# Using the Min-Max Method to Solve Multiobjective Optimization Problems with Genetic Algorithms

Carlos A. Coello Coello *
ccoello@soc.plym.ac.uk
Engineering Design Centre
University of Plymouth
Plymouth, Devon PL4 8AA, UK

**Abstract.** In this paper, a new multiobjective optimization technique based on the genetic algorithm (GA) is introduced. This method is based in the concept of min-max optimum, taken from the Operations Research literature, and can produce the Pareto set and the best trade-off among the objectives. The results produced by this approach are compared to those produced with other mathematical programming techniques and GA-based approaches using a multiobjective optimization tool called MOSES (Multiobjective Optimization of Systems in the Engineering Sciences). The importance of representation is hinted in the example used, since it can be seen that reducing the chromosomic length of an individual tends to produce better results in the optimization process, even if it's at the expense of a higher cardinality alphabet.

## 1 Introduction

Engineering optimization has been a very fertile area of research in the last few years, but the normal trend has been to deal with a single objective at a time, and use ideal and unrealistic problems, rather than real-world applications. Assuming only one objective is generally unrealistic for engineering optimization problems, since most real-world problems have several (possibly conflicting) objectives. The common practice, therefore, has been to let the designer to make decisions based on his/her experience, instead of using some well-defined optimality criterion.

Over the years, the Operations Research community has produced more than 20 mathematical programming techniques to deal with multiple objec-

*Part of this work was developed by the author while visiting LANIA (Laboratorio Nacional de Informática Avanzada) in Xalapa, Veracruz, México tives. However, the main focus of these approaches is to produce a *single* trade-off based on some notion of optimality, rather than producing several possible alternatives from which the designer may choose. More recently, the genetic algorithm (GA), an artificial intelligence search technique based on the mechanics of natural selection, has been found to be effective on some scalar optimization problems. In order to extend the GA to deal with multiple objectives, the structure of the GA has been modified to handle a vector fitness function.

This paper will review some of the previous work in multiobjective optimization using GAs, and a new approach, proposed by the author, will be introduced. Also, MOSES (Multiobjective Optimization of Systems in the Engineering Sciences), a system developed as a testbed for multiobjective optimization techniques by the author, will be briefly described together with an example of its use. The new approach, based on the notion of min-max optimum, is able to generate the Pareto set and better trade-offs than any of the other techniques included in MOSES. The importance of using alphabets of cardinality higher than two will be emphasized, and the results found with this alternative representation will be shown to be better than those produced using a traditional binary representation, both for single and multiobjective optimization.

## 1.1 Statement of the Problem

Multiobjective optimization (also called multicriteria optimization, multiperformance or vector optimization) can be defined as the problem of finding [13]:

> a vector of decision variables which satisfies constraints and optimizes a vector function whose elements represent the objective functions. These functions form a mathematical description of performance criteria which are usually in conflict with each other. Hence, the term "optimize" means finding such a solution which would give the values of all the objective functions acceptable to the designer.

Formally, we can state it as follows:

Find the vector $\bar{x}^* = [x_1^*, x_2^*, \ldots, x_n^*]^T$ which will satisfy the $m$ inequality constraints:

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

the $p$ equality constraints

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

and optimize the vector function

$$\bar{f}(\bar{x}) = [f_1(\bar{x}), f_2(\bar{x}), \ldots, f_k(\bar{x})]^T \tag{3}$$

where $\bar{x} = [x_1, x_2, \ldots, x_n]^T$ is the vector of decision variables.

## 1.2 Min-Max Optimum

The idea of stating the *min-max optimum* and applying it to multiobjective optimization problems, was taken from game theory, which deals with solving conflicting situations. The min-max approach to a linear model was proposed by Jutler and Solich and was been further developed by Osyczka [11], Rao [14] and Tseng & Lu [18].

The min-max optimum compares relative deviations from the separately attainable minima. Consider the $i$th objective function for which the relative deviation can be calculated from

$$z_i^{'}(\bar{x}) = \frac{|f_i(\bar{x}) - f_i^0|}{|f_i^0|} \tag{4}$$

or from

$$z_i^{''}(\bar{x}) = \frac{|f_i(\bar{x}) - f_i^0)|}{|f_i(\bar{x})|} \tag{5}$$

It should be clear that for (4) and (5) we have to assume that for every $i \in I$ and for every $\bar{x} \in X$, $f_i(\bar{x}) \neq 0$.

