IS-PAES: switching constraints on and off for Multiobjective Optimization

Arturo Hernández Aguirre
Salvador Botello Rionda
Giovanni Lizárraga Lizárraga
Center for Research in Mathematics, México
artha,botello,giovanni@cimat.mx

Carlos A. Coello Coello
CINVESTAV-IPN Computer Science Section
México, D.F. 07300, México
ccoello@cs.cinvestav.mx

Abstract- In this article we introduce Inverted and Shrinkable Pareto Archived Evolutionary Strategies, IS-PAES, an evolutionary algorithm for multiple objective optimization with constraint handling. IS-PAES inherits from PAES the use of an adaptable grid to keep diversity, but here this grid can grow and shrink dinamically until the constraints are met. We also propose a novel approach to select a mixture of promising individuals, and Pareto dominance computed over objective functions plus unfeasible constraints. Several examples of the literature are used to show the potential of ISPAES.

1 Introduction

The success 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. Such a mechanism is highly desirable since most real-world problems have constraints which could be of any type (equality, inequality, linear and nonlinear). Nevertheless, EAs lack a mechanism able to bias efficiently the search towards the feasible region in constrained search spaces. One popular approach to constraint handling is to treat the constraints as objective functions, and then solve the problem for all of them. IS-PAES follows this procedure, and also as PAES [1], selection is based on Pareto dominance. Nonetheless, we introduce a novel strategy that makes a difference: when individuals are located in the feasible region the constraints are not used to determine Pareto dominance. This idea has proved powerful since so far IS-PAES has reported some of the best results for single-objective optimization problems with constraints [26]. The remainder of this paper is organized as follows. Section 2 provides the problem definition. Section 3 introduces concepts used in the article. Section 4 describes some work related to our own. In Section 5, we describe the main algorithm of IS-PAES. Section 6 provides a comparison of results and Section 7 draws our conclusions and provides some paths of future research.

2 Problem Statement

We are interested in the general non-linear programming problem in which we want to:

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

subject to:

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

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

where $\vec{F}$ is the vector of objective functions $\vec{F} = [f_1(\vec{x}), \dots, f_k(\vec{x})]$, $\vec{x}$ is the vector of solutions $\vec{x} = [x_1, x_2, \dots, x_r]^T$, 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 $\mathcal{F}$.

3 Basic Concepts

The main idea in adopting multiobjective optimization concepts to handle constraints is to redefine the global optimization problem of $\vec{f}(\vec{x})$ as a multiobjective optimization problem in which we will have $k + m$ objectives, where m is the total number of constraints and k the number of objective functions. Then, we can apply any multiobjective optimization technique to the new vector $\bar{v} = (f(\vec{x}), f_1(\vec{x}), \dots, f_{k+m}(\vec{x}))$, where $f_1(\vec{x}), \dots, f_k(\vec{x})$ are the original objectives 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).

Pareto dominance means one individual dominates a second individual if the first is better in at least one of the objectives while the other objectives remain with no change (a$\preceq$b means a dominates b). Based on this main idea, several approaches have been proposed in the last few years. Some of them use population-based techniques (e.g., [3]), others use Pareto dominance in the selection mechanism of the EA (e.g., [2]), and others use Pareto ranking (e.g., [4]). However, all of these techniques are normally more useful to approach the feasible region, but are not as effective for reaching the global optimum of a problem. We argue in this paper that the main reason for this limitation has to do with the artificial "trade-off" created when constraints are added as new objectives (as noted at the beginning of this Secttion). Rather than focusing the effort on finding such artificial "trade-offs", IS-PAES proposes to focus the search in finding the boundary between the feasible and the infeasible regions and then concentrating the search effort on reaching the global optimum of the original objective function space. Therefore, for IS-PAES the Pareto "trade-off" neglects feasible constraints, therefore, any constraint satisfied is as important as any other, and in fact, they are not used to test Pareto dominance once the individuals are in the feasible interval of the constraint. Such is the nature of the algorithm proposed in this paper.

4 Related work

Three 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. Coello and Mezura [17] implemented a version of the Niched-Pareto Genetic Algorithm (NPGA) [19] 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. Camponogara & Talukdar [20] proposed an approach in which a global optimization problem was transformed into a bi-objective problem where the first objective is to optimize the original objective function and the second is to minimize:

   $$\Phi(\mathbf{x}) = \sum_{i=1}^{n} max(0, g_i(\mathbf{x})) \quad (4)$$

   Equation (4) 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 approach similar to a min-max formulation used in multiobjective optimization [23] combined with tournament selection was proposed by Jiménez and Verdegay [24].

   Jiménez et al. [25] 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).

