Use of Dominance-based Tournament Selection to Handle Constraints in Genetic Algorithms

Carlos A. Coello Coello[†]
Efrén Mezura Montes[*]
[†]CINVESTAV-IPN
Depto. de Ingeniería Eléctrica
Sección de Computación
Av. Instituto Politécnico Nacional No. 2508
Col. San Pedro Zacatenco
México, D. F. 07300, MEXICO
ccoello@cs.cinvestav.mx

[*]Maestría en Inteligencia Artificial
Universidad Veracruzana
Xalapa, Veracruz 91090
emezura@mail.mia.uv.mx

Abstract

In this paper, we propose a dominance-based selection scheme to incorporate constraints into the fitness function of a genetic algorithm used for global optimization. The approach does not require the use of a penalty function and, unlike traditional evolutionary multiobjective optimization techniques, it does not require niching (or any other similar approach) to maintain diversity in the population. The algorithm is validated using two test functions taken from the specialized literature on evolutionary optimization. The results obtained indicate that the approach is a viable alternative to the traditional penalty function, mainly in engineering optimization problems.

1 Introduction

The problem that is of interest to us is the general nonlinear optimization problem in which we want to:

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

subject to:

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

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

where $\vec{x}$ is the solution vector $\vec{x} = [x_1, x_2, \ldots, x_r]^T$, 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). Only inequality constraints will be considered in this work.

Despite the considerable number of constraint-handling methods that have been developed for genetic algorithms in the last few years (see for example [13]), most of them either require a large number of fitness function evaluations, complex encodings or mappings, or are limited to problems with certain (specific) characteristics.

The aim of this work is to show that using concepts from multiobjective optimization [3], it is possible to derive new constraint-handling techniques that are not only easy to implement, but also computationally efficient.

2 Related Work

The idea of using evolutionary multiobjective optimization techniques [3] to handle constraints is not entirely new. A few researchers have reported approaches that rely on the use of multiobjective optimization techniques as we will see in this section.

The most common approach is to redefine the single-objective optimization of f as a multiobjective optimization problem in which we will have $m + 1$ objectives, where m is the number of constraints. Then, we can apply any multiobjective optimization technique [8] to the new vector $\bar{v} = (f, f_1, \ldots, f_m)$, where $f_1, \ldots, f_m$ are the original constraints 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).

Surry et al. [21, 20] proposed the use of Pareto ranking [7] and VEGA [18] to handle constraints using this technique. In their approach, called COMOGA, the population was ranked based on constraint violations (counting the number of individuals dominated by each solution). Then, one portion of the population was selected based on constraint ranking, and the rest based on real cost (fitness) of the individuals. COMOGA compared fairly with a penalty-based approach in a pipe-sizing problem, since the resulting EA was less sensitive to changes in the parameters, but the results achieved were not better than those found with a penalty function [21]. It should be added that COMOGA [21] required several extra parameters, although its authors argue [21] that the technique is not particularly sensitive to the values of such parameters.

Parmee and Purchase [14] implemented a version of the *Vector Evaluated Genetic Algorithm* (VEGA) [18] that handled the constraints of a gas turbine problem as objectives to allow a genetic algorithm to locate a feasible region within the highly constrained search space of this application. However, VEGA

was not used to further explore the feasible region, and instead Parmee and Purchase [14] opted to use specialized operators that would create a variable-size hypercube around each feasible point to help the genetic algorithm to remain within the feasible region at all times. This approach was specially developed for a heavily constrained search space and it proved to be appropriate to reach the feasible region. However, this application of a multiobjective optimization technique does not aim at finding the global optimum of the problem, and the use of special operators suggested by the authors certainly limits the applicability of the approach.

Camponogara & Talukdar [1] proposed the use of a procedure based on an evolutionary multiobjective optimization technique. Their proposal was to restate a single objective optimization problem in such a way that two objectives would be considered: the first would be to optimize the original objective function and the second would be to minimize:

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

where β is normally 1 or 2.