If all the objective functions are going to be minimized, then equation (4) defines function relative increments, whereas if all of them are going to be maximized, it defines relative decrements. Equation (5) works conversely.

# 2 Multiobjective Optimization Using GAs

Some of the most important GA-based multiobjective optimization techniques will be briefly explained in this section.

# 3 VEGA

David Schaffer [16] extended Grefenstette's GENESIS program [8] to include multiple objective functions. Schaffer's approach was to use an extension of the Simple Genetic Algorithm (SGA) that he called the Vector Evaluated Genetic Algorithm (VEGA), and that differed of the first only in the way in which selection was performed. This operator was modified so that at each generation a number of sub-populations was generated by performing proportional selection according to each objective function in turn. Thus, for a problem with $k$ objectives, $k$ sub-populations of size $N/k$ each would be generated, assuming a total population size of $N$. These sub-populations would be shuffled together to obtain a new population of size $N$, on which the GA would apply the crossover and mutation operators in the usual way. Schaffer realized that the solutions generated by his system were non-inferior in a local sense, because their non-inferiority is limited to the current population, and while a locally dominated individual is also globally dominated, the converse is not necessarily true [16].

# 4 Lexicographic Ordering

The basic idea of this technique is that the designer ranks the objectives in order of importance. The optimum solution is then found by minimizing the objective functions, starting with the most important one and proceeding according to the order of importance of the objectives [15]. Fourman [6] suggested a selection scheme based on lexicographic ordering. In a first version of his algorithm, objectives were assigned different priorities by the user and each pair of individuals were compared according to the objective with the highest priority. If this resulted in a tie, the objective with the second highest priority was used, and so on. A second version of this algorithm, reported to work surprisingly well, consisted of randomly selecting the objective to be used in each comparison. As in VEGA, this corresponds to averaging fitness across fitness components, each component being weighted by the probability of each objective being chosen to decide each tournament [5]. However, the use of pairwise comparisons makes an important difference with respect to VEGA, since in this case scale information is ignored. Therefore, the population may be able to see as convex a concave trade-off surface, depending on its current distribution, and on the problem itself.

# 5 Weighted Sum : Hajela's Method

Hajela and Lin [9] included the weights of each objective in the chromosome, and promoted their diversity in the population through fitness sharing. Their goal was to be able to simultaneously generate a family of Pareto optimal designs corresponding to different weighting coefficients in a single run of the GA. Besides using sharing, Hajela and Lin used a vector evaluated approach based on VEGA to achieve their goal. They proposed the use of a utility function of the form:

$$\bar{U} = \sum_{i=1}^{l} W_i \frac{F_i}{F_i^*} \tag{6}$$

where $F_i^*$ are the scaling parameters for the objective criterion, $l$ is the number of objective functions, and $W_i$ are the weighting factors for each objective function $F_i$. In MOSES's implementation, a min-max approach was used to determine the utility function, so that the scaling factor was the ideal vector.

Hajela's approach also uses a sharing function of the form:

$$\phi(d_{ij}) = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma_{sh}}\right)^{\alpha}, & d_{ij} < \sigma_{sh} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

where $\alpha = 1$ for this work, $d_{ij}$ is a metric indicative of the distance between designs $i$ and $j$, and $\sigma_{sh}$ is the sharing parameter, which is typically chosen between 0.01 and 0.1. The fitness of a design $i$ is then modified as:

$$f_{s_i} = \frac{f_i}{\sum_{j=1}^{M} \phi(d_{ij})} \tag{8}$$

where $M$ is the number of designs located in vicinity of the $i$-th design.

Hajela incorporates weight combinations into the chromosomic string, and under his representation, a single number represents not the weight itself, but a combination of them. For example, the number 4 (under floating point representation) could represent the vector $X_w = (0.4, 0.6)$ for a problem with two objective functions. Then, sharing is done on the weights. Finally, a mating restriction mechanism was imposed, to avoid members within a radius $\sigma_{mat}$ to cross.

