ISPAES: Evolutionary Multi-Objective Optimization with Constraint Handling

Arturo Hernández Aguirre, Salvador Botello Rionda, Giovanni Lizárraga
Center for Research in Mathematics, Department of Computer Science
Mineral de Valenciana, Guanajuato, 36240, México
artha,botello,giovanni@cimat.mx

Carlos 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 control diversity, but here this grid can grow and shrink dynamically until the constraints are met. We also propose a novel approach to remove unfeasible individuals from the population while keeping high population diversity. Several examples of the literature are used to show the potential of ISPAES.

1 Introduction

Evolutionary Algorithms (EAs) in general (i.e., genetic algorithms, evolution strategies and evolutionary programming) lack a mechanism able to bias efficiently the search towards the feasible region in constrained search spaces. Such a mechanism is highly desirable since most real-world problems have constraints which could be of any type (equality, inequality, linear and nonlinear). The success of 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. When using penalty functions, the amount of constraint violation is used to punish infeasible solution so that feasible solutions survive for reproduction. Despite the popularity of penalty functions, they have several drawbacks, one of them being the careful fine tuning of the penalty factors or degree of penalization of either constraint. Another approach to constraint handling is to treat the constraints as objective functions, and then solve the problem for all of them. IS-PAES follows the latter 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 for dominance testing. (thus only the real objective functions are used). 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:

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

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

where $\vec{F}$ is the vector of objective functions $\vec{F} = [f_1(\vec{x}), \ldots, f_k(\vec{x})]$, $\vec{x}$ 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 constraint 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

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. 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 having this limitation has to do with the focus of the search in traditional multiobjective optimization algorithms. Rather than focusing the effort on finding good "tradeoffs" (as in multiobjective optimization), we propose 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. Notice that "tradeoffs" are neglected, therefore, any constraint satisfied is as good as any other, and in fact, they are not used to test Pareto dominance once individuals are in the feasible region. Such is the nature of the algorithm proposed in this paper.

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 $\bar{v} = (f(\vec{x}), f_1(\vec{x}), \ldots, f_{k+m}(\vec{x}))$, where $f_1(\vec{x}), \ldots, f_k(\vec{x})$ 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).

4 Related work

We will now provide a brief discussion of the different approaches that have been proposed in the literature adopting the three main ideas previously indicated.

4.1 COMOGA

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). 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.

4.2 VEGA

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. 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.

4.3 MOGA

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 [16] .

4.4 NPGA

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. However, unlike the NPGA, Coello and Mezura's approach does not require niches (or fitnes sharing [18]) to maintain diversity in the population.

4.5 Pareto Set and Line Search

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:

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

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.

4.6 Pareto Ranking and Domain Knowledge

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. 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.

4.7 Pareto Dominance and Preselection

Jiménez et al. [25] proposed an algorithm that uses Pareto dominance inside a preselection scheme to solve several types of optimization problems (multiobjective, constraint satisfaction, global optimization, and goal programming problems). The approach redefines the problem as an unconstrained multiobjective optimization problem in which objectives are given priorities. Feasible solutions with a good objective function value are given the highest priority.

4.8 Pareto Ranking and Robust Optimization

Ray [22] explored an extension of his previous work on constraint-handling [21] 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.

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 on which objective function space is located using a coordinate system. Such a grid is the diversity maintenance mechanism of PAES and its 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 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, upon termination, the size of the search space being inspected will be the feasible region. It is important to indicate that the pruning of portions of the infeasible region of a problem has been proposed before by other researchers (e.g., [7], although in this case the search engine is not an EA). However, the combination of this shrinking concept with the usage of a secondary population and a multiobjective evolutionary algorithm is a novel proposal. The main algorithm of IS-PAES is shown next (a complete description of the algorithm and more results is 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

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 memory 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: $x_1 \square x_2$ 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 $\square$ c) **then** c=h
else if ($\exists a_p \in$ file $|h \square a_p$) **then**
remove(a_p); add(h)
if (h $\square$ c) **then** c = h;

5.1 Inverted "ownership"

IS-PAES handles the population *as part of* a grid location relationship, whereas PAES handles a grid location *contains* population relationship. In other words, PAES keeps a list of individuals on either grid location, but in IS-PAES either 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 every location of the grid in order to produce an unsorted list, there after the list is sorted. 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. 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 ISPAES 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). The goal is to pick feasible individual, a task ISPAES does by removing the unfeasible ones.

