# Multi-Objective Evolutionary Algorithms for Structural Optimization

C.A. Coello Coello[1], G. Toscano Pulido[1], A. Hernández Aguirre[2]

[1]*CINVESTAV-IPN, Sección de Computación*

*Av. IPN No. 2508, Col. Sn. Pedro Zacatenco*

*México, D. F. 07300, MEXICO*

`ccoello@cs.cinvestav.mx`

[2]*CIMAT, Department of Computer Science*

*Guanajuato, Gto. 36240, MEXICO*

`artha@cimat.mx`

**Abstract**

In this paper, we propose a multi-objective evolutionary algorithm (MOEA) for structural optimization. The proposed approach emphasizes efficiency and has been found to be competitive with respect to other MOEAs in current use. One example is used to illustrate the way in which the approach works.

# 1 Introduction

Evolutionary algorithms have become an increasingly popular design and optimization tool in the last few years, with a constantly growing development of new algorithms and applications [1]. Despite this considerably large volume of research, new areas remain to be explored with sufficient depth. One of them is the use of evolutionary algorithms to solve multiobjective optimization problems. In nature, most problems are multiobjective, but we normally tend to restate them as single-objective optimization in problems by transforming all the objectives, but one into constraints. However, such an assumption keeps us from producing designs that represent better trade-offs among the original objectives of the problem and therefore limit the quality of our designs.

Evolutionary algorithms seem also particularly desirable for solving multiobjective optimization problems because they deal simultaneously with a set of possible solutions (the so-called population) which allows us to find several members of the Pareto optimal set in a single run of the algorithm, instead of having to perform a series of separate runs as in the case of the traditional mathematical programming techniques. Additionally, evolutionary algorithms are less susceptible to the shape or continuity of the Pareto front (e.g., they can easily deal with discontinuous and concave Pareto fronts), whereas these two issues are a real concern for mathematical programming techniques [3].

In this paper, we propose an evolutionary multiobjective optimization algorithm that has been found to be efficient and relatively easy to implement. We also present one example to illustrate the way in which our approach works.

## 2 Basic Concepts

We want to solve multiobjective optimization problems (MOPs) of the form:

$$\text{minimize } [f_1(\vec{x}), f_2(\vec{x}), \ldots, f_k(\vec{x})] \tag{1}$$

subject to the $m$ inequality constraints:

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

and the $p$ equality constraints:

$$h_i(\vec{x}) = 0 \quad i = 1, 2, \ldots, p \tag{3}$$

where $k$ is the number of objective functions $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$. We call $\vec{x} = [x_1, x_2, \ldots, x_n]^T$ the vector of decision variables. We wish to determine from among the set $\mathcal{F}$ of all numbers which satisfy (2) and (3) the particular set $x_1^*, x_2^*, \ldots, x_n^*$ which yields the optimum values of all the objective functions.

## 2.1 Pareto optimality

It is rarely the case that there is a single point that simultaneously optimizes all the objective functions. Therefore, we normally look for “trade-offs”, rather than single solutions when dealing with multiobjective optimization problems. The notion of “optimality” is therefore, different in this case. The most commonly adopted notion of optimality is called Pareto optimality. We say that a vector of decision variables $\vec{x}^* \in \mathcal{F}$ is *Pareto optimal* if there does not exist another $\vec{x} \in \mathcal{F}$ such that $f_i(\vec{x}) \leq f_i(\vec{x}^*)$ for all $i = 1, \ldots, k$ and $f_j(\vec{x}) < f_j(\vec{x}^*)$ for at least one $j$.

Unfortunately, the concept of Pareto optimality almost always gives not a single solution, but rather a set of solutions called the *Pareto optimal set*. The vectors $\vec{x}^*$ correspoding to the solutions included in the Pareto optimal set are called *non-*

*dominated*. The plot of the objective functions whose nondominated vectors are in the Pareto optimal set is called the *Pareto front*.