# 6 MOGA

Fonseca and Fleming [4] have proposed a scheme in which the rank of a certain individual corresponds to the number of chromosomes in the current population by which it is dominated. Consider, for example, an individual $x_i$ at generation $t$, which is dominated by $p_i^{(t)}$ individuals in the current generation. Its current position in the individuals' rank can be given by [4]:

$$rank(x_i, t) = 1 + p_i^{(t)} \tag{9}$$

All non-dominated individuals are assigned rank 1, while dominated ones are penalized according to the population density of the corresponding region of the trade-off surface. See Fonseca and Fleming [4] for details.

# 7 NSGA

The Non-dominated Sorting Genetic Algorithm (NSGA) was proposed by Srinivas and Deb [17], and is based on several layers of classifications of the individuals. Before the selection is performed, the population is ranked on the basis of nondomination: all nondominated individuals are classified into one category (with a dummy fitness value, which is proportional to the population size, to provide an equal reproductive potential for these individuals). To maintain the diversity of the population, these classified individuals are shared with their dummy fitness values. Then this group of classified individuals is ignored and another layer of nondominated individuals is considered. The process continues until all individuals in the population are classified. A stochastic remainder proportionate selection was used for this approach. Since individuals in the first front have the maximum fitness value, they always get more copies than the rest of the population. This allows to search for nondominated regions, and results in quick convergence of the population toward such regions. Sharing, by its part, helps to distribute it over this region.

# 8 NPGA

Horn and Nafpliotis [10] proposed a tournament selection scheme based on Pareto dominance. Instead of limiting the comparison to two individuals, a number of other individuals in the population was used to help determine dominance. When both competitors were either dominated or non-dominated (i.e., there was a tie), the result of the tournament was decided through fitness sharing [7]. The pseudocode for Pareto domination tournaments assuming that all of the objectives are to be maximized can be found in Horn and Nafpliotis [10].

# 9 An Approach Using a Min-Max Strategy

The idea of this approach is to generate the individuals in such a way that they all constitute feasible solutions. This can be ensured by checking that none of the constraints is violated by the solution vector encoded by the corresponding chromosome, and by designing special operators. Then the user has to provide a vector of weights, which are used to spawn as many processes as weight combinations are provided (normally this number will be reasonably small). Each process is really a separate GA in which the given weight combination is used in conjunction with a min-max approach to generate a single solution. After the $n$ processes are terminated ($n$=number of weight combinations provided), a final file is generated containing the Pareto set, which is formed by picking up the best solution from each of the processes spawned in the previous step. Since this approach requires knowing the ideal vector, the user is given the opportunity to provide such values directly (in case he/she knows them) or to use another GA to generate it.

# 10 Example

To illustrate the use of MOSES and the efficiency of the new technique proposed, one engineering design example were selected from the literature [3]. Since it is generally intractable to obtain an analytical representation of the Pareto front, it is usually very difficult to measure the performance of a multiobjective optimization technique. For the purposes of this paper the results were compared only in terms of the best trade-offs that could be achieved. For that sake, the following expression was used

$$L_p(f) = \sum_{i=1}^{k} w_i \left| \frac{f_i^0 - f_i(x)}{\rho_i} \right| \tag{10}$$

where $k$ is the number of objectives, $\rho_i = f_i^0$,or $f_i(x)$, depending on which gives the maximum value for $L_p(f)$, and $w_i$ refers to the weight assigned to each objective (if not known, equal weights are assigned to all the objectives). A sketch of the Pareto front produced by each technique can actually be obtained

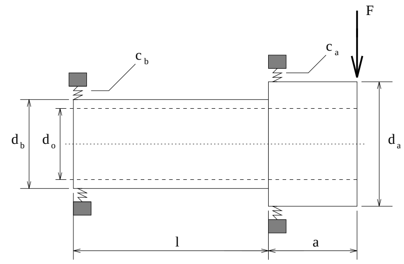


Figure 1: **Fig. 1** Sketch of the machine tool spindle used for the example.

with MOSES, but due to space limitations such graphs won't be included in this paper.

