The Max-Min algorithm for maximizing the spread of Pareto fronts applied to Multiobjective Structural Optimization

Sergio Ivvan Valdez Peña, Salvador Botello Rionda, Arturo Hernández Aguirre, and *David Greiner Sánchez

Center for Research in Mathematics(CIMAT), Computer Science Department, A.P. 402, Guanajuato, Gto. CP 36000, México. e-mail: ivvan, botello, artha@cimat.mx
*Institute of Intelligent Systems & Numerical Applications in Engineering (IUSIANI)
University of Las Palmas de Gran Canaria., Spain. e-mail: dgreiner@iusiani.ulpgc.es

Keywords: Structural Optimization, EDAs, Constrained optimization, Multi-objective, Spreading

ABSTRACT

This paper introduces an algorithm for the maximization of the spreading factor of Pareto fronts. A set of synthetic functions is used to contrast the spreading of our approach against the one produced by NSGA-II and SPEA-2. A bi-objective structural optimization design problem with constraints is also solved with the proposed technique. The goal is to minimize weight and displacements in the structure, subject to three physical constraints: Von Misses Stress (subject to a maximum permissible), small holes, and the number of pieces used to build the structure. The finite element method is used to evaluate the potential solutions elaborated by the search algorithm in the discrete space of the structural problem. Since we approach bi-criterion optimization problems, a selection mechanism based on Pareto dominance determines the best potential solutions. However, the number of potential solutions in this kind of problems is large and the size of the Pareto set is limited to a target number. Thus, the goal of this paper is to introduce an algorithm that picks as many potential solutions as the target number while keeping the maximum spreading of the Pareto front.

Estimation Distribution Algorithm. The representation is discrete, thus every structural element is encoded by a binary variable which indicates whether an element is present or not. A number of probability vectors is computed from a population sample. The length of each vector is the number of structural elements of the potential solutions. Thus, each element gets a probability value of being present (or absent), and later used to generate the new population. Each probability vector is linked with a point on the Pareto front, so we have as many probability vectors as target points on the Pareto front. A detailed description of the EDA algorithm used is available in [2] but the important features are the following. During the first step random vectors populate the algorithm and the constraints and objective functions are evaluated. A selection based on Pareto dominance criteria is applied over constraints and objective functions producing a set of feasible and not feasible individuals. The feasible ones are chosen at once. If the target size of the Pareto set is not met then more elements are chosen from the infeasible set but this time Pareto dominance is computed only over the constraints. When the total number of non-dominated individuals is larger than the target number, the Max-Min algorithm (described next) picks as many as the target number of desired individuals which have maximum spreading over the Pareto set. Then, each individual is used to update a probability vector.

Spreading Algorithm. Most Multi-objective Evolutionary Algorithms (MOEAs) are based on Pareto dominance criteria to select the best candidate solutions. This criteria applied on **m** functions can not discriminate neither guide the search when **m** is relatively large (greater than 3). Furthermore, an infinite number of Pareto solutions can be found in continuous domains for **m** equal to 2. Because of these issues, the efficient application of Pareto dominance has been investigated. Many authors have proposed different manners to discriminate the candidate solutions that provide good spread over the Pareto front while keeping population diversity. For instance, PAES [6] uses a grid over function space to determine the less populated regions; NSGA-II [1] estimates a crowding measure by computing the distance to neighbors; SPEA2 [3] estimates crowding by finding clusters of population. The proposed algorithm adds points to a new Pareto front so the next point chosen is that one that maintains high uniform distribution, thus maximizing spread. The NSGA-II and SPEA2 crowding measures do not ensure good spreading because eliminating clusters or crowded areas do not mean to choose the best uniformly distributed individuals. The Max-Min algorithm is efficient since distances are computed only between the selected points and the remaining ones; also sorting is not necessary. The pseudo code of the *Max-Min* algorithm is presented next.

```
#S non-dominated solutions from a population.
#F(S) evaluted functions for the set S.
#k number of objective functions.
#D Selected set using the MaxMin distance measure.
#AS Archive Size (number of individuals which will be selected).

#Normalize the values of the objective functions
  normalize( F(S)  );
#Select the minimum (minimization case) values of each objective function independently
  Select_minimum( F(S) ,D);
#Initialize the distance measure d_i
  For i=0; i ∈ S \ D   {
       d_i =k;
       For l=0; l ∈ D    {
            if (d_i > sqrt((f_{1,i} − f_{1,l})^2 + (f_{2,i} − f_{2,l})^2 + ... + (f_{k,i} − f_{k,l})^2) )
               d_i = sqrt((f_{1,i} − f_{1,l})^2 + (f_{2,i} − f_{2,l})^2 + ... + (f_{k,i} − f_{k,l})^2)
       }
  }
#find_max returns the index of the maximum value in the vector d̂
  m= find_max(d̂)
#Insert the selected individual m by the distance measure in D
  D ⟵ x_m
#Selects the needed individuals to fill the fixed size Pareto Set.
  For j=k+1; j < AS    {
      For i=0; i ∈ S \ D    {
           if (d_i > sqrt((f_{1,i} − f_{1,m})^2 + (f_{2,i} − f_{2,m})^2 + ... + (f_{k,i} − f_{k,m})^2) )
              d_i = sqrt((f_{1,i} − f_{1,m})^2 + (f_{2,i} − f_{2,m})^2 + ... + (f_{k,i} − f_{k,m})^2)
      }
      m= find_max(d)
      D ⟵ x_m
  }
```