The function select(file) is shown next.

Pseudo-code of select(file)

m : number of constraints
mr : number of violated constrains
i : constraint index
$maxsize$: max size of file
$listsize$: 50% 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 the list is 50% of maxsize. Since individuals could be feasible, unfeasible or only partially feasible, the list is generated by discarding from file the worst elements by constraint. Notice that selec(file) does not use a greedy approach, for instance, searching for the best feasible individuals at once. 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 high the diversity of the population. If all individuals are feasible the algorithm does not remove any, and returns the same file. In any other case, the number of elements returned is 50% the initial size. 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).

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 $\in$ { 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:

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

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($r = g = 2$), and reduction rate of the hypervolume is 10% ($\alpha = 0.9$). We used 500 generation in each problem.

Experiment 1

The truss of the Figure 1 has to carry a load of 100 kN. The objectives are the minimization of the volume (designing for the minimum cost of fabrication) and the minimization of the maximum stresses 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: $max(\sigma_{AC}, \sigma_{BC}) \leq 10^5; 1 \leq y \leq 3$

The stresses are calculated with a close form:

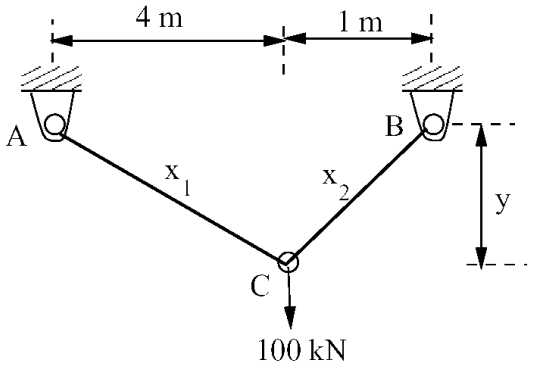

Figure 1. Truss of Experiment 1

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

In Figure 2 we show the real Pareto-optimal solution (calculated by enumeration) and the front reported by the ISPEAS 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].

Experiment 2

A compound gear train is designed to achieve a specific gear ratio between the driver and driven shafts (see the Figure 3).

The goal is to find the number of teeths in each of the four gears as to minimize the error between the obtained gear ratio and a required gear ratio of 1/6.931 and the maximum size of any of the four gears. There are four variables representing the number of teeths. We write the two-objective optimization problem:

$$\begin{aligned} Minimize f_1(x) &= \left[\frac{1}{6.931} - \frac{x_1 x_2}{x_3 x_4}\right] \\ Minimize f_2(x) &= \max(x_1, x_2, x_3, x_4) \end{aligned}$$

Subject to $12 \leq x_1, x_2, x_3, x_4 \leq 60, x_i \in \mathcal{Z}$

In Figure 4 we show the real Pareto-optimal solution (calculated by enumeration) and the result of the ISPAES algorithm. In this experiment, the ISPAES algorithm had a very good performance. We can see the results obtained by a single objective GAs (GeneAS-I and GeneAS-II)[12], by