## 3 Description of our approach

The term micro-genetic algorithm (micro-GA) refers to a small-population genetic algorithm with reinitialization where computational efficiency is emphasized. The approach proposed in this paper is the first attempt to use a micro-GA for multiobjective optimization. The proposed technique incorporates concepts that represent the state-of-the-art in evolutionary multiobjective optimization, namely elitism (implemented using an external or secondary memory to retain nondominated individuals found along the evolutionary process) and Pareto ranking (i.e., nondominated individuals are favored by the selection process). Despite the various efforts to use evolutionary multiobjective optimization algorithms in structural optimization problems (see [3] for a survey), few of those efforts have used Pareto ranking. Aggregating functions are still very popular in structural optimization despite their well-known limitations (e.g., they cannot generate non-convex portions of the Pareto front). The main emphasis of the work reported in this paper is precisely to show what can be achieved by using a "true" evolutionary multiobjective optimization algorithm (i.e., an algorithm that uses Pareto ranking and elitism).

The way in which our technique works is illustrated in Figure 1. First, a random population is generated. This random population feeds the population memory, which is divided in two parts: a replaceable and a non-replaceable portion. The non-replaceable portion of the population memory will never change during the entire run and is meant to provide the required diversity for the algorithm. In contrast, the replaceable portion will experience changes after each cycle of the micro-GA. The population of the micro-GA at the beginning of each of its cycles is taken (with a certain probability) from both portions of the population memory so that we can have a mixture of randomly generated individuals (non-replaceable portion) and evolved individuals (replaceable portion). During each cycle, the micro-GA undergoes conventional genetic operators (binary representation is used in our implementation): tournament selection, two-point crossover, uniform mutation, and elitism (only one nondominated vector is arbitrarily selected at each generation and copied intact to the following one). After the micro-GA finishes one cycle, we choose two nondominated vectors from the final population and compare them with the contents of the external memory (this memory is initially empty). If either of them (or both) remains as nondominated after comparing it against the vectors in this external memory, then they are included there (i.e., in the external memory). This is our historical archive of nondominated vectors.

To keep diversity along the Pareto front, we use a modified version of the adaptive grid proposed by Knowles & Corne [5] (see Figure 2). The idea is that once the archive that stores nondominated solutions has reached its limit, we divide the objective search space that this archive covers, assigning a set of coordinates to each solution. Then, each newly generated nondominated solution will be accepted only if the geographical location to where the individual belongs is less populated than the most crowded location.

Note that although our approach requires several parameters that have to be defined by the user, we have suggested in previous work default values that can be adopted when nothing is known about the problem to be solved (see [2] for details).

## 4 An Example

We have compared our micro-GA against some of the most competitive multi-objective evolutionary algorithms in current use (i.e., the NSGA-II [4] and PAES [5]), obtaining very good results (see [2] for details). Due to space limitations, in this paper we will only present one engineering design example, and results will be compared with respect to the true Pareto front of the problem (generated by enumeration). Obviously, in more complex problems the true Pareto front cannot be found by enumeration. However, we adopted this methodology just to illustrate that our technique is able to approximate the true Pareto front of a problem with a very good accuracy in a short period of time.

## 4.1 Design of a welded beam

This problem was originally proposed by Reklaitis et al. [6] and was transformed into a bi-objective problem by Wu [7]. A beam A is to be welded to a rigid support member B. The welded beam is to consist of 1010 steel is to support a force of 6000lb. The two design objectives are to simultaneously minimize the beam cost and minimize the end deflection. There are constraints on shear stress, normal stress and buckling load. The independent design variables in this case, which are all continuous, are the dimensions and (see Figure 3). The length $L$ is assumed to be specified at 14 inches. The parameters used by our micro-GA are the following: size of the external memory = 100, nominal convergence = 2 iterations, size of the population memory = 50, population size (of the micro-GA) = 4, number of subdivisions of the adaptive grid = 25, crossover rate = 0.8, mutation rate = 1.0/$L$ ($L$ = length of the chromosome), replacement cycle = 50, percentage of non-replaceable memory = 0.3, maximum number of generations = 3500. The true Pareto front of the problem (generated by enumeration) compared to the solution generated by our micro-GA are shown in Figure 4. The average running time of the micro-GA for this problem was of 1.475 seconds (on a PC with an Athlon XP 1500+ processor, running at 1.3 GHz and 256Mb of RAM). As can bee seen in Figure 4, our micro-GA produces a very good approximation of the Pareto front of this problem.