Experiments. The Max-Min algorithm was contrasted with the spreading mechanism of the NSGA-II and SPEA2 algorithm. The NSGA-II uses a distance measure based on the length of the sides of a hypercube delimited by the neighbors of an individual. SPEA2 uses a density function based in the K-neighbor distance. For comparison we use a 3-objective synthetic functions to generate a Pareto set. Define the variables as *x= random(0,1)*, *y= random(0,1)*; the Pareto front is defined as: $f_1= x$, $f_2= y$ and $f_3= (1-x^2)+(1-y^2)$. We can observe that every individual generated by these functions will be non-dominated. In order to compare the different approaches we used the spreading metric in Equation (1) (see [5]).

$$\nabla = \frac{\sum^{|F|} d_e + \sum_i^{|Q|} \mid d_i - \mu \mid}{\sum^{|F|} d_e + |Q|\mu} \quad (1)$$

$|F|$ is the number of objective functions, d_e is the distance between every extreme in the Pareto set we are measuring and the corresponding extreme in a Reference Pareto set, |Q| is the population size, d_i is the sum of normalized Euclidean distance from point *i-th* to their neighbors in every objective function (point *(i+1)-th* sorting every objective function), and μ is the mean of the distances of the Pareto set we are measuring. The spreading reported by each approach is shown in Table 1. Population size is the whole Pareto set generated by using the functions discussed before. For 30 independent runs the target archive of 50 points was computed, discriminating points according to the NSGA-II, SPEA2 and *Max-Min* algorithm. The mean and variance are also shown.

Population Size	mean SPEA2	mean NSGA-II	mean MAXMIN	var SPEA2	var NSGA-II	var MAXMIN
100	0.360485	0.350160	0.276169	0.001826	0.002133	0.000856
250	0.445818	0.413065	0.264809	0.002710	0.004110	0.000525
500	0.530040	0.470547	0.253144	0.005588	0.003189	0.000715
1000	0.560867	0.591298	0.257937	0.005586	0.002147	0.000606

Table 1. Spreading measures, for the different approaches, filling a fixed size archive of 50 individuals.

We can observe (graphically) the results of the different approaches in Figure 1.