## 10.1 Design of a Machine Tool Spindle

Consider the problem of a preliminary design of a machine tool spindle as presented in Figure 1 (taken from Eschenauer et al. [3]). The formulation of the multiobjective optimization problem is to minimize $f_1(x)$ and $f_2(x)$ as defined below [3].

$$f_1(x) = \frac{\pi}{4}\left[a(d_a^2 - d_o^2) + l(d_b^2 - d_o^2)\right] \tag{11}$$

$$f_2(x) = \frac{Fa^3}{3EI_a}\left(1 + \frac{l}{a}\frac{I_a}{I_b}\right) + \frac{F}{c_a}\left[\left(1 + \frac{a}{l}\right)^2 + \frac{c_a a^2}{c_b l^2}\right] \tag{12}$$

$$I_a = 0.049(d_a^4 - d_o^4), \quad I_b = 0.049(d_b^4 - d_o^4) \tag{13}$$

$$c_a = 35400|\delta_{ra}|^{\frac{1}{9}} d_a^{\frac{10}{9}}, \quad c_b = 35400|\delta_{rb}|^{\frac{1}{9}} d_b^{\frac{10}{9}} \tag{14}$$

$$\begin{array}{c} g_1(x) = l - l_g \leq 0 \\ g_2(x) = l_k - l \leq 0 \\ g_3(x) = d_{a1} - d_a \leq 0 \end{array} \tag{15}$$

$$\begin{array}{c} g_4(x) = d_a - d_{a2} \leq 0 \\ g_5(x) = d_{b1} - d_b \leq 0 \\ g_6(x) = d_b - d_{b2} \leq 0 \\ g_7(x) = d_{om} - d_o \leq 0 \end{array} \tag{16}$$

$$\begin{array}{c} g_8(x) = p_1 d_o - d_b \leq 0 \\ g_9(x) = p_2 d_b - d_a \leq 0 \end{array} \tag{17}$$

$$g_{10}(x) = |\Delta_a + (\Delta_a - \Delta_b)\frac{a}{l}| - \Delta \leq 0 \tag{18}$$

For this example, it is assumed that $d_a$ must be chosen from the set $X_3 = \{80, 85, 90, 95\}$, and $d_b$ from the set $X_4 = \{75, 80, 85, 90\}$. Additionally, the following constant parameters are assumed:

| Method | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $f_1$ | $f_2$ |
|---|---|---|---|---|---|---|
| Monte Carlo 1 | 59.08 | 189.17 | 90 | 75 | **606667.43** | 0.032467 |
| Monte Carlo 1 | 26.26 | 193.29 | 90 | 85 | 1457744.67 | **0.019242** |
| GA (Binary) | 60.00 | 200.00 | 80 | 75 | **466532.80** | 0.038087 |
| GA (Binary) | 25.00 | 190.09 | 95 | 90 | 1640191.80 | **0.016613** |
| GA (FP) | 56.16 | 194.49 | 95 | 90 | **312430.43** | 0.017951 |
| GA (FP) | 25.35 | 189.58 | 95 | 90 | 1641135.80 | **0.016615** |
| Literature | 63.89 | 183.29 | 85 | 80 | **531059.80** | 0.030182 |
| Literature | 66.45 | 183.36 | 95 | 85 | 694101.00 | **0.023078** |

Table 1: Comparison of results computing the ideal vector of the example (design of a machine tool spindle). For each method the best results for optimum $f_1$ and $f_2$ are shown in **boldface**.

$d_{om}$=25.00 mm
$d_{a1}$=80.00 mm
$d_{a2}$=95.00 mm
$d_{b1}$=75.00 mm
$d_{b2}$=90.00 mm
$p_1$=1.25
$p_2$=1.05
$l_k$=150.00 mm
$l_g$=200.00 mm
$a$=80.00 mm
$E = 210,000.0$ N/mm$^2$
$F = 10,000$ N
$\Delta_a = 0.00540000$ mm
$\Delta_b = -0.00540000$ mm
$\Delta = 0.01000000$ mm
$\delta_{ra} = -0.00100000$ mm
$\delta_{rb} = -0.00100000$ mm

## 11 Comparison of Results

The ideal vector that each method generates will be compared with the best results reported in the literature [3]. The two Monte Carlo methods included in MOSES were used, together with Osyczka's multiobjective optimization system [12] to obtain the ideal vector. Also, several GA-based approaches will be tested using the same parameters (same population size and same crossover and mutation rates). If niching is required, then the niche size will be computed according to the methodology suggested by the developers of the method.

