., editor, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2000)*, pages 771–777, San Francisco, California, 2000. Morgan Kaufmann.
10. 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.
11. T.P. Runarsson and X. Yao. Stochastic Ranking for Constrained Evolutionary Optimization. *IEEE Transactions on Evolutionary Computation*, 4(3):284–294, September 2000.
12. 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.
13. Patrick D. Surry and Nicholas J. Radcliffe. The COMOGA Method: Constrained Optimisation by Multiobjective Genetic Algorithms. *Control and Cybernetics*, 26(3):391–412, 1997.