## 5 Conclusions and Future Work

We have presented a multiobjective evolutionary algorithm that has been successfully applied to solve structural optimization problems. The algorithm is conceptually simple and very efficient.[1] The results obtained show that our algorithm can produce reasonably good approximations of the Pareto front of a problem at an affordable computational cost. Our main goal is to encourage engineers to get involved in evolutionary multiobjective optimization and to use this or other algorithms currently available in the specialized literature (see [3]). As part of

[1] A public-domain version of the algorithm is available at: `http://delta.cs.cinvestav.mx/ ccoello/EMOO/EMOOsoftware.html`

our future work, we plan to introduce on-line adaptation of the parameters of our micro-GA so that no parameters fine-tuning is required.

## Acknowledgements

The first author acknowledges support from CONACyT project 34201-A. The second author acknowledges support from CONACyT through a scholarship to pursue graduate studies at CINVESTAV-IPN. The third author acknowledges partial support from CONCyTEG project No. 01-02-202-111 and CONACyT No. I-39324-A.

## References

[1] Thomas Bäck, David B. Fogel, and Zbigniew Michalewicz, editors. *Handbook of Evolutionary Computation*. Institute of Physics Publishing and Oxford University Press, 1997.

[2] Carlos A. Coello Coello and Gregorio Toscano Pulido. Multiobjective Optimization using a Micro-Genetic Algorithm. In Lee Spector, Erik D. Goodman, Annie Wu, W.B. Langdon, Hans-Michael Voigt, Mitsuo Gen, Sandip Sen, Marco Dorigo, Shahram Pezeshk, Max H. Garzon, and Edmund Burke,

editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, pages 274–282, San Francisco, California, 2001. Morgan Kaufmann Publishers.

[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] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002.

[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] G.V. Reklaitis, A. Ravindran, and K.M. Ragsdell. *Engineering Optimization Methods and Applications*. Wiley-Interscience, 1985.

[7] Jin Wu. *Quality Assisted Multiobjective and Multidisciplinary Genetic Algorithms*. PhD thesis, Department of Mechanical Engineering, University of Maryland at College Park, College Park, Maryland, 2001.