2. Use of Pareto ranking [5] to assign fitness in such a way that nondominated individuals (i.e., feasible individuals in this case) are assigned a higher fitness value. Surry & Radcliffe [4] used a combination of the Vector Evaluated Genetic Algorithm (VEGA) [6] and Pareto Ranking to handle constraints in an approach called COMOGA (Constrained Optimization by Multi-Objective Genetic Algorithms). Ray et al. [21] 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.

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) [6]. Parmee & Purchase [15] proposed to use VEGA [6] 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.

5 IS-PAES Algorithm

IS-PAES has been implemented as an extension of the Pareto Archived Evolution Strategy (PAES) proposed by Knowles and Corne [1] for multiobjective optimization. PAES main feature is the use of an adaptive grid over objective function space. Such a grid is the diversity maintenance mechanism of PAES and its the main feature of this algorithm. Thus, the first feature introduced in IS-PAES is the "inverted" part of the algorithm that deals with this space usage problem. In IS-PAES, global information carried by the individuals surrounding the feasible region is used to concentrate the search effort on feasible areas as the evolutionary process takes place. In consequence, the search space being explored is "shrinked" over time, by cutting off unfeasible regions from the search space. Eventually, the size of the search space being inspected will be the feasible region. The combination of this shrinking concept with the usage of a secondary population (here after called external memory or file, whose size is maxsize), and avoiding to test Pareto dominance with valid constraints is a novel proposal. The main algorithm of IS-PAES is shown next (more experimental results are available in [26]).

```
Main algorithm of IS-PAES
    $maxsize$: max size of file
    $c$: current parent $\in$ X (decision variable space)
    $h$:child of c $\in$ X
    a$_h$: individual in file $\preceq h$
    $a_d$: 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 $\leq$MaxNew do
        h = mutate(c); cnew+=1;
        if (c$\preceq$h) then label A
        else if (h$\preceq$c) then {remove(c); add(g); c=h; }
        else if($\exists a_h \in$ file $| a_h \preceq$ h) then label A
        else if ( $\exists a_d \in$ file $| h \preceq a_d$ } then{
            add( h ); $\forall a_d \{remove(a_d)$; 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
```

Any time Pareto dominance is tested, for example h$\preceq$c, all original objective functions are used, plus **unfeasible constraints**. This is, assume the problem has two objective functions plus three constraints that are unfeasible during the first generations. At that time Pareto dominance is verified by using the five functions, then one constraint becomes valid so dominance is tested by using four functions. When all three constraints are valid, Pareto dominance is determined only through the original two objective functions. Every g number of generations, a new parent is chosen from the less populated area of the grid. The overall effect is to re-start the search as to distribute the population along the Pareto front. Every r number of generations the unfeasible region is cutted off (if any) by calling shrinkspace(file). The function **test(h,c,file)** determines how an individual can be added to the external memory. If there is space in file then the child is simply inserted and the next parent is chosen from the less populated area. If the file is full, one element has to be removed in order to insert the new child, but the condition for the child to enter the file is to remove one element from a highly dense populated grid location. Here we introduce the following notation: $x_1 \Box x_2$ means x_1 is located in a less populated region of the grid that x_2. The pseudo-code of this function is shown next.

Pseudo-code of test(h,c,file)

```
if (current $<$ maxsize) then
    add(h);
    if (h $\Box$ c) then c=h
else if ($\exists a_p \in$ file $|h \Box a_p$) then
    remove($a_p$); add(h)
    if (h $\Box$ c) then c = h;
```

5.1 Inverted “ownership”

PAES algorithm keeps a list of individuals on each 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.

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. IS-PAES removes from the file the worst individuals by calling the function *select(file)*. The new boundary of each decision variable is calculated by calling *getMinMax* (at this point IS-PAES is only using the elements found by select(file)).The third and last step of shrinkspace is to call function trim(). Trim will never cut off area from the feasible region. The pseudo-code of shrinkspace(file) is shown next.

Pseudo-code of Shrinkspace(file)

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

The description of each component is given next.

1. select(file). Although the goal is to pick the best individuals, a second but equally important task of select(file) is to preserve diversity through the balance between feasible individuals and acceptable unfeasible ones. The function select(file) is shown next.