The ideal vector of this problem was computed using the two Monte Carlo Methods included in MOSES (generating 100 points), and a GA (with a population of 100 chromosomes running during 50 generations) using binary and floating point representation. The corresponding results are shown in Table 1, including the best results reported in the literature [3]. The results for Monte Carlo Method 2 are the same than for Method 1. Notice that since Osyczka's multiobjective optimization system is not able to handle discrete variables, no results are available for the min-max method using Osyzcka's system. The GA using both binary and floating point representation found the ideal vector with a procedure to adjust its parameters that has been described somewhere else [2] (the results shown are the best after 81 runs). As can be seen in the results, the best result for the first objective was found using floating point representation, and the best result for the second objective was found using binary representation, although the difference for this second objective is not really significant with respect to the difference for the first objective. The use of this floating point representation in various single and multiobjective optimization problems has been found to be superior (in general) to the binary representation, mainly as we increase the number of variables or their respective allowable ranges [2]. In terms of the multiobjective optimization problem, the new technique introduced in this paper produces a better overall result than any of the other existing approaches, including the mathematical programming techniques. As it turns out, this technique also produces the best sketch of the Pareto front and is able to keep it for as many generations as necessary, contrasting with the other GA-based techniques that either lose the front very quickly (e.g., VEGA, NSGA, and Hajela's method) or aren't able to find it at all (e.g., GALC and the Lexicographic method). The only two approaches with which the new technique can really compete in terms of finding the Pareto front are NPGA y MOGA, not only in this but in most of the other problems analyzed by the author [1].

## 12 Conclusions

A new multiobjective optimization method based on the min-max optimization approach has been proposed. This approach is very robust because it transforms the multiobjective optimization problem into several single objective optimization problems, and it works very well independently of the representation scheme used. However, a floating point representation seems to work better for numerical optimization applications. The main drawback of the new approach is that it requires the ideal vector and a set of weights to delineate the Pareto set. Nevertheless, when the ideal vector is not known, a set of target (desirable) values for each objective can be provided instead. Also, finding proper weights is normally an easy task, since not many of them are required to get reasonably good results.

## 13 Future Work

Much additional work remains to be done, since this is a very broad area of research. For example, it is desirable to do more theoretical work on niches and population sizes for multiobjective optimization problems to verify some of the empirical results obtained by the author. In that sense, it is expected that MOSES may be useful as an experimentation tool for those interested in this area. To talk about convergence in this context seems a rather difficult task, since there is no common agreement on what optimum really means. However,

| **Method** | $\mathbf{x_1}$ | $\mathbf{x_2}$ | $\mathbf{x_3}$ | $\mathbf{x_4}$ | $\mathbf{f_1}$ | $\mathbf{f_2}$ | $\mathbf{L_p(f)}$ |
|---|---|---|---|---|---|---|---|
| Ideal Vector | | | | | 312430.43 | 0.01662 | 0.00000 |
| Monte Carlo 1 | 56.67 | 190.22 | 85 | 80 | 728581.78 | 0.02647 | 1.92555 |
| Monte Carlo 2 | 26.26 | 193.29 | 90 | 85 | 1457744.67 | 0.01924 | 3.82407 |
| GALC (B) | 42.27 | 187.83 | 95 | 90 | 1386131.13 | 0.01696 | 3.45719 |
| GALC (FP) | 42.78 | 188.01 | 95 | 90 | 1377893.38 | 0.01697 | 3.43203 |
| Lexicographic (B) | 62.02 | 200.00 | 95 | 85 | 856072.60 | 0.02184 | 2.05486 |
| Lexicographic (FP) | 61.98 | 190.91 | 95 | 80 | 709307.00 | 0.02619 | 1.84682 |
| VEGA (B) | 54.63 | 200.00 | 90 | 85 | 987526.38 | 0.02124 | 2.43936 |
| VEGA (FP) | 54.45 | 191.11 | 95 | 90 | 1151553.50 | 0.01775 | 2.75405 |
| NSGA (B) | 65.22 | 200.00 | 90 | 85 | 708412.19 | 0.02439 | 1.73510 |
| NSGA (FP) | 62.00 | 197.36 | 95 | 90 | 985238.13 | 0.01884 | 2.28746 |
| MOGA (B) | 65.52 | 200.00 | 90 | 85 | 699786.88 | 0.02453 | 1.71643 |
| MOGA (FP) | 67.75 | 189.34 | 95 | 90 | 800608.63 | 0.02011 | 1.77289 |
| NPGA (B) | 57.92 | 200.00 | 90 | 75 | 654768.06 | 0.03223 | 2.03595 |
| NPGA (FP) | 43.53 | 187.86 | 95 | 90 | 1363536.50 | 0.01701 | 3.38794 |
| Hajela (B) | 59.87 | 188.12 | 95 | 80 | 757841.81 | 0.02498 | 1.92946 |
| Hajela (FP) | 61.19 | 188.10 | 95 | 90 | 975296.19 | 0.01861 | 2.24167 |
| GAminmax (B) | 66.99 | 200.00 | 90 | 85 | 656950.38 | 0.02532 | 1.62676 |
| GAminmax (FP) | 71.98 | 188.17 | 95 | 90 | 672894.56 | 0.02169 | 1.45917 |