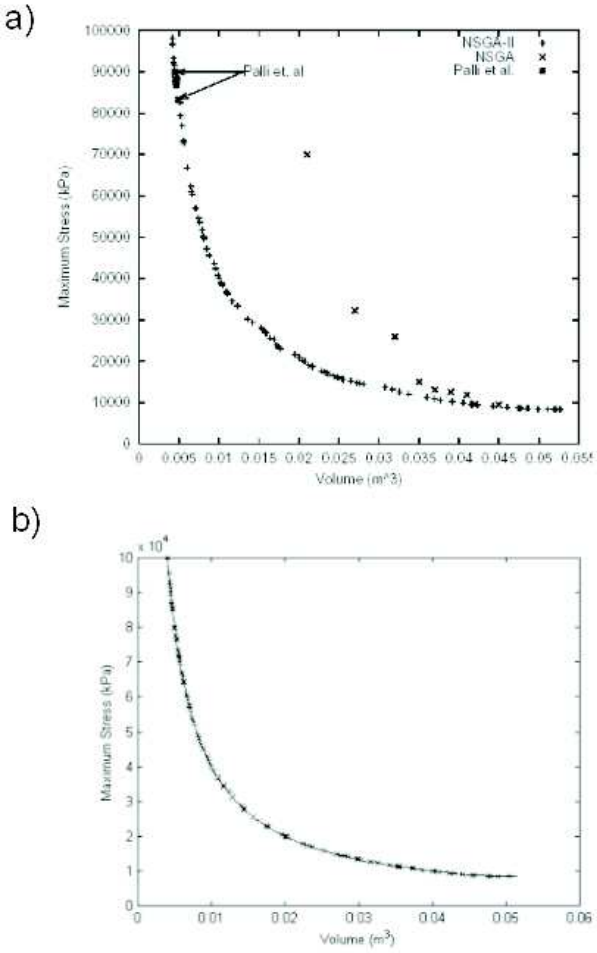

Figure 2. Pareto fronts of Experiment 1

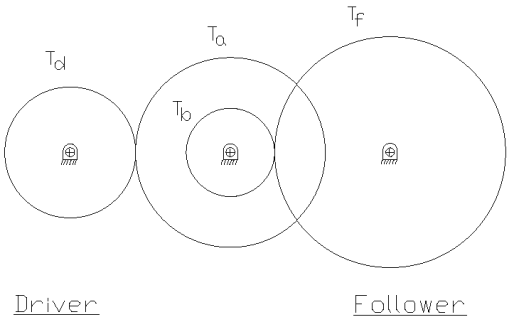

Figure 3. Compound gear of Experiment 2

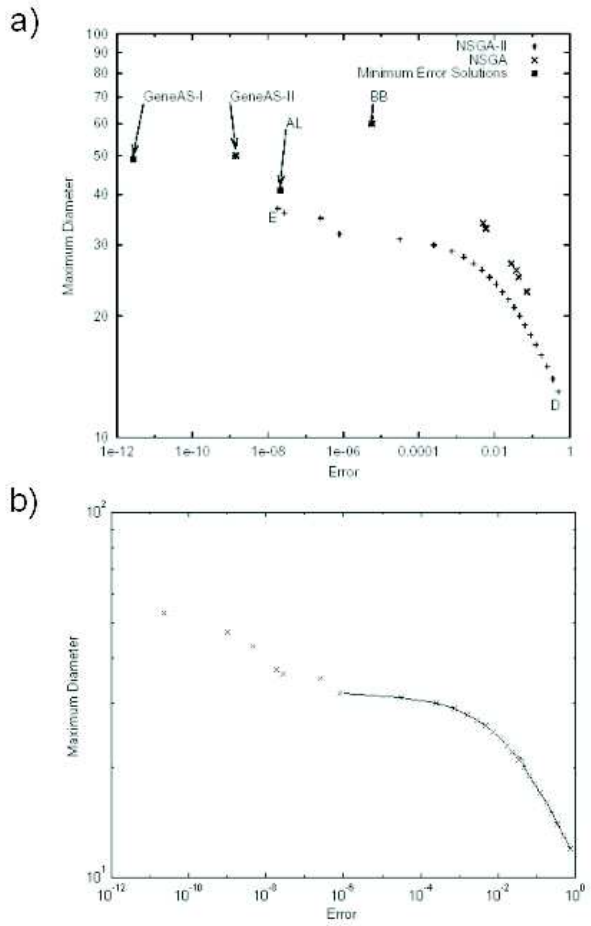

Figure 4. Pareto fronts of Experiment 2

the augmented Lagrangian (AL), and the branch-and-bound (BB) methods for the error minimization, NSGA[10] and NSGA-II [11][13]

7 Conclusions

We have proposed a new multiobjective optimization algorithm whose selection method is based on Pareto dominance concept. IS-PAES determines automatically the region where the optimum is located by bounding the area every time the unfeasible region is removed from the search space. Our method is more robust to scalability problems then the original PAES, as many experiments have demonstrated it [26]. At the same time, since IS-PAES treats the individuals on the grid in the described "inverted" mechanism, the computation performance and the use of computation resources is better than in PAES. The constraint handing technique reduces the dimensional complexity of the Pareto dominance, therefore, dominance is easier to identify. The experiments show good dispersion along the Pareto front, and the evolved front is over (or almost over) the real Pareto front, improving the results of other well know methods.

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.

References

[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