Pseudo-code of select(file)

```
$m$ : number of constraints
$mr$ : number of violated constraints
$i$ : constraint index
$maxsize$ : max size of file
$listsize$ : 15% of maxsize
$constraintvalue(x, i)$ : value of individual
        at constraint i
$mrfile(file, mr)$ : calculate mr
$worst(file, i)$ : worst individual in file for
        constraint i
$validconstraints = \{1, 2, 3, ..., m\}$
i=firstin(validconstraints);
mrfile(file,mr);
if (mr $<$ listsize) listsize=mr;
While (size(file) $>$ listsize and
            size(validconstraints) $>$ 0) {
    x=worst(file,i)
    if (x violates constraint i)
        file=delete(file,x)
    else valid constraints=
        remove index(valid constraints,i)
    if (size(valid constraints) $>$ 0)
        i=nextin(valid constraints)
}
```

The function select(file) returns a list whose elements are the best individuals found in file. The size of this list is 15% of maxsize. Since individuals could be feasible, unfeasible or only partially feasible, the mixture is generated as follows: IS-PAES loops over a list of constraint indexes, removing for each index the worst unfeasible element (one) at a time. When all individuals are feasible for some index, the index is removed from the list. This approach keeps the diversity of the population. If all individuals are feasible the algorithm does not remove any, and returns 15% of the file. This resulting list contains: 1) only the best feasible individuals, or 2) a combination of feasible and partially feasible, or 3) the “best” unfeasible individuals. Note $validconstraints$ is an ordered list of indexes. Another approach could be to store the constraint indexes in random order, or to shuffle the list every given number of generations so the constraints are really tested in random order. The three approaches have been tested and none seems to excel over the others, thus we show here the approach used in this paper (testing the constraints in fixed order). It is importante to note here how selec(file) does not use a greedy approach based on feasibility and Pareto dominance.

2. getMinMax(file). The function getMinMax(file) takes the mentioned list and finds the extreme values of the decision variables represented by those individuals. Thus, the vectors $\underline{x}_{pob}$ and $\overline{x}_{pob}$ are found.
3. trim($\underline{x}_{pob}$, $\overline{x}_{pob}$). Function trim()shrinks the feasible space around the potential solutions enclosed in the hypervolume defined by the vectors $\underline{\boldsymbol{x}}_{pob}$ and $\overline{\boldsymbol{x}}_{pob}$. Function $trim$ is shown next.

Pseudo-code of trim

```
$n$: size of decision vector;
$\overline{x}_i$: actual upper bound of the $i_{th}$ decision variable
$\underline{x}_i$: actual lower bound of the $i_{th}$ decision variable
$\overline{x}_{pob,i}$: upper bound of $i_{th}$ decision variable
        in population
$\underline{x}_{pob,i}$: lower bound of $i_{th}$ decision variable
        in population $\forall i$ : i ∈ { 1, . . . , n }
$slack_i = 0.05 \times (\overline{x}_{pob,i} - \underline{x}_{pob,i})$
$width\_pob_i = \overline{x}_{pob,i} - \underline{x}_{pob,i}$; $width_i^t = \overline{x}_i^t - \underline{x}_i^t$
$deltaMin_i = \frac{\beta * width_i^t - width\_pob_i}{2}$
delta$_i$ = max(slack$_i$, deltaMin$_i$);
$\overline{x}_i^{t+1} = \overline{x}_{pob,i} + delta_i$; $\underline{x}_i^{t+1} = \underline{x}_{pob,i} - delta_i$;
if ($\overline{x}_i^{t+1} > \overline{x}_{original,i}$) then
    $\underline{x}_i^{t+1} -= \overline{x}_i^{t+1} - \overline{x}_{original,i}$;
    $\overline{x}_i^{t+1} = \overline{x}_{original,i}$;
if ($\underline{x}_i^{t+1} < x_{original,i}$) then {
    $\overline{x}_i^{t+1} += \underline{x}_{original,i} - \underline{x}_i^{t+1}$;
    $\underline{x}_i^{t+1} = \underline{x}_{original,i}$;}
if ($\overline{x}^{t+1} > \overline{x}_{original,i}$) then $\overline{x}_i^{t+1} = \overline{x}_{original,i}$;
```

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 be a fraction α of the previous one,

$$H_{\text{new}} \geq \alpha H_{\text{old}} \tag{5}$$

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

Either x_i is reduced at the same rate β, thus

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

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

$$\beta^n = \alpha$$

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

In short, the search interval of each decision variable x_i is adjusted as follows:

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

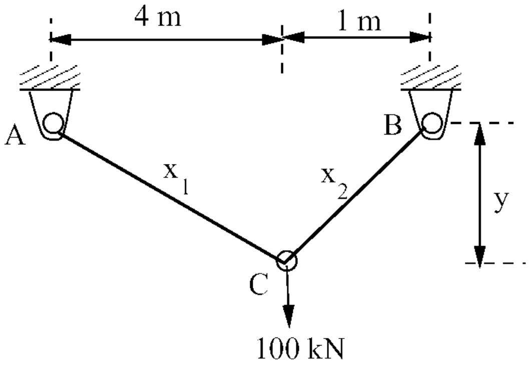

Figure 1: Optimization of a two-bar truss

In our experiments, $\alpha = 0.90$ worked well in all cases. Clearly, α controls the shrinking speed, hence the algorithm is sensitive to this parameter and it can prevent it from finding the optimum solution if small values are chosen. In our experiments, values in the range [85%,95%] were tested with no visible effect in the performance. Of course, α values near to 100% slow down the convergence speed.