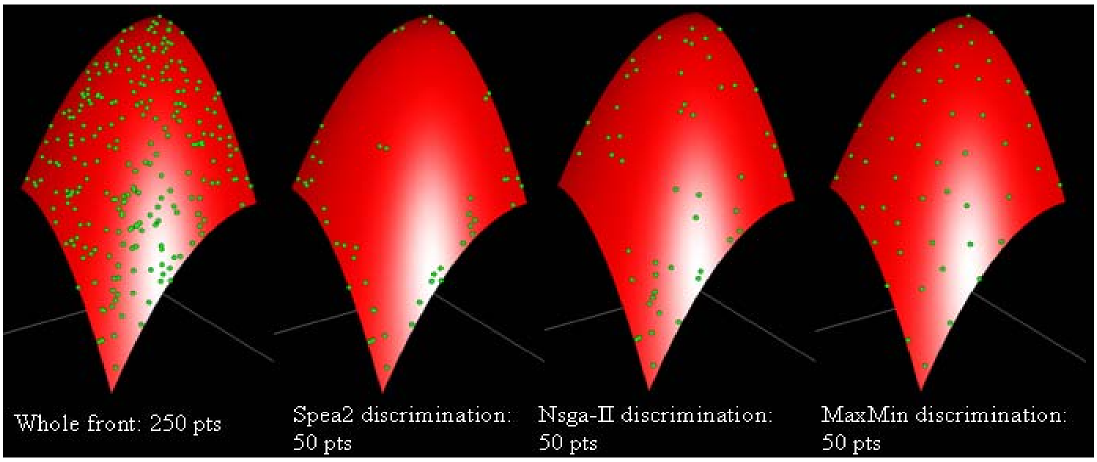
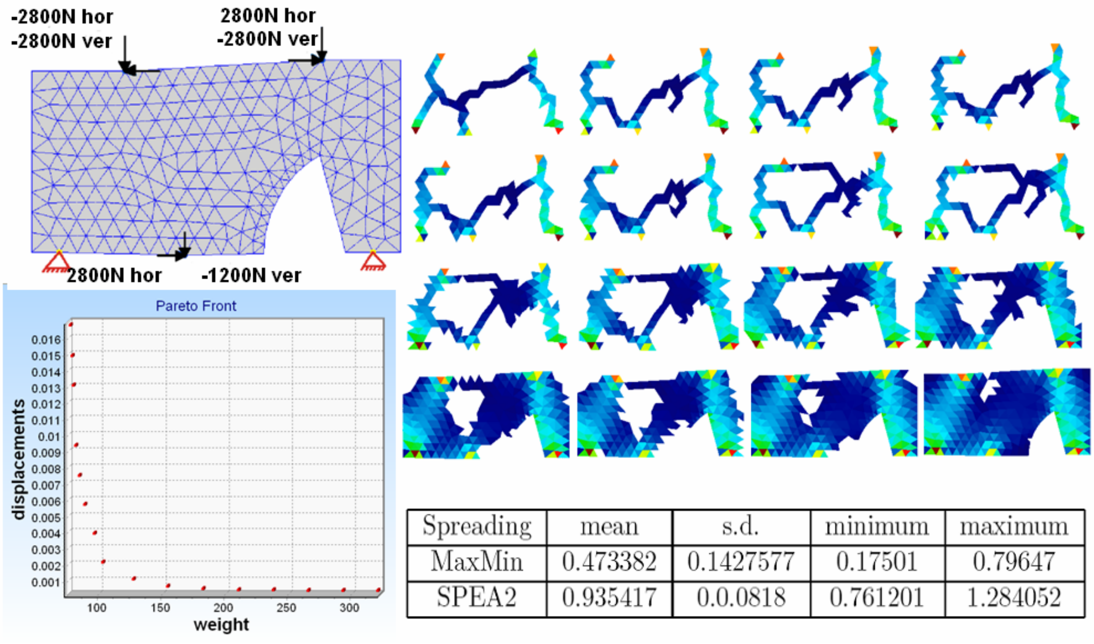


Figure 1. Comparison between different discrimination mechanism. 50 individuals were selected from 250 non-dominated solutions (see row 2 in table 1).

Figure 2 shows the results of another design problem. The shape optimization algorithm is used to design a bicycle frame by using 16 probability vectors. For 30 independent runs the spreading metric was calculated giving the results presented in Table 2. For this experiment the SPEA2 discrimination mechanism and the Max-Min algorithm were applied. As we can observe in Figure 2, the Pareto front resulting from the Max-Min algorithm presents a nice uniform spreading. The material properties are: Young modulus =70e9 Pa, Poisson modulus = 0.2, thickness =0.005 m, Maximum Von Misses Stress = 100e6 Pa.

Conclusions. The new approach to multi-objective shape optimization gives a set of compromise solutions which are representative of the solution space. The proposed algorithm is compared with the NSGA-II and SPEA2 mechanism that provide spread solutions over the Pareto front. In all cases the Max-Min provides better spreading over the Pareto front. By using the Max-Min algorithm we found good candidate designs as shown in Figure 2. In this case the design problems has a large amount of variables (in the hundreds), and therefore a huge search space (about 2^n possible solutions, where n is the number of finite elements).



Spreading	mean	s.d.	minimum	maximum
MaxMin	0.473382	0.1427577	0.17501	0.79647
SPEA2	0.935417	0.0.0818	0.761201	1.284052

Figure 2.Application example (bicycle frame), and comparison using the SPEA2 spreading mechanism and the proposed Max-Min algorithm.

REFERENCES

[1] K. Deb, S. Agrawal, A Pratab, and T. Meyarivan. "A fast elitist non-dominated sorting genetic algorithm for multi-objetctive optimization: NSGA-II", *Kangal report 200001.*

[2] S. I. Valdez, S. Botello, and A. Hernández, "Multiobjetctive Shape Optimization with constraints based on estimation distribution algorithms and correlated information", *GECCO'05, Washington DC, USA, 2005*

[3] E. Ziztler, M. Laumanns, and L. Thiele. "Spea2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization.", *Evolutionary methods for design, optimisation and control, CIMNE, Barcelona, Spain, 2002.*

[4] S. Baluja. "Population Based Incremental Learning: A method for integrating genetic search based function optimization and competitive learning", *CMU-CS-94-163, School of Computer Science, Carnegie Mellon University, 1996.*

[5] K. Deb, "Multi-objective Optimization using Evolutionary Algorithms", *Wiley Ed., pag. 328-330.*

[6] Knowles J. and Corne D., "The Pareto archived evolution strategy: A new baseline algorithm for multiobjective optimisation." *Proceedings of the 1999 Congress on Evolutionary Computation, Piscatway: New Jersey: IEEE Service Center, 98-105*