Once the problem is redefined, non-dominated solutions with respect to the two new objectives are generated. The solutions found define a search direction $d = (x_i - x_j)/|x_i - xj|$, where $x_i \in S_i$, $x_j \in S_j$, and S_i and S_j denote Pareto optimal sets. The direction search d is intended to simultaneously minimize all the objectives [1]. Line search is performed in this direction so that a solution x can be found such that x dominates x_i and x_j (i.e., x is a better compromise than the two previous solutions found). Line search takes the place of crossover in this approach, and mutation is essentially the same, where the direction d is projected onto the axis of one variable j in the solution space [1]. Additionally, a process of eliminating half of the population is applied at regular intervals (only the less fit solutions are replaced by randomly generated points).

Camponogara & Talukdar's approach [1] has obvious problems to keep diversity (a common problem with using evolutionary multiobjective optimization techniques), as is indicated by the fact that the technique discards the worst individuals at each generation. Also, the use of line search increases the cost (computationally speaking) of the approach and it is not clear what the impact is in the overall performance of the algorithm of the segment chosen to search.

Jiménez and Verdegay [11] proposed the use of a min-max approach [2] to handle constraints. The main idea of this approach is to apply a set of simple rules to decide the selection process:

1. If the two individuals being compared are both feasible, then select based on the minimum value of the objective function.

2. If one of the two individuals being compared is feasible and the other one is infeasible, then select the feasible individual.

3. If both individuals are infeasible, then select based on the maximum constraint violation (max $g_j(\vec{x})$, for $j = 1, \ldots, m$, and m is the total number of constraints). The individual with the lowest maximum violation wins.

A subtle problem with this approach is that the evolutionary process first concentrates only on the constraint satisfaction problem and therefore it samples points in the feasible region essentially at random [21]. This means that in some cases (e.g., when the feasible region is disjoint) we might land in an inappropriate part of the feasible region from which we will not be able to escape. However, this approach (as in the case of Parmee and Purchase's [14] technique) may be a good alternative to find a feasible point in a heavily constrained search space.

Coello [4] proposed the use of a population-based multiobjective optimization technique such as VEGA [18] to handle each of the constraints of a single-objective optimization problem as an objective. At each generation, the population is split into $m + 1$ sub-populations (m is the number of constraints), so that a fraction of the population is selected using the (unconstrained) objective function as its fitness and another fraction uses the first constraint as its fitness and so on.

For the sub-population guided by the objective function, the evaluation of such objective function for a given vector $\vec{x}$ is used directly as the fitness function (multiplied by (-1) if it is a minimization problem), with no penalties of any sort. For all the other sub-populations, the algorithm used was the following [4] (we are assuming that the GA is maximizing):

if $g_j(\vec{x}) < 0.0$	**then**	fitness $= g_j(\vec{x})$
else if $v \neq 0$	**then**	fitness $= -v$
else		fitness $= f$

where $g_j(\vec{x})$ refers to the amount of constraint violation corresponding to sub-population $j + 1$ (this is assuming that the first sub-population is assigned to the objective function f), and v refers to the number of constraints that are violated ($\leq m$).

This approach provided good results in several optimization problems, but required a relatively large number of fitness function evaluations to converge [4].

The limitations of the previously reported multiobjective optimization techniques used to handle constraints were the main motivation of this work.

3 Our Approach

The concept of nondominated vector is used in multiobjective optimization to denote solutions that represent good compromises or trade-offs, given a set of objective functions. None of the objective function values of these nondominated vectors can be improved without worsening another one (see [3] for details). Our hypothesis is that this concept can be used to extend evolutionary multiobjective optimization techniques to be used as single-objective optimization approaches in which the constraints are handled as additional objectives. Although this sort

of approach can be quite useful to reach the feasible region in highly constrained search spaces, it is not straightforward to extend it to solve single-objective optimization problems. The main difficulty is that we could bias the search towards a certain specific portion of the feasible region and, as a consequence, we could be unable to reach the global optimum. This paper presents a proposal based on a technique known as the Niched-Pareto Genetic Algorithm (NPGA) [10] that uses tournament selection based on nondominance. In the original proposal of the NPGA, the idea was to use a sample of the population to determine who is the winner between two candidate solutions to be selected, and to choose one of them based on nondominance with respect to the sample taken. Since only a portion of the population is used with this technique, it has a lower computational complexity with respect to traditional Pareto ranking approaches (the most common evolutionary multiobjective optimization technique).