The mutation of the control variable sigma follows the exponential behavior suggested by Bäck [8]. The initial value of either σ_i is calculated as follows:

$$\sigma_i = \overline{x}_i - \underline{x}_i / \sqrt{n} \quad i \in (1, \ldots, n) \tag{6}$$

6 Experiments

In all the following examples of this section we used the following parameters: The size of the file is 100. All the members of the file could participate in the new generation. Shrinkspace is called every 2 generations, also a new parent is chosen every 2 generations ($r = g = 2$), and reduction rate of the hypervolume is 10% ($\alpha = 0.9$). We used 500 generation in each problem.

6.1 Optimization of a two-bar truss

The first problem we report here is the optimization of a two-bar truss with two objective functions and one constraint. The truss, that has to carry a load of 100 kN, is shown in Figure 1.

The objectives are the minimization of the volume (designing for the minimum cost of fabrication) and the minimization of the maximum stress on each bar. This problem was originally studied using the $\epsilon-$constraint method[14] as a two-objective optimization problem with y as the only variable.

$$\begin{aligned} Minimize f_1(x) &= x_1\sqrt{(16+y^2)} + x_2\sqrt{(1+y^2)} \\ Minimize f_2(x) &= \max(\sigma_{AC}, \sigma_{BC}) \end{aligned}$$

subject to: $max(\sigma_{AC}, \sigma_{BC}) \leq 10^5; 1 \leq y \leq 3$

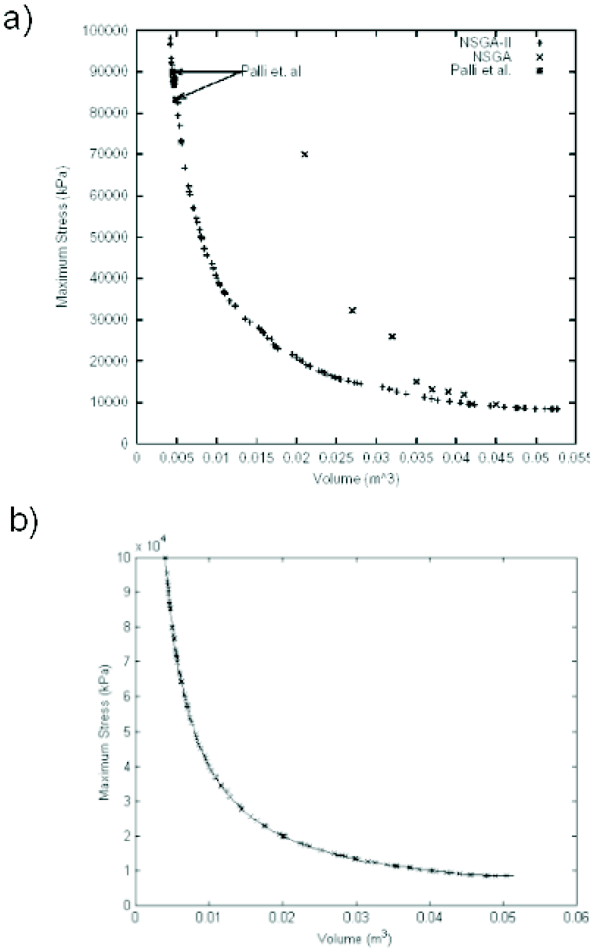

Figure 2: Pareto fronts of the 2-bar truss

The stresses are calculated with a close form:

$$\sigma_{AC} = \frac{20\sqrt{(16+y^2)}}{yx_1}; \sigma_{BC} = \frac{80\sqrt{(1+y^2)}}{yx_2}$$

In Figure 2 we show the real Pareto-optimal solution (calculated by enumeration) and the front reported by the ISPAES algorithm. The solution is spread over the following range: (.004 m^3, 100000 kPa) and (0.051387m^3, 8432.740427 kPa). ISPAES found a smooth front and the totality of the points are very close of the real Pareto front. We are comparing our results with those published by Deb, Patratap and Moira [13], where they include results with NSGA[10], NSGA-II[11] and the $\epsilon-$constraint method[14].

6.2 Optimization of a 10-bar Plane Truss

The next engineering optimization problem is the optimization of the 10-bar plane truss shown in Figure 3.