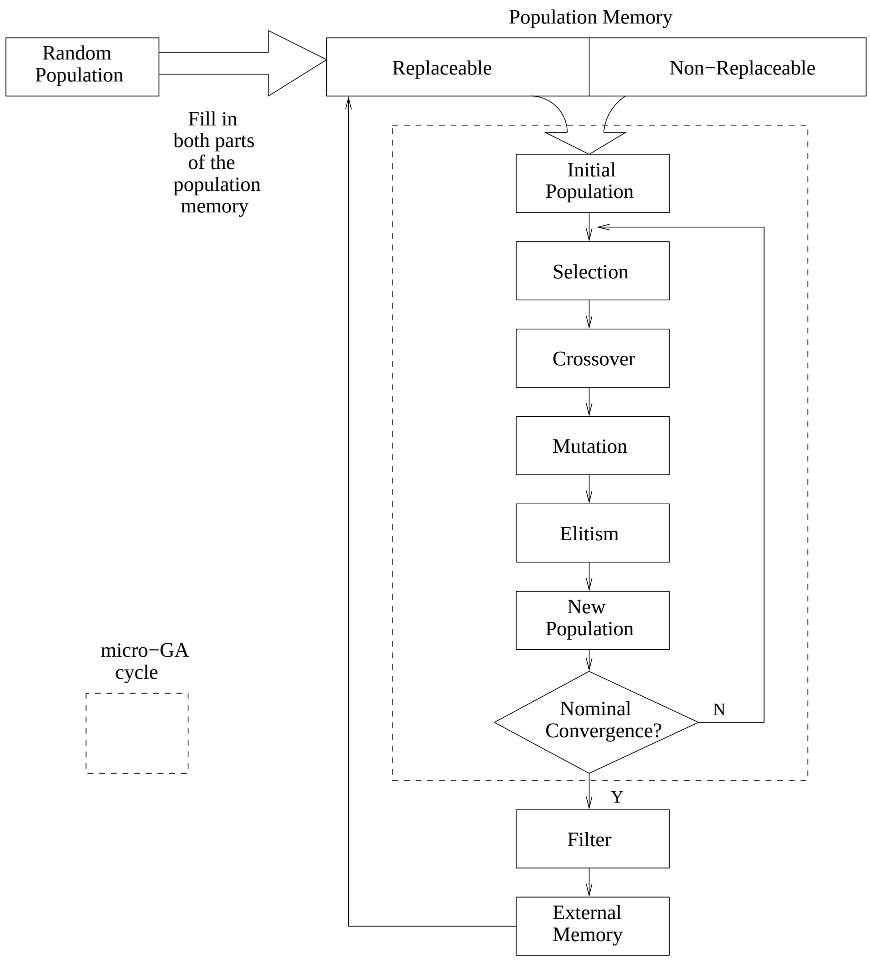


Figure 1: Diagram that illustrates the way in which our micro-GA works.

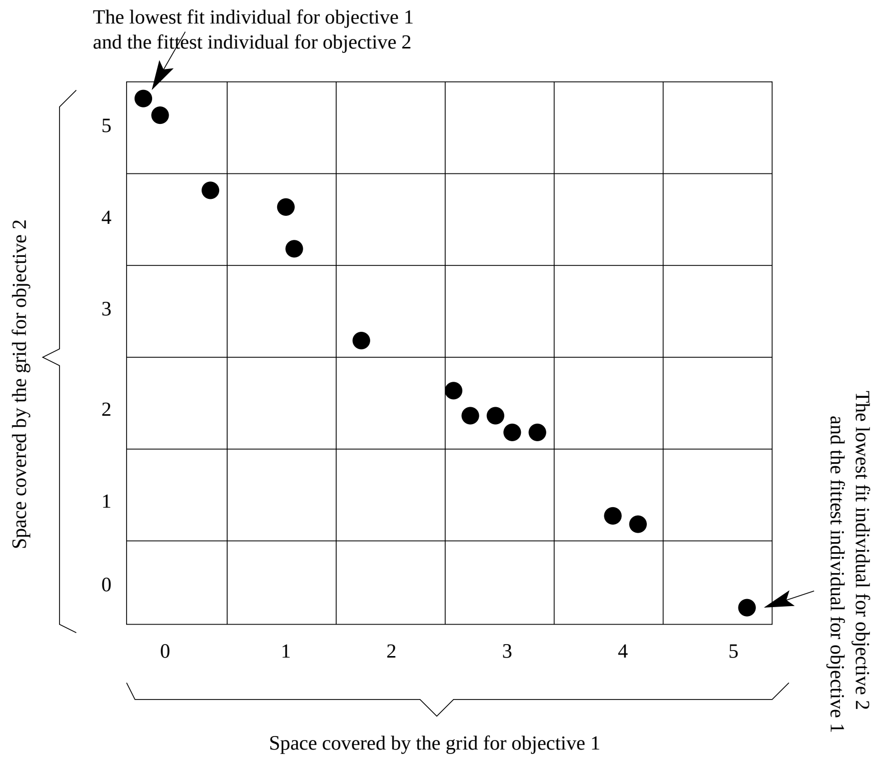


Figure 2: The adaptive grid used to hadle the external memory of the micro-GA.