To adapt the NPGA to solve single-objective constrained optimization problems, we performed the following changes:

- The tournament performed uses a parameter called selection ratio (S_r), which indicates the minimum number of individuals that will be selected through conventional tournament selection. The remainder will be selected using a purely probabilistic approach. In other words, $(1 - Sr)$ individuals in the population are probabilistically selected.
- When comparing two individuals, we can have three possible situations:
  1. **Both are feasible**. In this case, the individual with a better fitness value wins.
  2. **One is infeasible, and the other is feasible**. The feasible individual wins, regardless of its fitness function value.
  3. **Both are infeasible**. The individual with the lowest amount of constraint violation wins, regardless of its fitness function value.
- Our approach does not require niching or any other approach to keep diversity, since the value of S_r will control the diversity of the population. For the experiments reported in this paper, a value close to one (≥ 0.8) was adopted.

In the following experiments, we used a GA with binary representation, two-point crossover, and uniform mutation. The parameters used for our GA were the following: population size = 200 individuals, maximum number of generations = 400, crossover rate = 0.6, mutation rate = 0.03, $S_r = 0.99$ (i.e., one out of every one hunded selections will be done probabilistically, rather than in a deterministic way), tournament size = 10.

4 Examples

While the method has been tested on several examples taken from the optimization literature, only two will be used here to show the way in which the proposed

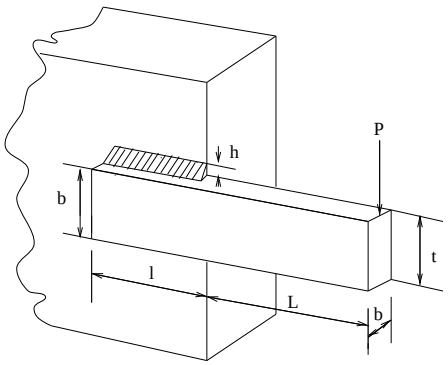

Figure 1: The welded beam used for the first example.

approach works. These examples have linear and nonlinear constraints, and have been previously solved using a variety of other techniques (both GA-based and traditional mathematical programming methods), which is useful to determine the quality of the solutions produced by the proposed approach.

4.1 Example 1 : Welded Beam Design

A welded beam is designed for minimum cost subject to constraints on shear stress (τ), bending stress in the beam (σ), buckling load on the bar (P_c), end deflection of the beam (δ), and side constraints [16]. There are four design variables as shown in Figure 1 [16]: h (x_1), l (x_2), t (x_3) and b (x_4).

The problem can be stated as follows:

Minimize:

$$F(\vec{x}) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14.0 + x_2) \tag{5}$$

Subject to:

$$g_1(\vec{x}) = \tau(\vec{x}) - \tau_{max} \leq 0 \tag{6}$$

$$g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{max} \leq 0 \tag{7}$$

$$g_3(\vec{x}) = x_1 - x_4 \leq 0 \tag{8}$$

$$g_4(\vec{x}) = 0.10471 x_1^2 + 0.04811 x_3 x_4 (14.0 + x_2) - 5.0 \leq 0 \tag{9}$$

$$g_5(\vec{x}) = 0.125 - x_1 \leq 0 \tag{10}$$

$$g_6(\vec{x}) = \delta(\vec{x}) - \delta_{max} \leq 0 \tag{11}$$

$$g_7(\vec{x}) = P - P_c(\vec{x}) \leq 0 \tag{12}$$

where

$$\tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau'\tau'' \frac{x_2}{2R} + (\tau'')^2} \tag{13}$$