Single objective optimization We want to find the coss-sectional area of each bar of this truss such that its weight is minimized, subject to stress and displacement constraints. The weight of the truss is given by:

$$F(\mathbf{x}) = \sum_{j=1}^{10} \gamma A_j L_j \quad (7)$$

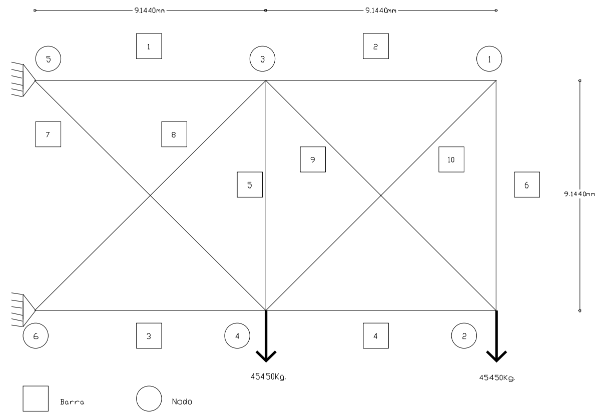

Figure 3: Optimization of a 10-bar plane truss

where: $\mathbf{x}$ is a candidate solution, A_j is the cross-sectional area of the jth member. L_j is the length of member j and γ is the volumetric weight of the material.

The maximum allowable displacement for each node (vertical and horizontal) is assumed as 5.08 cm. There are 10 stress constraints and 8 displacement constraints in total. The maximum and minimum allowable value for the cross-sectional areas are 0.5062 cm^2 and 999.0 cm^2, respectively. The remaining assumed data are: Young's modulus E = 7.3$\times 10^5$ kg/cm^2, maximum allowable stress = 1742.11 kg/cm^2, γ = 7.4239$\times 10^{-3}$ kg/cm^3, and a vertical load of -45454.0 kg applied at nodes 2 and 4.

Table 1 shows the minimum value found for this problem by IS-PAES and three different heuristic algorithms: GSSA (general stochastic search algorithm with a population size of five, crossover rate of zero, and mutation rate $[0, 10]$/(number_of_bars) [27], and simulated annealing with $\alpha = 1.001$), VGA (variable-length genetic algorithm of Rajeev and Krishnamoorthy [28], with population size of 50), and ISA (Iterated Simulated Annealing, of Ackley [29].

Element	IS-PAES	GSSA	VGA	ISA
1	190.53	205.17	206.46	269.48
2	0.6466	0.6452	0.6452	79.810
3	146.33	134.20	151.62	178.45
4	95.07	90.973	103.23	152.90
5	0.6452	0.6452	0.6452	70.390
6	3.0166	0.6452	0.6452	10.260
7	47.677	55.487	54.84	147.87
8	129.826	127.75	129.04	14.710
9	133.282	133.56	132.27	156.06
10	0.6452	0.6452	0.6452	87.740
V (cm^3)	801624.5	805777	833258	1313131
W (kg)	5951	6186	6186	9750

Table 1: Comparison of weights for the 10-bar plane truss

We can see in Table 1 that IS-PAES found better results than any of the other methods.

Multiple objective optimization The two objective functions are the minimization of the weight structure, and the vertical deformation of node 2, subject to the original

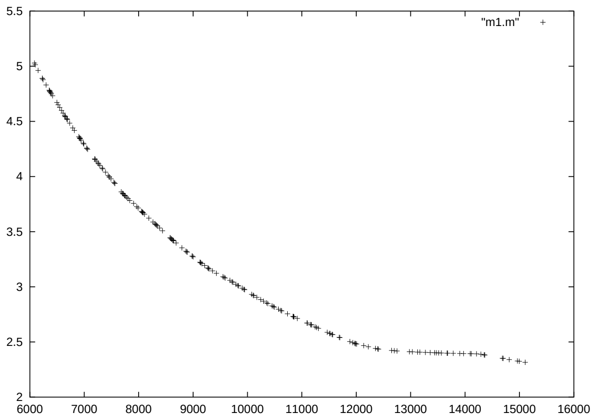

Figure 4: Pareto front of the 10-bar plane truss (weight vs displacement)

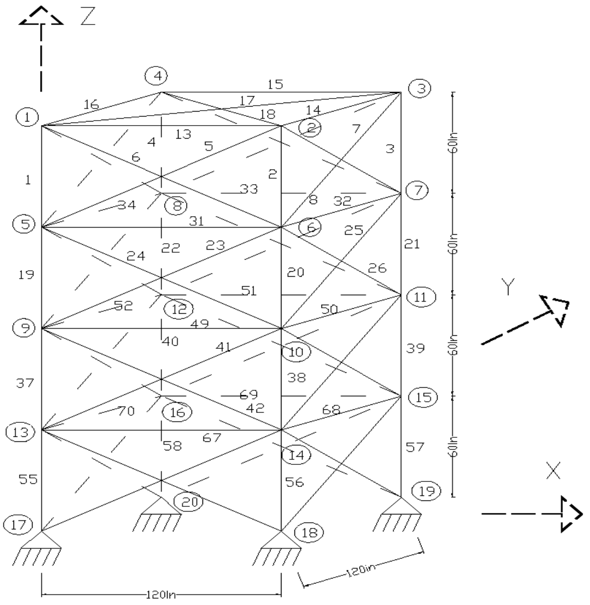

Figure 5: Optimization of 72-bar 3D structure

constraints. In Figure 4 we show the Pareto front for this problem.