Table 2: Comparison of the best overall solution found by each one of the methods included in MOSES for the example given. GA-based methods were tried with binary (B) and floating point (FP) representations. The following abbreviations were used: GALC = Genetic Algorithm with a linear combination of objectives using scaling. In all cases, weights were assumed equal to 0.5 (equal weight for every objective).

if we use concepts from Operations Research such as the min-max optimum, it should be possible to develop such a theory of convergence for these kinds of problems. Also, it is highly desirable to be able to find more ways of incorporating knowledge about the domain into the GA, as long as it can be automatically assimilated by the algorithm during its execution and does not have to be provided by the user (to preserve its generality).

## References


[1] Carlos Artemio Coello Coello. *An Empirical Study of Evolutionary Techniques for Multiobjective Optimization in Engineering Design.* PhD thesis, Department of Computer Science, Tulane University, New Orleans, LA, apr 1996.

[2] Carlos A. Coello Coello, Filiberto Santos Hernández, and Francisco Alonso Farrera. Optimal design of reinforced concrete beams using genetic algorithms. *Expert Systems with Applications: An International Journal*, 12(1):101–108, January 1997.

[3] Hans Eschenauer, Juhani Koski, and Andrzej Osyczka, editors. *Multicriteria Design Optimization.* Springer-Verlag, Berlin, Germany, 1990.

[4] 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.

[5] Carlos M. Fonseca and Peter J. Fleming. An overview of evolutionary algorithms in multiobjective optimization. Technical report, Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, U. K., 1994.

[6] M. P. Fourman. Compaction of symbolic layout using genetic algorithms. In *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, pages 141–153. Lawrence Erlbaum, 1985.

[7] David E. Goldberg and J. Richardson. Genetic algorithm with sharing for multimodal function optimization. In J. J. Grefenstette, editor, *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, pages 41–49. Lawrence Erlbaum, 1987.

[8] J. J. Grefenstette. GENESIS: A system for using genetic search procedures. In *Proceedings of the 1984 Conference on Intelligent Systems and Machines*, pages 161–165, 1984.

[9] P. Hajela and C. Y. Lin. Genetic search strategies in multicriterion optimal design. *Structural Optimization*, 4:99–107, 1992.

[10] J. Horn and N. Nafpliotis. Multiobjective Optimization using the Niched Pareto Genetic Algorithm. Technical Report IlliGAl Report 93005, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA, 1993.

[11] A. Osyczka. An approach to multicriterion optimization for structural design. In *Proceedings of International Symposium on Optimal Structural Design*. University of Arizona, 1981.

[12] Andrzej Osyczka. *Multicriterion Optimization in Engineering with FORTRAN programs*. Ellis Horwood Limited, 1984.

[13] Andrzej Osyczka. Multicriteria optimization for engineering design. In John S. Gero, editor, *Design Optimization*, pages 193–227. Academic Press, 1985.

[14] S. Rao. Game theory approach for multiobjective structural optimization. *Computers and Structures*, 25(1):119–127, 1986.

[15] S. S. Rao. Multiobjective optimization in structural design with uncertain parameters and stochastic processes. *AIAA Journal*, 22(11):1670–1678, nov 1984.

[16] J. David Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, pages 93–100. Lawrence Erlbaum, 1985.

[17] N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. Technical report, Department of Mechanical Engineering, Indian Institute of Technology, Kanput, India, 1993.

[18] C. H. Tseng and T. W. Lu. Minimax multiobjective optimization in structural design. *International Journal for Numerical Methods in Engineering*, 30:1213–1228, 1990.