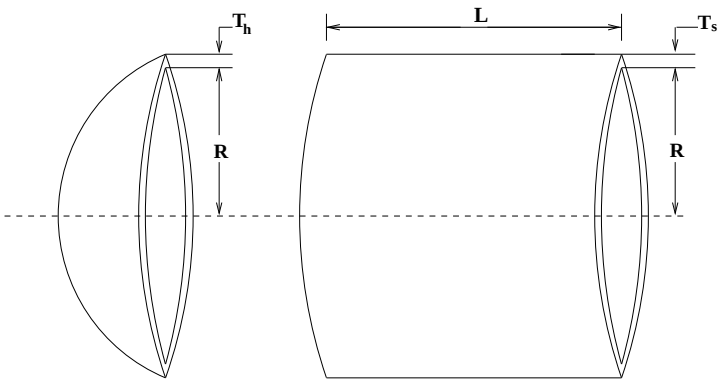

Figure 2: Center and end section of the pressure vessel used for the second example.

$$\tau' = \frac{P}{\sqrt{2}x_1x_2}, \tau'' = \frac{MR}{J}, M = P\left(L + \frac{x_2}{2}\right) \tag{14}$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2} \tag{15}$$

$$J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\} \tag{16}$$

$$\sigma(\vec{x}) = \frac{6PL}{x_4x_3^2}, \delta(\vec{x}) = \frac{4PL^3}{Ex_3^3x_4} \tag{17}$$

$$P_c(\vec{x}) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right) \tag{18}$$

$$P = 6000\ lb, \quad L = 14\ in, \quad E = 30 \times 10^6\ psi, \quad G = 12 \times 10^6\ psi \tag{19}$$

$$\tau_{max} = 13,600\ psi, \quad \sigma_{max} = 30,000\ psi, \quad \delta_{max} = 0.25\ in \tag{20}$$

4.2 Example 2 : Design of a Pressure Vessel

A cylindrical vessel is capped at both ends by hemispherical heads as shown in Figure 2. The objective is to minimize the total cost, including the cost of the material, forming and welding. There are four design variables: T_s (thickness of the shell), T_h (thickness of the head), R (inner radius) and L (length of the cylindrical section of the vessel, not including the head). T_s and T_h are integer multiples of 0.0625 inch, which are the available thicknesses of rolled steel plates, and R and L are continuous. Using the same notation given by Kannan and Kramer [12], the problem can be stated as follows:

Minimize :

$$F(\vec{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \tag{21}$$

Subject to :

$$g_1(\vec{x}) = -x_1 + 0.0193x_3 \leq 0 \tag{22}$$

$$g_2(\vec{x}) = -x_2 + 0.00954x_3 \leq 0 \tag{23}$$

$$g_3(\vec{x}) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1,296,000 \leq 0 \tag{24}$$

$$g_4(\vec{x}) = x_4 - 240 \leq 0 \tag{25}$$

5 Comparison of Results

5.1 Example 1

This problem was solved before by Deb [5] using a simple genetic algorithm with binary representation, and a traditional penalty function as suggested by Goldberg [9]. It has also been solved by Ragsdell and Phillips [15] using geometric programming. Ragsdell and Phillips also compared their results with those produced by the methods contained in a software package called "Opti-Sep" [19], which includes the following numerical optimization techniques: ADRANS (Gall's adaptive random search with a penalty function), APPROX (Griffith and Stewart's successive linear approximation), DAVID (Davidon-Fletcher-Powell with a penalty function), MEMGRD (Miele's memory gradient with a penalty function), SEEK1 & SEEK2 (Hooke and Jeeves with 2 different penalty functions), SIMPLX (Simplex method with a penalty function) and RANDOM (Richardson's random method).