6.3 Optimization of 72-bar 3D structure

The next problem is the design of 72-bar 3D structure (transmission tower) shown in Figure 5, has been addressed elsewhere in the literature [3].

The truss is subject to two distinct loading conditions and sixteen independent design variables. All nodes subject to displacement constraint $\Delta \leq 0.25$ inches in x and y direction. All bars have a stress constraint -1759.25 kg/cm^2 $\leq (\sigma_a)_i \leq 1759.25$ kg/cm^2, $i = 1, 2 \ldots 72$. The minimum size constraint is 0.254 cm^2 $\leq A_i, i = 1, 2 \ldots 72$. The properties of the material are: modulus of elasticity: 7.031×10^6kg/cm^2, volumetric weigth: 2.77×10^{-3} kg/cm^3. The first loading condition has a point load in node 1 with 2270 kg in x direction, 2270 kg in y direction and -2270 kg in z direction. The second loading condition has four load points in nodes 1,2,3 and 4, with -2270 kg in z direction. The design problem is the design of the truss for both loading conditions. In Table 2 we give the group description of the truss.

Group Number	Member
1	A_1-A_4
2	A_5-A_{12}
3	A_{13}-A_{16}
4	A_{17}-A_{18}
5	A_{19}-A_{22}
6	A_{23}-A_{30}
7	A_{31}-A_{34}
8	A_{35}-A_{36}
9	A_{37}-A_{40}
10	A_{41}-A_{48}
11	A_{49}-A_{52}
12	A_{53}-A_{54}
13	A_{55}-A_{58}
14	A_{59}-A_{66}
15	A_{67}-A_{70}
16	A_{71}-A_{72}

Table 2: 72-bar 3D cross sections by group

We solve this problem in continuos (without a catalog) and discrete (with a catalog) search spaces.

1. Continuos seach space solution We compare IS-PAES against several results of other authors in Table 3; as it can be observed IS-PAES provides the best solution. In Table 4 we show basic statistics for 30 runs.

2. Discrete search space solution We solved three cases of this problem using the catalog of *Altos Hornos de México, S.A.* with 65 entries for the cross-sectional areas: 1) stress constraints only; 2) stress and displacement constraints; 3) displacement constraints, and considers bar traction and compression stress, as well as their proper weight. The values of material properties and constraints remain with no change for all three cases. Solutions for 3 cases are shown in Table 5

6.4 Optimization of a steel dome

Our last experiment is devoted to the optimization of the weight of the steel dome shown Figure 6 for discrete search space (using the catalog of *Altos Hornos de México, S.A.*)

Single objective optimization The truss has seven independent design variables. For all nodes, displacement constraint $\Delta \leq 20$ mm in z direction. All bars subject to the stress constraint given by the mentioned catalog. The material properties are: modulus of elasticity= 2.1x10^{6}kg/cm^2, yield strength: 2750 kg/cm^2. The loading conditions are a punctual load in z direction with different magnitude: -500 kg in node 1; -40 kg in nodes 17, 23, 29, and 35; -120 kg in nodes 16, 18, 22, 24, 28, 30, 34, and 36; -200 kg on the rest of the nodes.

We solved three cases of this problem using the catalog

Algorithm	Best Minimun Weight (Kg)
IS-PAES	172.02
Venkayya [30]	173.06
Gellatly [31]	179.77
Renwei [32]	172.36
Schmit[33]	176.44
Xicheng [34]	172.90
GAOS [27]	173.94

Table 3: ISPAES vs results of several authors for 72-bar 3D structure

Parameter	Weight (Kg)
Best	172.02
Worst	172.09
Mean	172.05
Std. dev.	0.015
Median	172.04
Fact. Sol.	30