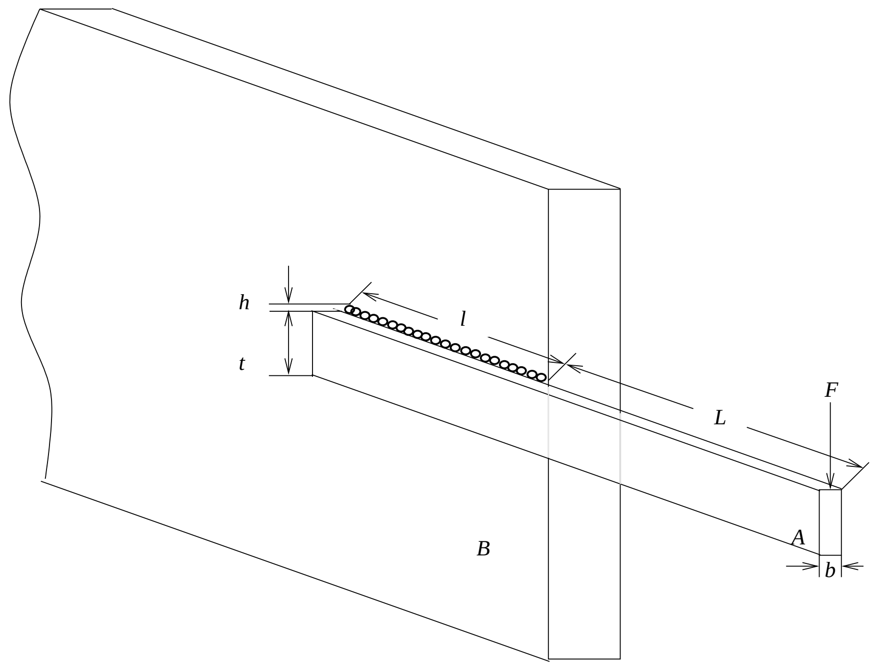


Figure 3: Welded beam used for the second example.

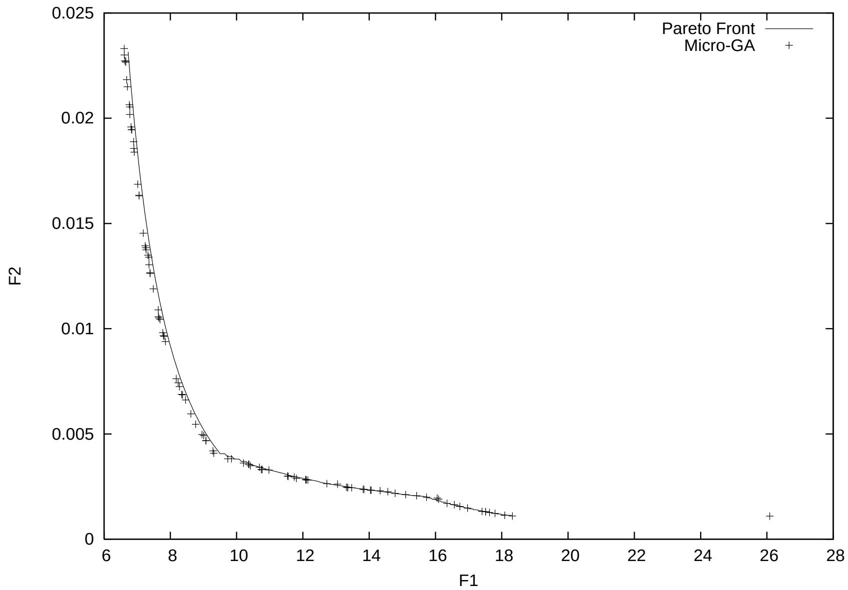


Figure 4: Pareto front of the example. The true Pareto front is shown as a continuous line and the solutions found by our micro-GA are displayed as crosses.