Their results are compared against those produced by the approach proposed in this paper, which are shown in Table 1. In the case of Siddall's techniques [19], only the best solution produced by the techniques contained in "Opti-Sep" is displayed. The solution shown for the technique proposed here is the best produced after 30 runs, and using the following ranges for the design variables: $0.1 \leq x_1 \leq 2.0$, $0.1 \leq x_2 \leq 10.0$, $0.1 \leq x_3 \leq 10.0$, $0.1 \leq x_4 \leq 2.0$.

The mean from the 30 runs performed was $f(\vec{x}) = 1.792654$, with a standard deviation of 0.074713. The worst solution found was $f(\vec{x}) = 1.993408$, which is better than any of the solutions produced by any of the other techniques depicted in Table 1. The number of fitness function evaluations of our approach was 80000.

5.2 Example 2

This problem was solved before by Deb [6] using GeneAS (Genetic Adaptive Search), by Kannan and Kramer using an augmented Lagrangian Multiplier approach [12], and by Sandgren [17] using a branch and bound technique. Their results were compared against those produced by the approach proposed in

Design Variables	Best solution found			
	This paper	**Deb [5]**	**Siddall [19]**	**Ragsdell [15]**
$x_1(h)$	0.205986	0.2489	0.2444	0.2455
$x_2(l)$	3.471328	6.1730	6.2189	6.1960
$x_3(t)$	9.020224	8.1789	8.2915	8.2730
$x_4(b)$	0.206480	0.2533	0.2444	0.2455
$g_1(\vec{x})$	-0.074092	-5758.603777	-5743.502027	-5743.826517
$g_2(\vec{x})$	-0.266227	-255.576901	-4.015209	-4.715097
$g_3(\vec{x})$	-0.000495	-0.004400	0.000000	0.000000
$g_4(\vec{x})$	-3.430043	-2.982866	-3.022561	-3.020289
$g_5(\vec{x})$	-0.080986	-0.123900	-0.119400	-0.120500
$g_6(\vec{x})$	-0.235514	-0.234160	-0.234243	-0.234208
$g_7(\vec{x})$	-58.666440	-4465.270928	-3490.469418	-3604.275002
$f(\vec{x})$	**1.728226**	**2.43311600**	**2.38154338**	**2.38593732**

Table 1: Comparison of the results for the first example (optimal design of a welded beam).

this paper, and are shown in Table 2. The solution shown for the technique proposed here is the best produced after 30 runs, and using the following ranges for the design variables: $1 \leq x_1 \leq 99$, $1 \leq x_2 \leq 99$, $10.0 \leq x_3 \leq 200.0$, $10.0 \leq x_4 \leq 200.0$. The values for x_1 and x_2 were considered as integer (i.e., real values were rounded up to their closest integer value) multiples of 0.0625, and the values of x_3 and x_4 were considered as real numbers.

The mean from the 30 runs performed was $f(\vec{x}) = 6177.253268$, with a standard deviation of 130.929702. The worst solution found was $f(\vec{x}) = 6469.322010$. We can see that in this case, our average solution was better than any of the solutions produced by any of the other techniques depicted in Table 2. The total number of fitness function evaluations performed was 80000.

6 Conclusions and Future Work

This paper has introduced a new constraint-handling approach that is based on a multiobjective optimization technique called NPGA. The approach is intended to be used with evolutionary algorithms as a way to reduce the burden normally associated with the fine-tuning of a penalty function.

The proposed approach performed well in several test problems in terms of the quality of the solutions found and it requires a relatively low number of fitness function evaluations. The results produced were compared against those generated with other (evolutionary and mathematical programming) techniques reported in the literature.