Table 4: ISPAES statistics for 72-bar 3D

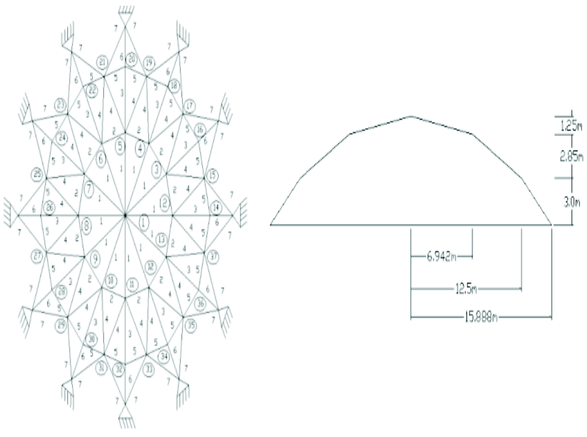

Figure 6: Optimization of a steel dome

Parameter	Case1 (Kg)	Case2 (Kg)	Case3 (Kg)
Best	92.3295	192.7194	630.400
Worst	92.3295	193.4353	640.3640
Mean	92.3295	192.9098	633.2354
Std. dev.	0.0	0.3060	2.7371
Median	92.3295	192.7194	632.9665
Fact. Sol.	30	30	30

Table 5: IS-PAES solutions to 72-bar structure using a catalog

Parameter	Case1 (Kg)	Case2 (Kg)	Case3 (Kg)
Best	703.57	703.57	13642.33
Worst	703.57	703.57	13651.93
Mean	703.57	703.57	13644.56
Std. dev.	0.0	0.0	4.1304
Median	703.57	703.57	13642.33
Fact. Sol.	30	30	30

Table 6: IS-PAES solutions to steel dome using a catalog

of *Altos Hornos de México, S.A.* with 65 entries for the cross-sectional areas: 1) stress constraints only; 2) stress and displacement constraints; 3) displacement constraints, and considers bar traction and compression stress, as well as their proper weight. The values of material properties and constraints remain with no change for all three cases. Solutions for 3 cases are shown in Table 6. Columns “Case1” and “Case2” are equal because displacement constraints are too small.

Multiple objective optimization The two objective functions are the minimization of the weight structure, and the vertical deformation of central node, subject to the original constraints. In Figure 7 we show the Pareto front for this problem.

7 Conclusions

We have proposed a new multiobjective optimization algorithm whose selection method is based on Pareto dominance but dominance is only tested over objective functions and unfeasible constraints. Constraints are “turned off” as they become valid and not used to test Pareto dominance. This mechanism focuses the search in the original “trade-off” of the objective functions, neglecting the artificial “trade-off” created when constraints are added as new objectives. IS-PAES determines the feasible region by bounding the hypervolume that contains the population and by removing unfeasible individuals through function select(file). Select(file) tests constraints in random order to remove unfeasible individuals preserving a good mixture of individuals and therefore population diversity. All solutions found by IS-PAES are highly competitive and among the bests reported in the literature for a selection method based on Pareto dominance.

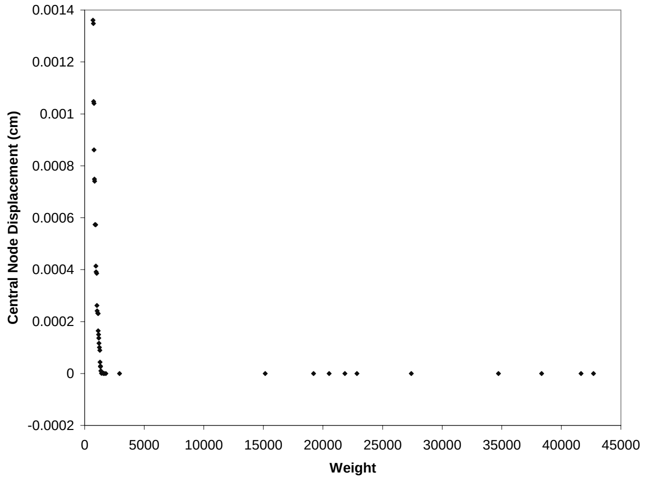

Figure 7: Pareto front of the steel dome

Acknowledgements

The first author acknowledges partial support from CONCyTEG project No. 03-02-K118-037. The second author acknowledges support from CONACyT through project P40721-Y. The last author acknowledges support from CONACyT through project 32999-A. We also thank Dr. Kalyanmoy Deb for the use of Figure 2

Bibliography

[1] J.D. Knowles and D.W. Corne, (2000),*Approximating the Nondominated Front using the Pareto Archived Evolution Strategy*, Evolutionary Computation, Vol 8, No. 2, 149–172

[2] Carlos A. Coello Coello,(2000),*Constraint-handling using an evolutionary multiobjective optimization technique*, Civil Engineering and Environmental Systems, Vol 17, pp 319–346

[3] Carlos A. Coello Coello,(2000),*Treating constraints as objectives for single-objective evolutionary optimization*, Engineering Optimization, Vol 32, No 3, pp 275–308

[4] Patrick D. Surry and Nicholas J. Radcliffe, (1997),*The COMOGA Method: Constrained Optimizationptimisation by Multiobjective Genetic Algorithms*, Control and Cybernetics, Vol 26, No 3, pp 391–412

[5] David Goldberg, (1989), Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Publishing Company, Reading, MA.

[6] J.D. Shaffer, (1985), Multiple Objective Optimization with Vector Evaluated Genetic Algorithm. In Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms, pp 93–100

[7] F.Y. Cheng and X.S. Li, (1999), *Generalized Center Method for Multiobjective Engineering Optimization*, Engineering Optimization, Vol 31, pp 641–661

[8] Thomas Bäck, (1996), *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, New York

[9] Surry, P.D., Radcliffe, N.J., Boyd, I.D. *Patrick D. Surry, Nicholas J. Radcliffe, Ian D. Boyd: A Multi-objective Approach to Constrained Optimization 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, Sheffield, U.K., 1995. Springer-Verlag.

[10] Srinivas, N. and Deb, K. (1995). *Multiobjective function optimization using nondominated sorting genetic algorithms*, Evolutionary Computation, 2(3), 221–248.

[11] Deb. K ,Amrit Pratap, Sameer Agarwal and T. Meyarivan (2000). *A fast and elitist multi-objective genetic algorithm-NSGA-II.* KanGAL Report Number 2000001, Indian Institute of Technology, Kanpur, India, 2000.

[12] Deb. K , and Goyal, M. (1998). *A Robust optimization procedure for mechanical component design based on genetic adaptive search*. Trans. of the ASME: Journal of Mechanical Design, 120 (2), 162-164.

[13] Deb, K., Pratap, A and Moita S.(2000). *Mechanical component design for multiple objectives using elitist non-dominated sorting GA*. Technical report 200002, KanGAL http://www.iitk.ac.in/kangal

[14] PAlli, N., Azram, S., McCluskey, P and Sundararajan, R. (1999). An interactive multistage ϵ-inequality constraint method for multiple objectives decision making. ASME Journal of Mechanical Design, 120(4), 678-686.

[15] 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.

[16] 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 Kauffman Publishers.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] Tapabrata Ray and K.M. Liew. A Swarm Metaphor for Multiobjective Design Optimization. *Engineering Optimization*, 34(2):141–153, March 2002.

[23] V. Chankong and Y.Y. Haimes. (1983) *Multiobjective decision making: theory and methodology*, In Andrew P. Sage, editor. Systems Science and Engineering. North Holland

[24] Fernando Jiménez, José L. Verdegay, and Antonio F. Gómez-Skarmeta. (1999) *Evolutionary Techiniques for Constrained Multiobjective Optimization Problems*. In Annie S. Wu, editor. Proceedings of the 1999 GECCO Conference. Workshop programs, pp 115-116

[25] Fernando Jiménez and Antonio F. Gómez-Skarmeta and G. Sánchez. (2002) *How Evolutionary Multiobjective Optimization can be used for Goals and Priorities based Optimization*. In E. Alba et. al. editors. Primer Congreso Español de Algoritmos Evolutivos y Bioinspirados (AEB'02). Mérida España, Universidad de Extremadura

[26] Hernández Arturo, Botello Salvador, Lizárraga Giovanni, Coello Carlos. (2002) IS-PAES: A Single and Multiple-Objective Optimization Method. Technical Report I-02-19-CC, Centro de Investigaciones en Matemáticas

[27] Botello S, Marroquin JL, Oñate and Horebeek J. "Solving Structural Optimization Problems with Genetic Algorithms and Simulated Annealing", *International Journal for Numerical Methods in Engineering* 45; 1069-1084 (1999).

[28] S. Rajeev and C.S. Krishamoorthy, "Genetic Algorithms–Based Methodologies for Design Optimization of Trusses", *Journal of Structural Engineering* 123 (3), 350–358 (1997)

[29] Ackley D. H., " An empirical study of bit vector function optimization" *Genetic algorithms and simulated annealing* D. Lawrence ed., Morgan Kauffmann Publishers, Los Altos Calif., 170–271.

[30] Venkayya VB. " Design of optimun structures". *Computers & Structures* 1; 265–309 (1971)

[31] Gellatly RA, Berke L. "Optimal structural Design". *AFFDL-TR-70-165* , (1971)

[32] Renwei X, Peng L. "Structural optimization based on second orden approximations of functions and dual theory". *Computer Methods in Applied Mechanics and Engineering* 65; 101–4, (1987)

[33] Schmit LA, Farshi B. "Some approximation concepts for structural synthesis ". *AIAA J* 12; 231-3 (1974).

[34] Xicheng W, Guixu M. " A parallel iterative algorithm for structural optimization ". *Computer Methods in Applied Mechanics and Engineering* 96; 25-32 (1992)

[35] Erbatur F, Hasancebi O, Tutuncu I, Kilic H "Optimal design of planar and space structures with genetic algorithms ". *Computer & Structures* 75; 209–224 (2000)