Our future work involves more validation of the approach with a larger set of test functions (we have used about ten test functions so far although,

Design Variables	Best solution found			
	This paper	**GeneAS [6]**	**Kannan [12]**	**Sandgren [17]**
$x_1(T_s)$	0.8125	0.9375	1.125	1.125
$x_2(T_h)$	0.4375	0.5000	0.625	0.625
$x_3(R)$	42.097398	48.3290	58.291	47.700
$x_4(L)$	176.654047	112.6790	43.690	117.701
$g_1(\vec{x})$	-0.000020	-0.004750	0.000016	-0.204390
$g_2(\vec{x})$	-0.035891	-0.038941	-0.068904	-0.169942
$g_3(\vec{x})$	-27.886075	-3652.876838	-21.220104	54.226012
$g_4(\vec{x})$	-63.345953	-127.321000	-196.310000	-122.299000
$f(\vec{x})$	**6059.946341**	**6410.3811**	**7198.0428**	**8129.1036**

Table 2: Comparison of the results for the second example (optimization of a pressure vessel).

due to space limitations, only the results of two of them are presented in this paper), and an statistical analysis of the impact of its parameters on the overall performance of the algorithm.

7 Acknowledgments

The first author acknowledges support from CONACyT through the NSF-CONACyT project number 32999 A. The second author acknowledges support from CONACyT through a scholarship to pursue graduate studies at the Maestría en Inteligencia Artificial of the Universidad Veracruzana.

References

[1] 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.

[2] V. Chankong and Y. Y. Haimes. *Multiobjective Decision Making: Theory and Methodology*. Systems Science and Engineering. North-Holland, 1983.

[3] Carlos A. Coello Coello. A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. *Knowledge and Information Systems. An International Journal*, 1(3):269–308, August 1999.

[4] Carlos A. Coello Coello. Treating Constraints as Objectives for Single-Objective Evolutionary Optimization. *Engineering Optimization*, 32(3):275–308, 2000.

[5] Kalyanmoy Deb. Optimal Design of a Welded Beam via Genetic Algorithms. *AIAA Journal*, 29(11):2013–2015, November 1991.

[6] Kalyanmoy Deb. GeneAS: A Robust Optimal Design Technique for Mechanical Component Design. In Dipankar Dasgupta and Zbigniew Michalewicz, editors, *Evolutionary Algorithms in Engineering Applications*, pages 497–514. Springer-Verlag, Berlin, 1997.

[7] 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.

[8] Carlos M. Fonseca and Peter J. Fleming. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3(1):1–16, Spring 1995.

[9] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Co., Reading, Massachusetts, 1989.

[10] 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.

[11] Fernando Jiménez and José L. Verdegay. Evolutionary techniques for constrained optimization problems. In *7th European Congress on Intelligent Techniques and Soft Computing (EUFIT'99)*, Aachen, Germany, 1999. Springer-Verlag.

[12] B. K. Kannan and S. N. Kramer. An Augmented Lagrange Multiplier Based Method for Mixed Integer Discrete Continuous Optimization and Its Applications to Mechanical Design. *Journal of Mechanical Design. Transactions of the ASME*, 116:318–320, 1994.

[13] Zbigniew Michalewicz and Marc Schoenauer. Evolutionary Algorithms for Constrained Parameter Optimization Problems. *Evolutionary Computation*, 4(1):1–32, 1996.

[14] 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.

[15] K. M. Ragsdell and D. T. Phillips. Optimal Design of a Class of Welded Structures Using Geometric Programming. *ASME Journal of Engineering for Industries*, 98(3):1021–1025, 1976. Series B.

[16] Singiresu S. Rao. *Engineering Optimization.* John Wiley and Sons, third edition, 1996.

[17] E. Sandgren. Nonlinear integer and discrete programming in mechanical design. In *Proceedings of the ASME Design Technology Conference*, pages 95–105, Kissimine, Florida, 1988.

[18] 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.

[19] James N. Siddall. *Analytical Design-Making in Engineering Design.* Prentice-Hall, 1972.

[20] Patrick D. Surry and Nicholas J. Radcliffe. The COMOGA Method: Constrained Optimisation by Multiobjective Genetic Algorithms. *Control and Cybernetics*, 26(3), 1997.

[21] Patrick D. Surry, Nicholas J. Radcliffe, and Ian D. Boyd. A Multi-Objective Approach to Constrained Optimisation 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. Springer-Verlag, Sheffield, U.K., 1995.